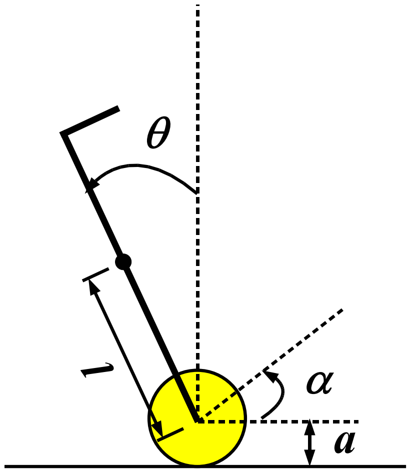
Espace d'état
TP N° 2

Le SegWay, représenté ci-dessous, est un véhicule à deux roues pour un seul essieu. Dans ce TP, on cherchera à stabiliser ce véhicule au moyen d'une commande par retour d'état. Le comportement du Segway, est très proche de celui du monocycle dessiné ci-dessous.



On note M la masse de la roue, a son rayon et J_M son moment d'inertie. On note m la masse du pendule, l la distance entre le centre de gravité du pendule et le centre du disque, J_P le moment d'inertie du pendule.
$\theta(t)$ désigne l'angle entre le pendule et la verticale. $\alpha(t)$ est la position angulaire de la roue.
Le comportement du Segway peut être modélisé par une représentation d'état. Nous choisissons comme vecteur d'état :

$$x(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \end{bmatrix} = \begin{bmatrix} \alpha(t) \\ \theta(t) \\ \dot{\alpha}(t) \\ \dot{\theta}(t) \end{bmatrix}$$

L'entrée, $u(t)$, est le couple moteur appliqué sur le roue.
L'équation d'état obtenue est la suivante :

$$\left\{ \begin{array}{rcl} \dot{x}_1 & = & x_3 \\ \dot{x}_2 & = & x_4 \\ \dot{x}_3 & = & \dfrac{\mu_3 \left[\mu_2 x_4^2 - \mu_g \cos(x_2)\right] \sin(x_2) + \left[\mu_2 + \mu_3 \cos(x_2)\right] u(t)}{\mu_1 \mu_2 - \mu_3^2 \cos^2(x_2)} \\ \\ \dot{x}_4 & = & \dfrac{\left[\mu_1 \mu_g - \mu_3^2 x_4^2 \cos(x_2)\right] \sin x_2 - \left[\mu_1 + \mu_3 \cos(x_2)\right] u}{\mu_1 \mu_2 - \mu_3^2 \cos^2(x_2)} \end{array} \right.$$

avec

$$\mu_1 = J_M + a^2(m+M), \quad \mu_2 = J_P + ml^2, \quad \mu_3 = aml, \quad \mu_g = glm$$

Un capteur de position angulaire permet de mesurer la position angulaire de la roue. L'équation de sortie est donc :

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} x(t)$$

Applications numériques : $M = 1\,\mathrm{kg}$, $a = 0.1\,\mathrm{m}$, $J_M = 0.0025\,\mathrm{kg.m^2}$, $m = 0.5\,\mathrm{kg}$, $l = 0.5\,\mathrm{m}$, $J_P = 0.0417\,\mathrm{kg.m^2}$

1 Simulation du système non linéaire

L'objectif de cette partie est de simuler, sous simulink, le comportement du système non linéaire. Le bloc *S-Funtion* permet de simuler facilement le comportement de représentations d'état complexes. Ce bloc se situe dans la bibliothèque *User-Defined Functions.* Afin d'obtenir de l'aide sur ce type de bloc, on pourra utiliser la commande *help sfuntmpl.* On pourra également s'inspirer de l'exemple ci-dessous.

