

## **INITIATION AU MODULE SCICOS DU LANGAGE SCILAB**

Scilab est un langage scientifique permettant d'effectuer des calculs numériques. C'est un logiciel libre développé par l'I.N.R.I.A. et par ces utilisateurs qui peuvent apporter leur contribution à son développement. Ce logiciel est analogue à MATLAB. Tout comme celui-ci il contient un module de programmation graphique. Celui-ci s'appelle SCICOS et permet de programmer de façon assez simple les calculs liés à bon nombre de problèmes scientifiques. Il permet notamment une résolution numérique des systèmes différentiels.

La suite va vous présenter succinctement ce module et une petite partie de ses possibilités.

### **1. PRINCIPE DE FONCTIONNEMENT DE SCICOS :**

La programmation graphique permet de gagner du temps dans la mesure où il n'y a plus à traduire un algorithme en langage de programmation.

Une fois le « schéma » de calcul établi, il n'y a qu'à le disposer correctement dans la fenêtre de programmation graphique.

#### **1.1. Les outils de programmation : les blocs**

Pour réaliser les opérations de calcul sur les données ainsi que différentes tâches (lecture et écriture dans des fichiers, affichage graphique,...), l'utilisateur dispose de blocs (qui sont en fait des petits programmes) qu'il doit relier entre eux pour réaliser son programme.

Les blocs disposent de 2 types entrées et sorties :

- les entrées et sorties de données : elles se situent à gauche des blocs pour les entrées et à droite pour les sorties. Elles sont représentées par des triangles noirs.
- Les entrées et sorties d'évènements : elles sont situées en haut des blocs pour les entrées et en bas pour les sorties. Elles sont représentées par des triangles rouges.

Les blocs ne comportent pas systématiquement tous ces types d'entrées et sorties (rarement même).

Les premières permettent d'acheminer des données vers le bloc et de les envoyer vers le bloc suivant une fois son exécution terminée.

Les suivantes servent à « cadencer » le travail des blocs pour autoriser ou non leur exécution, pour déclencher la suite d'un programme ou son arrêt, pour gérer la fréquence d'un affichage, ...

#### **1.2. Les principaux blocs de scicos :**

Les blocs sont répartis dans des palettes auxquelles on accède par le menu « edit ». Ces palettes sont au nombre de 7 :

- « Inpouts/Outputs » : regroupe les blocs permettant de gérer les entrées et sorties de données dans le diagramme. Par exemple en entrant des constantes pour les paramètres du problème ou en sortant un graphique pour illustrer la solution numérique trouvée.
- « Linear » : regroupe les blocs permettant d'effectuer des opérations linéaires (addition de données, transformée de Laplace,...).
- « Non Linear » : regroupe les blocs permettant d'effectuer des opérations non linéaires (multiplication de données, opérations trigonométriques,...).
- « Events » : regroupe les blocs permettant de gérer les évènements pour une bonne exécution du programme.

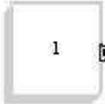
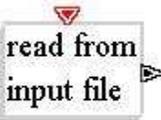
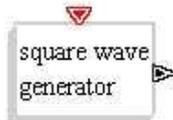
...

Les principaux outils sont décrits dans les pages suivantes :

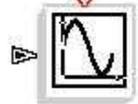
**aidelnOut** Diagram Edit Simulate Object Misc stop

### Aide sommaire sur les principaux outils de la palette Inputs - Outputs.

#### Entrées:

	<p>Permet de générer une constante. Elle peut être désignée par une lettre si elle est définie dans le "context" (menu "edit").</p>		<p>Générateur de temps. Son incrémentation est liée au paramétrage de la simulation.</p>
	<p>Générateur d'événements (commande de blocs). On fixe la période et la date initiale.</p>		<p>Lit les données dans un fichier. Le format (7(e10.3,1x)) par défaut correspond au .txt. Il faut fournir le chemin complet du fichier.</p>
	<p>Générateur de valeurs aléatoires. On fixe la moyenne et la déviation ou le minimum et l'écart max.</p>		<p>Génère une dent de scie. La période est fixée par le générateur d'événements et la tension maximale a même valeur que la période.</p>
	<p>Générateur de signal carré [-A;+A]. On donne A l'amplitude. La période est liée au générateur d'événements.</p>		<p>Génère une sinusoïde dont on donne l'amplitude, la fréquence et la phase.</p>

