

## ❖ **SIMULINK**

### **Lancement de Simulink:**

La fenêtre suivante contenant les librairies de simulink, apparaît ainsi qu'une fenêtre de travail.

---

### **Construction d'un modèle dans la fenêtre de travail**

Méthode de placement d'un composant

- on sélectionne une librairie de simulink : double clic pour l'ouvrir (exemple librairie : Linear)
- on sélectionne un composant (exemple Sum):
- on maintient l'appui sur le bouton gauche de la souris
- on fait glisser l'élément dans la fenêtre de travail
- on relâche le bouton.

Exercice : construire l'environnement décrit dans la figure suivante, on indique au-dessus de chaque élément, la librairie d'origine.

---

### **Réalisation des connexions**

Méthode :

- On sélectionne avec la souris, le symbole > situé sur un composant
  - On maintient l'appui sur le bouton et on tire le lien vers un symbole >
  - On peut relâcher le bouton pour changer de direction.
  - On vérifie que la connexion est correcte par le fait que la flèche est accentuée
- 

### **Paramétrage des composants**

Méthode

- On effectue un double clic sur le composant exemple Mux, la fenêtre de paramétrage s'ouvre,
- On tape les valeurs désirées : ici la valeur 2 pour indiquer 2 entrées,
- On ferme cette fenêtre par Close, les nouvelles valeurs sont prises en compte.

## **Exercice**

- paramétrer le générateur sinusoïdal avec une amplitude de 2 et une fréquence de 0.5 Hz
  - le gain sera réglé à 0.5.
  - régler la fréquence de départ du chirp à 0.01 hz
- 

## **Désignation des composants**

Chaque composant possède un nom par défaut exemple Gain, on peut modifier ce nom.

Méthode

- clic sur le nom
  - on tape un nouveau nom
- 

## **Marquage des liens ou connexions**

Méthode

- Double clic sur le lien que l'on veut marquer,
- une fenêtre apparaît qui vous indique le bon déroulement de l'opération. • on tape un nom exemple sin,

Réitérer l'opération pour le chirp

---

## **Propagation des marquages**

Méthode

- à la sortie d'un composant : double clic sur le lien,
  - taper le symbole < puis cliquer hors de cette fenêtre,
  - taper Ctrl D (contrôle D)
- 

## **Renvoi d'un signal et récupération**

Afin de ne pas surcharger le dessin, on peut utiliser 2 composants situés dans la librairie connections qui permettent d'effectuer une transition sans fil. Ces 2 composants s'appellent GOTO et FROM

Le "tag" possède un nom que l'on peut modifier

Chaque

« tag » doit être modifié un par un, attention aux correspondances.

---

## **Personnalisation de la fenêtre de travail**

Il est possible de redimensionner chaque composant

- on le sélectionne,
- on saisit une poignée,
- on étire ou on diminue.

Dans le menu Format de la fenêtre on dispose d'autres commandes (il faut d'abord sélectionner un composant).

- Font permet de choisir le type de caractères
- Flip name : de placer le nom au dessus /en dessous
- Hide name de cacher le nom
- Flip block de retourner le bloc
- Rotate block de le tourner de 90°
- Foreground color de sélectionner une couleur pour le texte
- Background color : de sélectionner une couleur pour le bloc

On peut de même personnaliser les liens ou connexions :

- Wide Vector Lines permet de dimensionner l'épaisseur des liens en fonction du nombre de signaux,
- Line Width permet d'obtenir l'indication du nombre de signaux sur les liens
- Ctrl D permet de mettre à jour tout ceci en cas de modification.

---

## **Modifications**

Modification des composants On peut :

- ajouter un composant à tout moment,
- supprimer un composant en le sélectionnant et touche Suppr,
- Modifier la position d'un composant en le sélectionnant on laisse la touche gauche de la souris appuyée et on le déplace.
- dupliquer un composant : on le sélectionne, on appuie sur la touche Ctrl on faisant glisser le composant.

On peut revenir en arrière de toute opération en utilisant l'icône Undo

déroulement Modification des liens

simulation • en utilisant les poignées situées sur le lien (une fois

sélectionner celles-ci apparaissent),

- en appuyant sur le bouton droit de la souris on ajoute un nouveau départ,
- shift et bouton gauche permet d'ajouter de nouvelles poignées de changement de direction.

## **SIMULATION**

---

### **Paramétrage de la simulation**

La simulation utilise un certain nombre de paramètres : menu simulation → parameters

- instant de départ ( 0 par défaut)
- instant d'arrêt (mettre 20s)

On étudiera ultérieurement les autres paramètres. Faire close ce qui valide les modifications.

