ENSTA PARIS

AUT202 – AUTOMATIQUE : DYNAMIQUE ET CONTROLE DES SYSTEMES

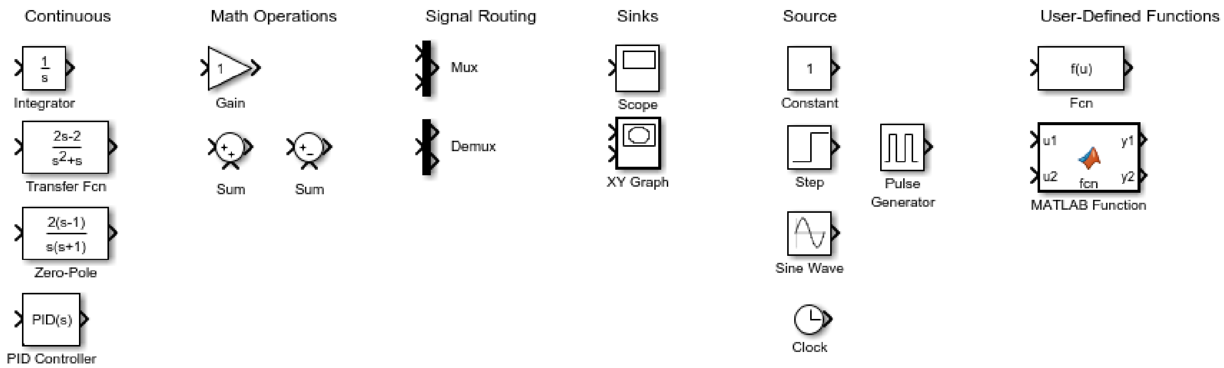
TRAVAUX PRATIQUES MATLAB N°1

1ER FEVRIER 2023

INTRODUCTION

MATLAB est un logiciel scientifique spécialisé dans le calcul numérique. Il s'utilise comme un logiciel interactif ou comme un langage de programmation. La documentation se consulte à l'aide de **help** suivi du nom de la fonction utilisée.

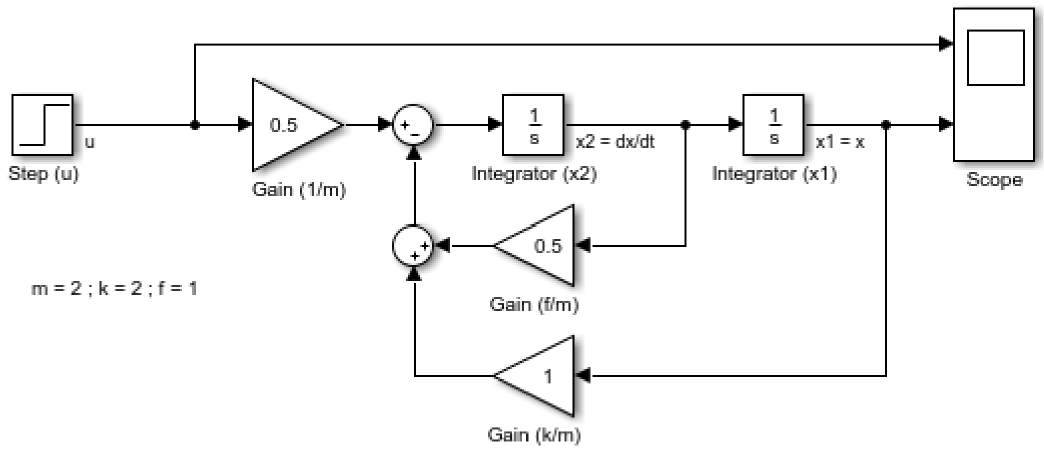
Le module Simulink de MATLAB permet de simuler le comportement de systèmes dynamiques représentés sous forme de schéma bloc. L'interface est constituée d'une bibliothèque de blocs génériques et d'une feuille de travail sur laquelle l'utilisateur assemble les blocs génériques dont il a besoin pour représenter son système dynamique. Quelques blocs utiles sont listés ci-dessous :