#### Sorties:

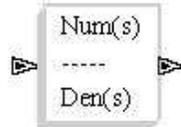
	<p>Permet de stocker les données dans un fichier. On peut choisir le format (idem read from file). Il faut préciser le chemin et le nom du fichier.</p>		<p>Génère un affichage numérique de la valeur en entrée de données.</p>
	<p>Oscilloscope à 2voies. On peut choisir la fenêtre d'affichage, le type de courbe et les échelles.</p>		<p>Afficheur <math>Y=f(X)</math>. Y est sur l'entrée haute et X est sur l'entrée basse. On peut choisir la fenêtre d'affichage, le type de courbe et les échelles.</p>
	<p>Afficheur <math>X=f(t)</math>. X peut être un vecteur. On peut choisir la fenêtre d'affichage, le type de courbe et les échelles.</p>		<p>Poubelle. Permet de terminer une "ligne" de données qui n'a plus d'utilité.</p>

Click to open block or make a link

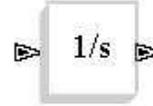
Démarrer Scilab Others aidelnOut 15:50

### Aide sommaire sur les outils des palettes "Linear" et "Events":

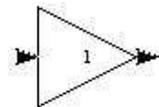
#### Linear:



Ce bloc réalise la transformée de fourier du signal d'entrée. En entrant  $Num(s)=1$  et  $Den(s)=s$  on obtient un intégrateur. La condition initiale doit être ajoutée avec une constante et un sommateur après la sortie du bloc.



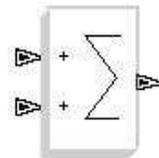
Bloc permettant de réaliser l'intégration du signal d'entrée. On entre la condition initiale sur la primitive. Il est préférable d'utiliser le bloc précédent en raison d'anomalies de déclenchement.



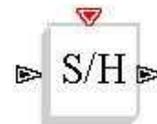
Ce bloc multiplie le signal d'entrée par une constante. Elle peut être sous forme d'un nom si elle est définie le "context" de la simulation (menu "edit").



Réalise la somme de deux entrées.



Ce bloc réalise la somme pondérée des entrées. Les coefficients sont rentrés sous forme d'un vecteur. Le nombre d'entrées s'ajuste automatiquement.



Attend l'arrivée d'un événement pour sortir le signal arrivé sur l'entrée.

#### Events:



Génère un événement à l'instant t choisi.



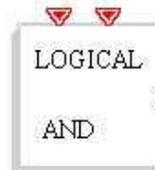
Retarde le passage de l'évènement de l'entrée vers la sortie. On paramètre le retard ainsi que la date de début de fonctionnement.



On entre une constante n. Le bloc génère un événement 1/n fois sur la sortie 1 et (n-1)/n fois sur la sortie 2.



Stoppe l'exécution du programme s'il est stimulé.



Génère +1 ou -1 en sortie selon le résultat de l'opération logique "et" sur les deux événements d'entrée.

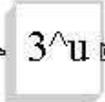
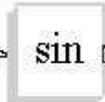
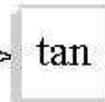
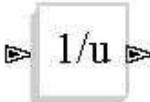
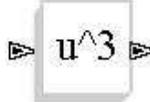
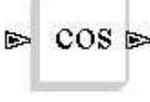
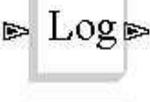
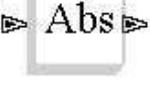


Effectue l'opération logique "et" sur les événements en entrée pour en générer un en sortie.



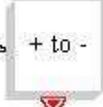
On entre la période de base T et un facteur n. Le bloc génère un événement toutes les nTsec sur la sortie 1 et toutes les T sec sur la sortie2.

