

# Simulink

## 1 – 1<sup>ère</sup> Méthode d'utilisation

### \* Sous Simulink

- saisir le schéma fonctionnel
- saisir les fonctions de transfert sous forme de polynômes
- signaux d'entrée : "**step**"
  - . start time : instant auquel l'échelon est appliqué
  - . stop time : durée de l'échelon
  - . initial value : valeur initiale
  - . final value : valeur finale
  - . sample time : période d'échantillonnage ( $>$ Tolérance ; ex :  $10^{-2} > 10^{-3}$ s)
- relier les sorties à observer à des oscilloscopes : "**scope**"
- saisir les paramètres de la simulation dans "**Simulation → Parameters**"
  - . Tol : tolérance de calcul sur l'état ( $<$ sample time)
- exécuter la simulation avec : "**Simulation → Start**".

## 2 – 2<sup>ème</sup> Méthode d'utilisation

### \* Sous Simulink

- saisir le schéma fonctionnel
- saisir les fonctions de transfert sous la forme : numG, denG, numC, denC
- saisir les paramètres de la simulation dans "**Simulation → Parameters**", en leur donnant un nom
- saisir les paramètres des signaux d'entrée dans les "**step**", en leur donnant un nom.
- enregistrer le schéma en lui donnant un nom du type : "**tnom\_du\_schéma**", le "t" devant le nom permettant de repérer les schémas destinés à des simulations temporelles.

### \* Sous Matlab

- créer un fichier **.m**
- l'enregistrer en lui donnant un nom du type : "**pnom\_du\_schéma**", le "p" devant le nom permettant de référencer le schéma utilisé
- définir les fonctions de transfert numG, denG, numC, denC
- définir les paramètres des signaux d'entrée
- définir les paramètres de la simulation.

### \* Sous Simulink

- exécuter la simulation avec : "**Simulation → start**".

### 3 – 3<sup>ème</sup> Méthode d'utilisation : simulation temporelle

#### \* Sous Simulink

Comme pour la 2<sup>ème</sup> méthode, faire :

- saisir le schéma fonctionnel
- saisir les fonctions de transfert sous la forme : numG, denG, numC, denC
- saisir les paramètres de la simulation dans "**Simulation → Parameters**", en leur donnant un nom
- saisir les paramètres des signaux d'entrée dans les "**step**", en leur donnant un nom.

Puis :

- relier les sorties à observer à des "**ToWorkspace**" à la place des oscilloscopes ; définir :
  - . la taille (donner un nom qui sera défini dans le fichier Matlab)
  - . le type : **matrix**
  - . la période d'échantillonnage ( $> \text{Tolérance}$  ; ex :  $10^{-2} > 10^{-3}\text{s}$ )
- définir le vecteur temps avec une horloge "**clock**" reliée à un "**ToWorkspace**"
- enregistrer le schéma en lui donnant un nom du type : "**tnom\_du\_schéma**", le "t" devant le nom permettant de repérer les schémas destinés à des simulations temporelles.

#### \* Sous Matlab

Comme pour la 2<sup>ème</sup> méthode, faire :

- créer un fichier **.m**
- l'enregistrer en lui donnant un nom du type : "**ptnom\_du\_schéma**", le "p" devant le nom permettant de référencer le schéma utilisé
- définir les fonctions de transfert numG, denG, numC, denC
- définir les paramètres des signaux d'entrée
- définir les paramètres de la simulation.

Puis :

- définir la taille des "**ToWorkspace**"
- pour exécuter la simulation à partir du fichier Matlab, faire :  
**[x,y,z] = sim('tnom\_du\_schéma') ;**  
( la nature de x, y, z sera spécifiée après)
- prévoir de tracer les figures avec plot :  
**figure(1);**  
**plot(t,s);**
- exécuter la simulation en exécutant le fichier .m sous Matlab.

#### 4 – 4<sup>ème</sup> Méthode d'utilisation : simulation fréquentielle

##### \* Sous Simulink

Comme pour la 2<sup>ème</sup> méthode, faire :

- saisir le schéma fonctionnel
- saisir les fonctions de transfert sous la forme : numG, denG, numC, denC
- saisir les paramètres de la simulation dans "**Simulation → Parameters**", en leur donnant un nom

Puis :

- remplacer les "**step**" par des "**in1**" (bibliothèque Signals&Systems)
- relier les sorties à observer à des "**out1**" à la place des oscilloscopes
- supprimer l'horloge
- enregistrer le schéma en lui donnant un nom du type : "**fnom\_du\_schéma**", le "f" devant le nom permettant de repérer les schémas destinés à des simulations fréquentielles.