### **Lancement de la simulation**

- menu simulation → start
- ou ctrl T
- ou icône >
- Une sonnerie indique la fin de la simulation

Exercice lancer la simulation après avoir ouvert le scope

- On peut effectuer des zooms avant/arrière avec les 3 premiers icônes
- ajuster les axes avec le 4°
- sauver les données avec le 5°
- régler le scope avec le 6°
- imprimer avec le dernier

## **Lien entre Simulink et Matlab**

Pour diverses opérations, il est intéressant de disposer des signaux dans l'environnement de matlab ou de récupérer des signaux définis dans matlab.

---

## Envoi de signaux vers l'environnement de matlab

Les blocs ToWorkspace de la librairie Sinks permettent de diriger les signaux vers l'environnement de matlab dans l'exemple traité jusqu'à présent ceci est réalisé avec le bloc nommé "signaux" sur lequel arrive le tag "de"

Exercice :

Dans la fenêtre de commande matlab, taper `plot(tout,signaux)`  
signaux est une variable contenant les signaux générés tout est une variable que l'on verra plus tard

Récupération de signaux issus de matlab

Le bloc FromWorkspace de la librairie Source permet de définir des signaux dans l'environnement Matlab et de les utiliser dans l'environnement de Simulink.

### Exemple

On définit la variable  $T=0:0.01:20$ ; ainsi que le signal désiré  $U=\sin(2*\pi*t^3)$ ;

Attention les vecteurs T et U doivent être des vecteurs colonnes.

## Simulation d'un Systeme Non Lineaire

**Construction d'un modèle non linéaire** Exemple :  $\dot{x}_1 = x_2 + u \cos(x_1)$   $\dot{x}_2 = u$   $y = x_1$

On commence par créer le système à l'aide du bloc Fcn de la librairie NonLinear

L'entrée de ce bloc est nommée u et peut-être un vecteur Les composantes sont désignées par u(1) u(2) etc...

La fonction correspondant à  $\dot{x}_1$  s'écrit :  $u(3)+u(1)*\cos(u(2))$

Si les entrées sont respectivement u,  $x_1$  et  $x_2$

Ceci est programmé dans la fenêtre de définition de Fcn

On va encapsuler le système précédent dans un sous système, cette méthode permet de rendre plus lisible un schéma et de paramétrer le sous-système.

Méthode

- on sélectionne tous les éléments qui formeront le sous-système

- On tape Ctrl G (ou Edit>Create SubSystem)
- On obtient : un bloc nommé Subsystem que l'on peut renommer, les entrées et sorties peuvent être elle aussi renommées à l'intérieur du bloc (double clic pour ouvrir le bloc).
- Des ports d'entrée et de sortie on été rajoutés automatiquement, on peut en ajouter par exemple le port correspondant à x2
- On va introduire un gain suivant x2 d'une valeur désignée par g Æ edit Æ Mask Subsystem Æ initialization
- En cliquant maintenant sur le soussysteme, on obtient la fenêtre

suivante, dans laquelle on fixe une valeur de g

- On prend en compte cette valeur pour la dérivée de x2 par exemple par Æ edit Æ Look under Mask
- 

## **Modelisation d'un Systeme Lineaire Monovariabile et Simulation**

But : Construire un modèle continu en représentation externe et le simuler.

1 Définition du système :  $H(p) = p_2 + 0.2p_{+1}$

**Format :**

- Une fonction de transfert monovariabile peut être représentée
- Soit par le rapport de 2 polynômes en p (ou s). Dans Matlab, un polynôme se représente par un vecteur de paramètres suivant les puissances décroissantes de p par exemple : le dénominateur de l'exemple ci-dessus se représente par le vecteur : [1 0.2 1]
- Soit par une représentation par zéros et pôles
- Matlab utilise un modèle LTI pour la définition de systèmes

---

## **Création d'un modèle LTI {Linear Time Invariant}**

- Définir le numérateur exemple : >num = 1 ;
- Définir le dénominateur exemple >den = [1 0.2 1] ;
- Construire le système LTI exemple > sys1= tf(num,den) ;
- Faire afficher sys1 en tapant le nom puis sur la touche Return

Ou

- Vecteur définissant les zéros exemple : `> zz=[] %aucun`
- Vecteur définissant les pôles exemple : `> pp=[ -0.1000 + 0.9950i -0.1000-0.9950i]`
- Gain exemple `> k=1`
- Construire le système : `> sys2=zpk(zz,pp,k)`

### **Les propriétés d'un modèle LTI sont obtenues en tapant :**

`> get(sys1)`

On obtient :

`num = {[0 0 1]}`

`den = {[1 0.2 1]}`

`Variable = 's' .....` choix possible {continu : {s p} échantillonné : {z z<sup>-1</sup> q}}

`Ts = 0 .....` 0 / système continu sinon la période d'échantillonnage

`Td = 0 .....` 0 ou la valeur d'un retard sur l'entrée

`InputName = {} .....` on peut donner un nom à l'entrée

`OutputName = {} .....` Idem pour la sortie

`Notes = {}                      UserData = []`