Prenons l'exemple d'un oscillateur amorti dont la dynamique s'écrit :

$$m.\frac{d^2}{dt^2}x(t) = -k.x(t) - f.\frac{d}{dt}x(t) + u(t)$$

On peut modéliser ce système dans le module Simulink de MATLAB à l'aide des blocs de base **Integrator**, **Gain** et **Sum** de la manière suivante :

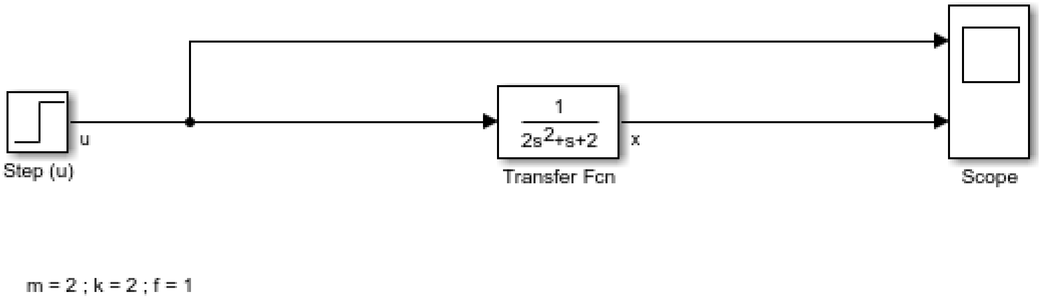


Le bloc **Step** permet de générer une consigne u(t) en échelon. Le bloc **Scope** permet de visualiser l'évolution de variables d'intérêt.

L'oscillateur amorti étant un système dynamique linéaire, on peut aussi le représenter à l'aide d'une fonction de transfert :

$$m.\frac{d^2}{dt^2}x(t) = -k.x(t) - f.\frac{d}{dt}x(t) + u(t) \quad \Leftrightarrow \quad H(s) = \frac{X(s)}{U(s)} = \frac{1}{m.s^2 + f.s + k}$$

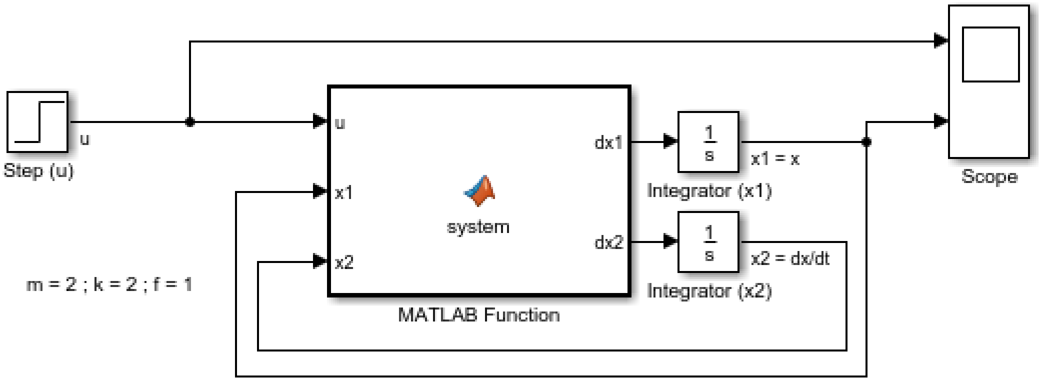
On peut modéliser ce système dans le module Simulink de MATLAB à partir de sa fonction de transfert à l'aide du bloc **Transfer Fcn** de la manière suivante :



L'oscillateur amorti peut également être représenté sous forme d'état :

$$\begin{cases} x_1(t) = x(t) \\ x_2(t) = \frac{d}{dt}x(t) \end{cases} \quad \begin{cases} \frac{d}{dt}x_1(t) = x_2(t) \\ \frac{d}{dt}x_2(t) = -\frac{k}{m}x_1(t) - \frac{f}{m}x_2(t) + \frac{1}{m}.u(t) \end{cases}$$