##### \* Sous Matlab

Comme pour la 2<sup>ème</sup> méthode, faire :

- créer un fichier **.m**
- l'enregistrer en lui donnant un nom du type : "**pfnom\_du\_schéma**", le "p" devant le nom permettant de référencer le schéma utilisé
- définir les fonctions de transfert numG, denG, numC, denC
- définir les paramètres des signaux d'entrée
- définir les paramètres de la simulation.

Puis :

- définir un vecteur pulsations :  **$\omega = \text{logspace}(p1, p2, N);$**
- pour exécuter la simulation à partir du fichier Matlab, faire :  
 **$[a, b, c, d] = \text{linmod}('fnom\_du\_schéma');$**   
où a, b, c, d sont les matrices d'état
- pour tracer un diagramme de Bode en multivariable :

**$[g, p] = \text{bode}(a, b, c, d, i, \omega);$**  (tous les diagrammes de Bode par rapport à l'entrée n°i)  
 **$gdB = 20 * \log_{10}(g);$**

...

**$\text{semilogx}(\omega, gdB(:, j));$**  (diagrammes de gain par rapport à la sortie n°j)

...

**$\text{semilogx}(\omega, p(:, j));$**  (diagrammes de phase par rapport à la sortie n°j).

- exécuter la simulation en exécutant le fichier .m sous Matlab.

## 5 – 5<sup>ème</sup> Méthode d'utilisation : simulations temporelle et fréquentielle avec un seul schéma Simulink et un seul fichier Matlab

### \* Sous Simulink

Comme pour la 2<sup>ème</sup> méthode, faire :

- saisir le schéma fonctionnel
- saisir les fonctions de transfert sous la forme : numG, denG, numC, denC
- saisir les paramètres de la simulation dans "**Simulation → Parameters**", en leur donnant un nom

Puis :

- *ajouter* aux "**step**" des "**in1**" (bibliothèque Signals&Systems) (mettre 3 entrées aux comparateurs)
- relier les sorties à observer à des "**out1**" à la place des oscilloscopes
- supprimer l'horloge
- enregistrer le schéma en lui donnant un nom du type : "**tfnom\_du\_schéma**", le "tf" devant le nom permettant de repérer les schémas destinés à des simulations temporelles et fréquentielles.

### \* Sous Matlab

Comme pour la 4<sup>ème</sup> méthode, faire :

- créer un fichier **.m**
- l'enregistrer en lui donnant un nom du type : "**ptfnom\_du\_schéma**", le "p" devant le nom permettant de référencer le schéma utilisé
- définir les fonctions de transfert numG, denG, numC, denC
- définir les paramètres des signaux d'entrée
- définir les paramètres de la simulation.

Puis :

- définir un vecteur pulsations :  **$\omega = \text{logspace}(p1, p2, N);$**
- pour exécuter la simulation à partir du fichier Matlab, faire :  
 **$[a, b, c, d] = \text{linmod}('tfnom\_du\_schéma');$**   
où a, b, c, d sont les matrices d'état
- pour tracer un diagramme de Bode en multivariable :  
 **$[g, p] = \text{bode}(a, b, c, d, i, \omega);$**  (tous les diagrammes de Bode par rapport à l'entrée n°i)  
 **$gdB = 20 * \log_{10}(g);$**   
...  
 **$\text{semilogx}(\omega, gdB(:, j));$**  (diagrammes de gain par rapport à la sortie n°j)  
...  
 **$\text{semilogx}(\omega, p(:, j));$**  (diagrammes de phase par rapport à la sortie n°j)
- la simulation temporelle, faire :  
 **$[t, x, y] = \text{sim}('tfnom\_du\_schéma');$**   
où t est le vecteur temps, x l'évolution de l'état et y le vecteur des outputs
- pour tracer une courbe temporelle, faire :  
 **$\text{plot}(t, y(:, k));$**  où y(:, k) est le output n° k
- exécuter la simulation en exécutant le fichier .m sous Matlab.

### **Intérêts de cette méthode :**

- plus d'horloge
- plus de taille de "ToWorkspace" à définir
- simulations temporelle et fréquentielle avec un seul schéma Simulink et un seul fichier Matlab.