



TP-2

Notés, Automatique

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

1 Consignes

1.1 Préliminaires

Le non respect des consignes suivantes impliquera une note de 0 à la note de simulation.

1.2 Rendu

Une fois que vous aurez récupéré les fichiers via `git` (voir les slides du tp), il faut aller dans le répertoire `tp`. À la fin de votre travail celui-ci devra contenir en outre :

- `simu_pendule_inv_etu`
- `pendule_inv_etu.slx`
- `simu_pendule_inv_etu.m`
- `pendule_inv_capteur_etu.slx`
- `simu_pendule_inv_capteur_etu.m`
- `pendule_inv_echant_etu.slx`
- `simu_pendule_inv_echant_etu.m`
- Ressources

Lorsque vous aurez terminé ce travail (**et uniquement à ce moment-là**), il vous faut

1. renommer le répertoire `tp` en `<nom1>_<nom2>_<groupe>`;
2. compresser ce répertoire (commande `tar -cvzf <nom1>_<nom2>_<groupe>.tgz <nom1>_<nom2>_<groupe>`);
3. déposer le fichier compressé `<nom1>_<nom2>_<groupe>.tgz` dans Moodle (dans le bon groupe) **les fichiers `.rar` sont à prohiber.**

1.3

Vous avez à votre disposition afin de tester vos résultats des codes matlab (par exemple pour le pendule inversé se sont des fichiers de nom `simu_pendule_inv*.m`). Pour pouvoir réaliser ces tests il est nécessaire de **respecter la structure des répertoires, les noms des fichiers SIMULINK ainsi que les noms des constantes et variables**. Pour le pendule simple inversé par exemple, il faut impérativement respecter les noms des constantes et variables suivants (en respectant la casse) :

```
t0 = 0;
g = 9.81; l = 10;
xe = [0 0]'; % vecteur colonne
ue = 0;
x0 = [pi/20 0]'; % vecteur colonne
tf = 10;
K = [30 10]; % vecteur ligne
pas % pour le pas d'intégration numérique dans le cas du schéma d'Euler
delta_t % constante d'échantillonnage
X % Etat de type Timeseries (option par défaut)
U % contrôle de type Timeseries (option par défaut)
```

1.4 Notation

Nous donnons ci-après à titre indicatif un barème qui pourra être modifié.

— Simulation du pendule simple inversé contrôlé : 10

2 Simulation du pendule inversé contrôlé

2.1 Problème

L'objectif de ce TP est de simuler le pendule inversé contrôlé par retour de sortie. On rappelle que ce système 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, **attention ce sont ici les notations mathématiques et non Matlab**.

— $g = 9.81$;
— $l = 10$;

- $t_0 = 0$;
- $x_e = (0, 0)$;
- $u_e = 0$;
- $u(t) = u_e + K(x(t) - x_e)$
- $K = (k_1, k_2)$.

2.2 contrôle par retour d'état

On rappelle que pour contrôler asymptotiquement le système, si $(\alpha_0, \dot{\alpha}_0)$ est suffisamment proche de x_e , il suffit que :

- $k_1 > g$;
- $k_2 > 0$.

Noms des fichiers :

- fichier SIMULINK : `pendule_inv_etu.slx` ;
- script de test `simu_pendule_inv_etu.m`.

Réaliser le schéma SIMULINK de la figure 1.

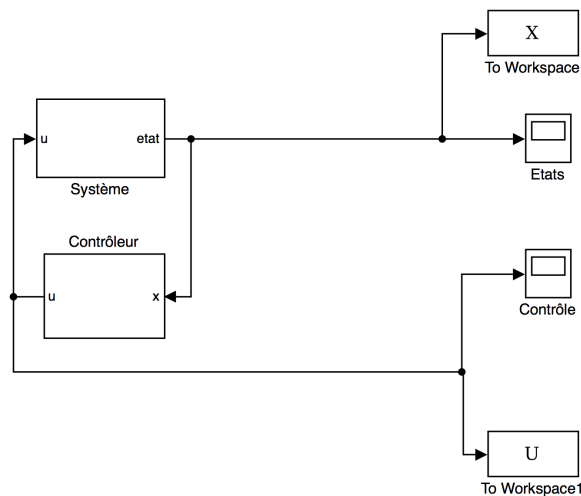


FIGURE 1 – Schéma SIMULINK d'un contrôle par retour d'état.

On modifiera le script de test (sans changer le nom!) afin de visualiser les résultats pour les données de la table 1.