On peut modéliser ce système dans le module Simulink de MATLAB à partir de sa forme d'état à l'aide des blocs **Integrator** et **MATLAB Function** de la manière suivante :



La fonction d'état est codée dans le bloc **MATLAB Function** :

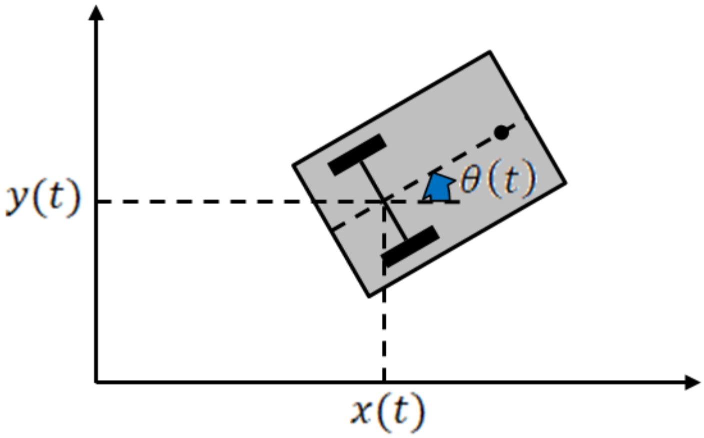
```
function [dx1, dx2] = system(u, x1, x2)

m = 2;
k = 2;
f = 1;

dx1 = x2;
dx2 = -k/m*x1 - f/m*x2 + 1/m*u;
```

EXERCICE 1 – Robot à roue

On considère un robot mobile (tel qu'utilisé dans de nombreux concours de robotique). Ce robot admet sur un même essieu deux roues motorisées de manière indépendante. On note :

- $\big(x(t), y(t)\big)$ les coordonnées cartésiennes du milieu de l'essieu ;
- $\theta(t)$l'angle du robot avec un axe fixe ;
- $\Omega_d(t)$ et $\Omega_g(t)$ les vitesses de rotation des roues droite et gauche.

On suppose que les roues roulent sans glisser ni déraper.



Q1/ Montrer que la dynamique est :

$$\begin{cases} \dfrac{d}{dt}x(t) = v(t).\cos\big(\theta(t)\big) \\ \dfrac{d}{dt}y(t) = v(t).\sin\big(\theta(t)\big) \\ \dfrac{d}{dt}\theta(t) = \omega(t) \end{cases}$$

où $\big(x(t), y(t), \theta(t)\big)$ est l'état et $u(t) = \big(v(t), \omega(t)\big)$ est la commande que l'on relira à $\Omega_d(t)$ et $\Omega_g(t)$, ρ le rayon des roues et l la distance entre les roues.

Q2/ Modéliser dans le module Simulink de MATLAB le robot à roue avec comme commande $\big(v(t), \omega(t)\big)$ et comme sortie l'état $\big(x(t), y(t), \theta(t)\big)$.

Q3/ Simuler la trajectoire du chariot pendant 10s avec un état initial nul et la commande suivante :

$$v(t) = \begin{cases} 1 & \text{si } t > 1s \\ 0 & \text{sinon} \end{cases} \qquad \omega(t) = \begin{cases} 1 & \text{si } t > 3s \\ 0 & \text{sinon} \end{cases}$$

On pourra utiliser le bloc Scope XY Graph pour visualiser la trajectoire dans le plan.

Q4/ Modifier les conditions initiales et les commandes (par exemple en utilisant un bloc Sine Wave) et observer le comportement du système.

Q5/ Quels sont les points d'équilibre du système ? Vérifier en simulation.

On désire suivre une trajectoire de référence rectiligne suivant l'axe des abscisses avec une vitesse constante $a > 0$:

$$\begin{cases} x_r(t) = a.t \\ y_r(t) = 0 \\ \theta_r(t) = 0 \end{cases} \qquad \begin{cases} v_r(t) = a \\ \omega_r(t) = 0 \end{cases}$$

Pour cela, on pose :

$$\begin{cases} x(t) = x_r(t) + \delta x(t) \\ y(t) = y_r(t) + \delta y(t) \\ \theta(t) = \theta_r(t) + \delta\theta(t) \end{cases} \quad \begin{cases} v(t) = v_r(t) + \delta v(t) \\ \omega(t) = \omega_r(t) + \delta\omega(t) \end{cases}$$

où les variables d'écart $\delta x(t)$, $\delta y(t)$, $\delta\theta(t)$, $\delta v(t)$ et $\delta\omega(t)$ sont supposées petites.