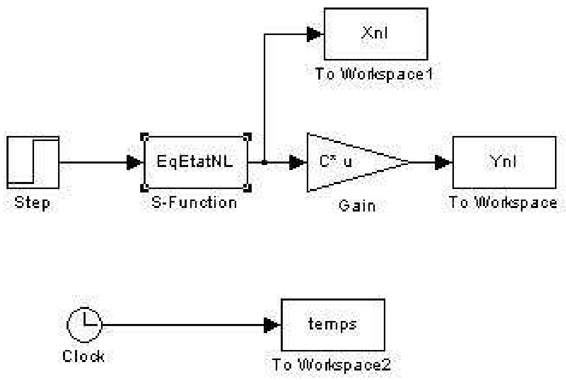
L'exemple ci-après présente une utilisation du bloc *S-Funtion* pour la simulation d'un système modélisé par une représentation d'état non linéaire. On ne codera pas cet exemple mais on s'en inspirera pour simuler le comportement du Segway. Le système considéré dans cet exemple a pour équation d'état :

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -x_1^2\, x_2 + \mu_1\, \sin(x_1)\, u \end{cases}$$

L'équation de sortie est linéaire :

$$y = C\, x(t)$$

Il faut, tout d'abord, définir, sous Matlab, la valeur du paramètre μ_1, de la matrice C et du vecteur d'état initial X_0. Le système est simulé à l'aide du schéma Simulink suivant :



Le paramètre *Save format* des blocs *To Workspace* est *Array.* Afin d'avoir accès au vecteur d'état, le bloc *S-function* ne simule que l'équation d'état. La sortie du bloc *S-function* est donc le vecteur d'état. Le vecteur de sortie $y(t)$ est obtenu en utilisant le bloc *Gain* avec pour valeur C. On veillera à sélectionner pour le paramètre *Multiplication* du bloc *Gain*, le format *Matrix(K*u).* Les paramètres choisis pour le bloc *S-Funtion* sont :

- Champ *S-function name* : EqEtatNL. On indique ainsi que l'équation à simuler est décrite par la fonction *EqEtatNL* contenue dans le fichier *EqEtatNL.m.* Cette fonction est présentée ci-après.
- Champ *S-function parameters* : *mu1 , X0.* Ce sont les paramètres fournis à la fonction *EqEtatNL.*

L'entête de la fonction *EqEtatNL* a un format imposé :

function[sys,Xinit,str,ts]=EqEtatNL(t,X,u,FLAG,mu1,X0)

Les paramètres d'entrée *mu1* et *X0* sont les valeurs qui ont été définies dans le champ *S-function parameters* du bloc *S-Function.* Les valeurs des autres paramètres d'entrée de la fonction sont gérées par Simulink :

- t est le temps courant
- X est la valeur du vecteur d'état au temps courant.

- u est la valeur du vecteur d'entrée au temps courant
- $FLAG$ est une varable gérée par simulink afin d'indiquer l'ordonnancement des calculs :
  - $FLAG$ n'a la valeur 0 qu'à l'instant initial. Il faut, alors indiquer, dans la variable de sortie *sys*, la taille des différents signaux d'entrée, de sortie et du vecteur d'état. Il faut indiquer la valeur de l'état initial dans *Xinit*.
  - Lorsque *FLAG=1*, il faut calculer $\dot{x}(t)$ i.e. coder l'équation d'état. La valeur de $\dot{x}(t)$ doit être donnée à la variable *sys*.
  - Lorsque *FLAG=3*, il faut donner à *sys*, la valeur du vecteur de sortie de la *S-Function*. Dans cet exemple, on prendra *sys=X* de façon à disposer de la valeur du vecteur d'état en sortie de la *S-Function*.