Cas	x_0	t_f	K	Intégrateur
Cas 1.1	$(\pi/20, 0)$	10	$(30, 10)$	par défaut, ode45
Cas 1.2	$(\pi/20, 0)$	10	$(10, 1)$	par défaut, ode45
Cas 1.3	$(\pi/20, 0)$	100	$(10, 1)$	par défaut, ode45
Cas 1.4	$(\pi/20, 0)$	100	$(10, 1)$	Euler, ode1
Cas 1.5	$(\pi/20, 0)$	1000	$(10, 1)$	Euler, ode1
Cas 1.6	$(\pi/20, 0)$	1000	$(10, 1)$	par défaut, ode45
Cas 1.7	$(\pi/20, 0)$	100	$(10, 1)$	Euler, ode1, pas=10
Cas 2.1	$(\pi/10, 0)$	100	$(10, 1)$	par défaut, ode45
Cas 2.2	$(\pi/10, 0)$	100	$(30, 10)$	par défaut, ode45

TABLE 1 – Données pour le contrôle par retour d'état.

2.3 Capteurs

Noms des fichiers :

- fichier SIMULINK : `pendule_inv_capteur_etu.slx`;
- script de test `simu_pendule_inv_capteur_etu.m`.

On suppose maintenant que l'on a accès qu'à α . On introduit donc dans le schéma deux sous systèmes : un capteur et un prédictor (on utilisera un intégrateur continu pour la prédiction) pour reconstruire α , voir la figure 2.

Réaliser le schéma SIMULINK de la figure 2.

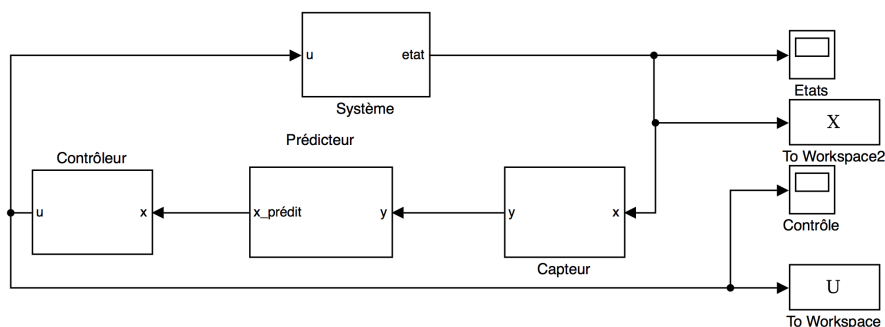


FIGURE 2 – Schéma SIMULINK d'un contrôle par retour de sortie avec prédiction de l'état.

On modifiera le script de test (sans changer le nom !) afin de visualiser les résultats pour les données de la table 2.

Cas	x_0	t_f	K	pas/RelTol	Intégrateur
Cas 1	$(\pi/20, 0)$	100	(10, 1)	par défaut	par défaut, ode45
Cas 2	$(\pi/20, 0)$	100	(10, 1)	1e-10	par défaut, ode45
Cas 3	$(\pi/20, 0)$	100	(10, 1)	0.001	Euler, ode1
Cas 4	$(\pi/20, 0)$	100	(10, 1)	1	Euler, ode1
Cas 5	$(\pi/20, 0)$	100	(10, 1)	2	Euler, ode1
Cas 6	$(\pi/20, 0)$	100	(10, 1)	5	Euler, ode1

TABLE 2 – Données pour le contrôle par retour d'état avec capteurs.

2.4 Échantillonnage

Noms des fichiers :

- fichier SIMULINK : `pendule_inv_echant_etu.slx`;
- script de test `simu_pendule_inv_echant_etu.m`.

En pratique on a accès aux données du capteur avec une période d'échantillonnage de Δt . Pour cela on utilisera à l'intérieur du sous système Capteur le block SIMULINK **Zero-Order Hold** pour réaliser l'échantillonnage et un intégrateur discret pour la prédiction.

On modifiera le script de test (sans changer le nom!) afin de visualiser les résultats pour les données de la table 3

Cas	x_0	t_f	K	Δt	Intégrateur
Cas 1	$(\pi/20, 0)$	100	(30, 10)	0.1	par défaut, ode45
Cas 2	$(\pi/20, 0)$	100	(30, 10)	0.2	par défaut, ode45
Cas 3	$(\pi/20, 0)$	100	(30, 10)	0.3	par défaut, ode45
Cas 4	$(\pi/20, 0)$	100	(30, 10)	0.4	par défaut, ode45

TABLE 3 – Données pour le contrôle par retour d'état avec capteurs et échantillonnage.