Q6/ Vérifier que la trajectoire de référence est bien solution des équations de la dynamique.

Q7/ Montrer qu'au premier ordre, les variables d'écart satisfont la dynamique linéaire suivante :

$$\begin{cases} \frac{d}{dt}\delta x(t) = \delta v(t) \\ \frac{d}{dt}\delta y(t) = a.\delta\theta(t) \\ \frac{d}{dt}\delta\theta(t) = \delta\omega(t) \end{cases}$$

On constate qu'au premier ordre, les équations forment deux systèmes indépendants.

Q8/ Proposer un bouclage d'état qui stabilise la dynamique longitudinale :

$$\frac{d}{dt}\delta x(t) = \delta v(t)$$

Q9/ Proposer un bouclage d'état qui stabilise la dynamique latérale :

$$\begin{cases} \frac{d}{dt}\delta y(t) = a.\delta\theta(t) \\ \frac{d}{dt}\delta\theta(t) = \delta\omega(t) \end{cases}$$

Q10/ Implémenter dans la simulation un bouclage d'état qui stabilise les dynamiques longitudinales et latérales. Vérifier que le système en boucle fermé converge bien vers la trajectoire de référence avec $a = 1$ pour des conditions initiale faiblement en écart avec la trajectoire de référence.

Q11/ Comment se comporte le système bouclé lorsque la vitesse de référence a augmente ? diminue ? Que peut-on proposer pour que le comportement du système bouclé soit similaire quelle que soit la valeur de a ?

EXERCICE 2 – Oscillateur de Van der Pol

On cherche à simuler un oscillateur de Van der Pol. Physiquement, l'oscillateur de Van der Pol est un circuit électrique oscillant avec une résistance variable pouvant être négative, qui peut être modélisé par l'équation suivante :

$$\frac{d^2}{dt^2}x(t) + \rho.(x(t)^2 - 1).\frac{d}{dt}x(t) + x(t) = u(t)$$

Le terme d'amortissement varie ainsi non-linéairement.

On étudie le comportement de l'oscillateur de Van der Pol sans excitation externe. Le terme $u(t)$ est donc supposé nul.

Q1/ **Modéliser dans le module Simulink de MATLAB l'oscillateur de Van der Pol. Utiliser les conditions initiales $\boldsymbol{x(0) = 1}$ et $\boldsymbol{\frac{d}{dt}x(0) = 0,4}$ et le paramètre $\boldsymbol{\rho = 1}$.**

Q2/ **Représenter la trajectoire issue de la condition initiale précédente. On pourra utiliser un temps de simulation de 100 secondes.**

Q3/ **Vers quoi l'état du système converge-t-il ? Quel est le terme dans l'équation de l'oscillateur de Van der Pol qui engendre ce phénomène ?**

Q4/ **Faire varier les conditions initiales. Que constate-t-on ? Aurait-on obtenu la même propriété avec un oscillateur linéaire ?**

On étudie maintenant le comportement de l'oscillateur de Van der Pol avec excitation externe. Le terme $u(t)$ n'est donc plus supposé nul.

Q5/ **Ajouter une perturbation $\boldsymbol{u(t)}$. Que constate-t-on ? Quel genre de terme peut-on rajouter sans risquer de modifier trop la période de l'oscillateur ? Proposer un cas problématique.**

Q6/ **Dupliquer l'oscillateur de Van der Pol. Proposer un schéma pour synchroniser les deux oscillateurs. On pourra par exemple chercher à ralentir ou accélérer l'un des deux systèmes en fonction de mesures. La méthode obtenue est-elle robuste à des perturbations dans les équations des oscillateurs ?**