Pour cet exmple, la fonction *EqEtatNL* est :

```
function[sys,Xinit,str,ts]=EqEtatNL(t,X,u,FLAG,mu1,X0)

switch(FLAG)

    case 0 %Initialisation
        dimensions=simsizes;
        dimensions.NumContStates=2; %Dimension du vecteur d'état x(t)
        dimensions.NumOutputs=2; %Dimension de la sortie de la S-Funtion
        dimensions.NumInputs=1; %Dimension de l'entrée
        dimensions.NumSampleTimes=1;%Toujours choisir cette valeur pour ce paramètre
        sys=simsizes(dimensions);

        Xinit=X0; %valeur de l'état initial
        str=[]; %toujours définir ce paramètre ainsi
        ts=[0,0]; %toujours définir ainsi pour les systèmes à temps continu


    case 1 %Définition de l'équation d'état
        sys(1,1)=X(2,1);
        sys(2,1)=-(X(1,1))^2*X(2,1)+mu1*sin(X(1,1))*u;


    case 3 %Equation de sortie de la S-Function
        sys=X;
end;
```

2 Etude en boucle ouverte

1. Montrer que ($x_e = 0$; $u_e = 0$) est un point d'équilibre du système non linéaire.
2. Montrer qu'en linéarisant la représentation d'état autour du point d'équilibre ($x_e = 0$; $u_e = 0$), on obtient une représentation d'état linéaire :

$$\begin{cases} \dot{x} = A\,x + B\,u \\ y = C\,x \end{cases}$$

avec

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \dfrac{-\mu_3\mu_g}{\mu_1\mu_2-\mu_3^2} & 0 & 0 \\ 0 & \dfrac{\mu_1\mu_g}{\mu_1\mu_2-\mu_3^2} & 0 & 0 \end{bmatrix}, \qquad \begin{bmatrix} 0 \\ 0 \\ \dfrac{\mu_2+\mu_3}{\mu_1\mu_2-\mu_3^2} \\ \dfrac{-(\mu_1+\mu_3)}{\mu_1\mu_2-\mu_3^2} \end{bmatrix}$$

3. Définir les matrices A, B et C sous Matlab et définir la représentation d'état linéaire à l'aide l'aide de la commande *ss*. Le système linéaire est-il commandable, est-il observable, est-il stable?

4. On choisit comme état initial $x_0 = \begin{bmatrix} 0 & 0.08 & 0 & 0 \end{bmatrix}^T$, comme entrée un échelon d'amplitude 0.1 N.m. et comme temps de fin de simulation 2 secondes. Pour chaque composante du vecteur d'état, tracer sur une même figure l'évolution obtenue par simulation du système non linéaire et celle obtenue par simulation du système linéarisé. Vous semble-t-il pertinent de synthétiser une loi de commande en utilisant le système linéarisé.

3 Commande par retour d'état

On souhaite stabiliser le Segway. Pour cela, on met en œuvre une commande par reour d'état. Dans cette partie, on supposera que l'on a accès au vecteur d'état. Un observateur sera synthétisé dans la partie suivante.
La synthèse se fera en se basant sur le système linéarisé.

1. Synthétiser une commande par retour d'état avec action intégrale telle que les pôles de la boucle fermée soient tous égaux à -1.

2. Simuler le comportement du système linéaire en boucle fermée pour un échelon de consigne d'amplitude 0.1 rad. Tracer également l'évolution de la seconde composante du vecteur d'état. Le comportement est-il satisfaisant?

3. Asservir le système non linéaire avec la loi de commande synthétisée ci-dessus et comparer le comportement du système non linéaire avec celui du système linéaire.

4 Observateur

Afin de pouvoir mettre en œuvre la loi de commande synthétisée dans la section précédente, il est nécessaire de connaitre le vecteur d'état. Pour cela on met en œuvre un observateur.

1. Synthétiser un observateur pour le système linéaire. On choisira les pôles de l'observateur tous égaux à -22.

2. Evaluer le comportement de cet observateur en simulation. Dans un premier temps, afin d'évaluer le comportement de l'observateur, la commande sera calculée à partir des valeurs réelles de l'état. On comparera le comportement de l'observateur sur le système linéaire et sur le système non linéaire. On évaluera également le comportement de l'observateur lorsque son état initial est différent de l'état initial réel.

3. Evaluer le comportement du système lorsque la commande est calculée à partir des états estimés par l'observateur.