TP notés, Automatique
Simulation du pendule inversé contrôlé par retour d'état et de sortie
Simulation du robot Lego
Portage sur le robot Lego

1 Consignes

1.1 Préliminaires

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

La date limite du rendu sous Moodle dans le groupe correspondant est le vendredi 6 décembre à 18h00.

1.2 Rendu

Vous créerez un répertoire qui s'appellera par vos noms (le travail est à réaliser en binôme). Vous mettrez ensuite sous `Moodle` (dans le groupe qui vous concerne) le fichier obtenu après archivage et compression (commande `tar -cvzf <nom1>_<nom2>_<groupe>.tgz <nom1>_<nom2>_<groupe>`). **En particulier les fichier `.rar` sont à prohiber.**
Une fois décompressé, la structure du répertoire obtenu `<nom1>_<nom2>_<groupe>` **devra être** :

- `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`
- `simu_robot_etu`
  - `fonction.m`

- `matrices.m`
- `robot_etu.slx`
- `robot_capteur_etu.slx`
- `robot_echant_etu.slx`
- `simu_robot_etu.m`
- `Ressources`

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 example, il faut impérativement respecter les nom des et constantes et variables suivants (, respectant la casse) :

```
t0 = 0;
g = 9.81; l = 10;
xe = [0 0]';
ue = 0;
x0 = [pi/20 0]';
tf = 10;
K = [30 10];
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
- Simulation du Robot : 4
- Robot Lego NXT : 6
  L'évaluation de cette partie sera faite lors du dernier TP, toute absence sera sanctionnée par un 0.
- Il y aura quelques questions lors de l'examen sur cette partie pratique.

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.

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

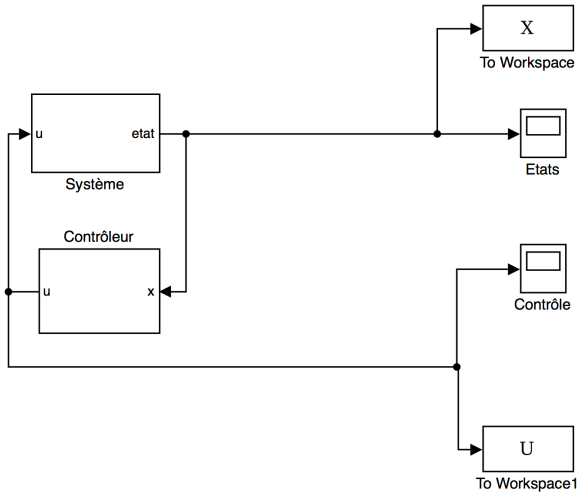

FIGURE 1 – *Schéma* SIMULINK *d'un contrôle par retour d'état.*

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 $\dot{\alpha}$. On introduit donc dans le schéma deux sous systèmes : un capteur et un prédicteur (on utilisera un intégrateur discret pour la prédiction) pour reconstruire α , voir la figure 2.

Réaliser le schéma SIMULINK suivant

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

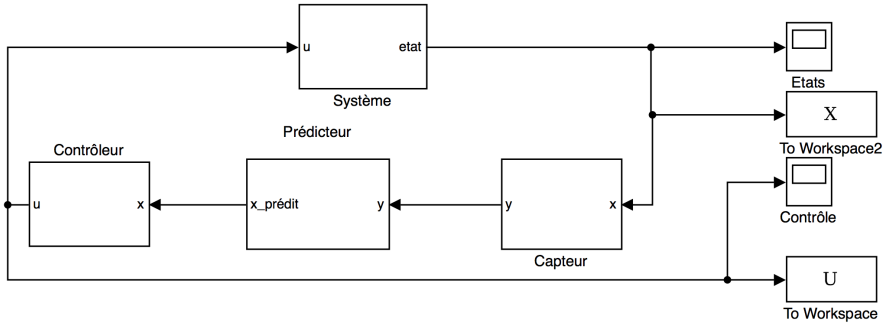

FIGURE 2 – *Schéma* SIMULINK *d'un contrôle par retour de sortie avec prédiction de l'état.*

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 Δ_t. Pour cela on utilisera à l'intérieur du sous système Capteur le block SIMULINK `Zero-Order Hold` pour réaliser l'échantillonage 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	Δ_t	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.*

3 Simulation du robot Lego

3.1 Préliminaires

Cette séance consister à modéliser et simuler le robot Lego pendule inversé qui sera exploité dans les séances de TP suivantes.

3.2 Modèle continue

3.2.1 Utilisation de MatLab pour représenter le modèle du robot Lego pendule inversé

L'objectif est d'étudier le modèle du robot Lego pendule inversé. Construire un modèle Simulink continu comportant un système et un contrôleur par retour d'état. La fonction `f(x,u)` sera implantée avec un bloc MATLAB Function qui contiendra le code disponible dans le fichier `fonction.m`.

Simuler le modèle réalisé.

3.2.2 Synthèse du contrôleur pour le modèle du robot Lego pendule inversé

L'objectif est de calculer les coefficients K pour le contrôleur par retour d'états. Il faut pour cela calculer linéariser le système et calculer les matrices A et B. Celles-ci sont disponibles dans le fichier MatLab `matrices.m`. .

Il faut ensuite calculer les coefficients K à partir des valeurs propres souhaitées et des matrices A et B. La boîte à outils MatLab propose la fonction `place(A,B,V)` dont les paramètres sont les matrices A et B ainsi que le vecteur de valeurs propres V et qui renvoie $-K$.

Simuler le modèle controlé réalisé.

3.3 Introduction des capteurs et actionneurs

L'état du système est observé par des capteurs qui ne restituent pas l'intégralité des composantes de l'état. Un gyroscope mesure la vitesse de changement d'angle du corps du robot $\dot{\psi}(t)$ et un capteur mesure l'angle $\theta(t)$. Pour compenser cette perte d'information, il faut introduire un sous-système prédicteur qui recalcule les informations manquantes à partir des informations disponibles.

Modéliser le capteur et le prédicteur.

Simuler le modèle contrôlé ainsi réalisé.

3.4 Construction du modèle hybride

Le contrôleur et le prédicteur seront implantés en logiciel dans le robot Lego, donc sous la forme d'un modèle discret. Le système représentant la physique reste un modèle continu.

Introduire dans le capteur un bloc Zero-Order Hold de l'onglet Discrete de la bibliothèque Simulink. L'état reconstruit en sortie du capteur est ainsi discret.

Modifier le prédicteur pour utiliser des opérateurs discrets implantés à partir de blocs élémentaires ou de code MatLab.

Simuler le modèle controlé réalisé.

3.5 sensibilité aux paramètres

En vous inspirant de l'appel à SIMULINK via matlab par un appel du type `sim(fich_simulink_etu,[t0 tf],options_sim);` (voir cet appel dans le script `Ressources/simu_robot.m`) compléter le script `simulation_robot.m` afin de réaliser une étude de sensibilité à l'un des paramètres suivants :

- x_0 ;
- K ;
- Δ_t.

4 Code embarqué sur le robot Lego

Voir les documents spécifiques