## Aide sommaire sur les outils de la palette "Non Linear":

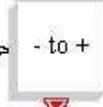
-  Multiplie les 2 signaux d'entrée.
-  Elève une constante (paramètre) à la puissance donnée par la valeur du signal d'entrée.
-  Calcule le sinus du signal d'entrée.
-  Calcule la tangente du signal d'entrée.
-  Permet de générer une saturation du signal en entrant les valeurs max et min ainsi que la pente entre les deux.
-  Sort la valeur minimale des coordonnées du vecteur en entrée.
-  Donne une sortie Y fonction de l'entrée X obtenue par interpolation linéaire à partir de deux vecteurs donnés en paramètres: 1 pour les valeurs de X et 1 pour celles de Y correspondantes.
-  Calcule l'inverse du signal d'entrée.
-  Elève le signal à la puissance n que l'on paramètre.
-  Calcule le cosinus du signal d'entrée.
-  Calcule le logarithme du signal d'entrée.
-  Sort la valeur absolue du signal d'entrée.
-  Sort la valeur maximale des coordonnées du vecteur en entrée.
-  Permet de réaliser une interpolation bilinéaire sur les deux entrées X et Y pour donner une sortie Z fonction de X et Y. On donne trois vecteurs: 1 pour les valeurs de X, 1 pour celles de Y et 1 pour celles de Z correspondantes.

## Aide sommaire sur les outils des palettes "Treshold", "Others" et "Branching":

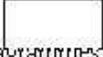
### Treshold:

 Génère un événement si l'entrée passe par 0 selon une pente négative.

 Génère un événement si toutes les entrées passent par 0 en même temps. Le nombre d'entrées est paramétrable.

 Génère un événement si l'entrée passe par 0 selon une pente positive.

### Others:

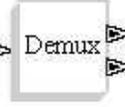
 Permet d'introduire des commentaires dans les diagrammes.

### Branching:

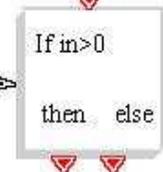
 Transforme un ensemble de n entrées (paramétrable de 1 à 8) en un vecteur.

 Lorsque ce bloc reçoit un événement sur une de ses entrées rouges, il renvoie en sortie le signal correspondant à l'entrée noire de même numéro.

 Ajoute des événements.

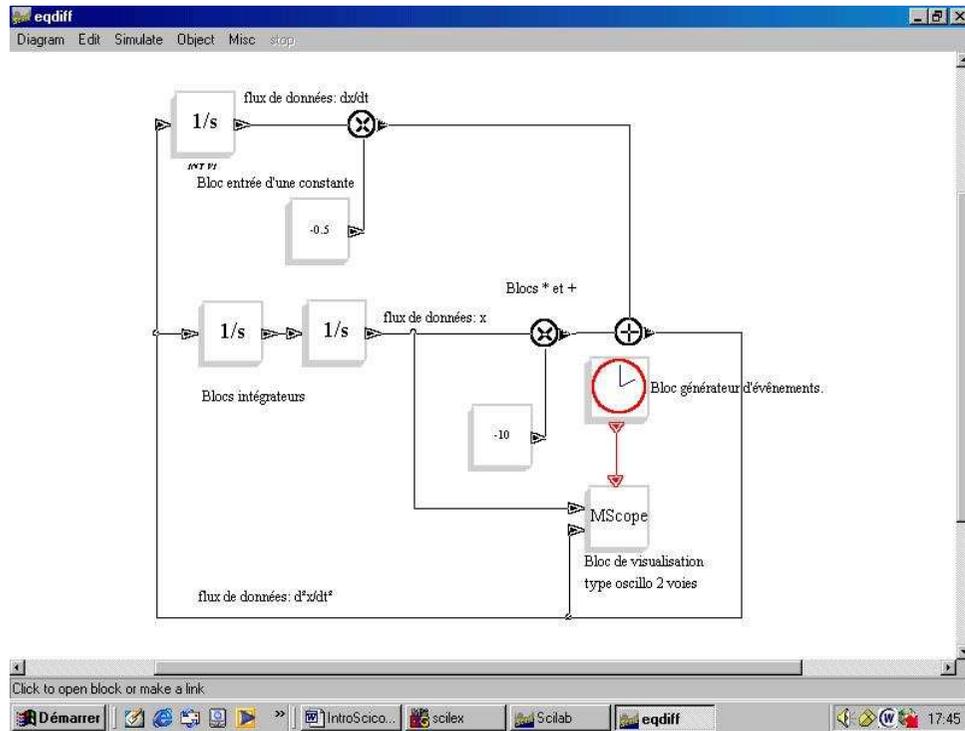
 Réalise l'opération inverse du bloc précédent.

 Lorsque ce bloc reçoit un événement sur une de ses entrées rouges, il renvoie le signal d'entrée sur la sortie de même numéro.

 Le bloc génère un événement sur la sortie 1 si l'entrée est positive et sur la sortie 2 si elle est négative.

### 1.4. Un exemple de diagramme :

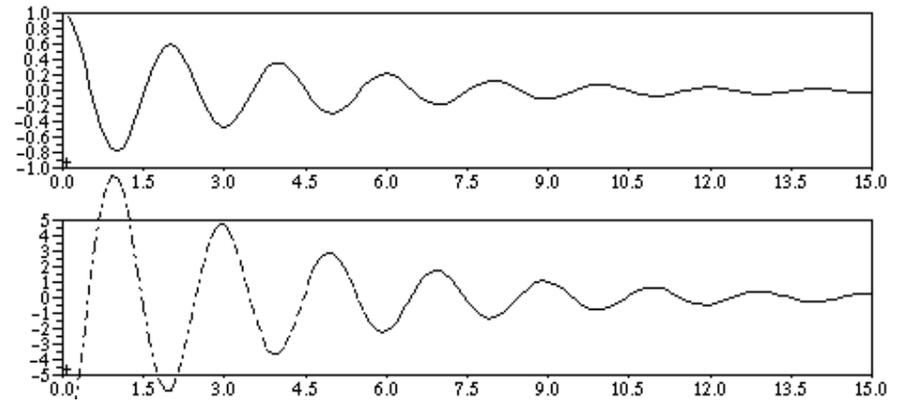
Le diagramme ci-dessous représente le programme permettant de modéliser un oscillateur libre amorti.



La construction de ce diagramme sera détaillée au paragraphe suivant. On peut remarquer les différents blocs qui permettent de réaliser les fonctions d'intégration, ainsi que les opérations arithmétiques (additions et multiplications). Les blocs d'entrée de constante permettent de jouer sur les paramètres de l'oscillateur.

Les blocs sont reliés entre eux par des fils représentant des flux de données. On remarque que le flux représentant la dérivée seconde de la grandeur oscillante ( $d^2x/dt^2$ ) permet d'obtenir la dérivée première de  $x$  en utilisant un bloc intégrateur puis  $x$  en utilisant un second bloc intégrateur.

Les données correspondant à l'accélération et la position de l'oscillateur sont alors envoyées vers un oscilloscope. On remarque qu'un générateur d'événements permet de gérer l'affichage de l'oscilloscope. Le résultat est affiché dans une fenêtre graphique représentée ci-dessous.



D'autres options sont possibles comme par exemple la sauvegarde des valeurs dans un fichier texte qui peut être lu et exploité sous un tableur comme Excel ou sous un logiciel de modélisation numérique tel que Curexper ou autre.

Pour accéder au paramétrage des blocs il suffit de cliquer dessus. Une fenêtre de dialogue apparaît alors avec les différents paramètres ajustables. Les constantes peuvent être entrées sous la forme d'un nom à condition de le définir dans la rubrique « context » du menu « edit ».

Pour une exécution correcte de la simulation il est important de bien paramétrer la simulation dans la rubrique « setup » du menu « simulate ». La durée de la simulation, son pas d'intégration et d'autres facteurs peuvent y être ajustés. Le pas d'intégration est primordial dans la simulation numérique : trop grand il amène à des « solutions » qui ne correspondent pas du tout à la réalité de l'expérience et trop petit il augmente inutilement la durée des calculs.

### 1.5. Les menus de scicos :

**Le module scicos se lance depuis la fenêtre initiale de scilab en tapant scicos().**

Une nouvelle fenêtre s'ouvre alors sur un diagramme vide. La fenêtre propose 6 menus principaux dans son bandeau supérieur. Ceux-ci permettent d'accéder aux différentes fonctions du module scicos du logiciel.

#### 1.5.1. Le menu « Diagram »

Celui-ci regroupe de nombreuses fonctions de base pour travailler sur les diagrammes. C'est un équivalent du classique menu « fichiers » des logiciels sous windows.

On y trouve, entre autres, les fonctions suivantes :

- ouverture d'un nouveau diagramme : **New**
- sauvegarde d'un diagramme : **Save**
- sauvegarde sous un nom spécifique : **Save as**
- chargement d'un diagramme : **Load**
- retracé en cas de bug d'affichage : **Replot**
- sauvegarde dans une fenêtre graphique pour impression : **Export** (puis **Graphic Window**)
- sortie du module scicos vers scilab : **Quit**

#### 1.5.2. Le menu « Edit » :

Il permet d'accéder aux fonctions d'édition classiques ainsi qu'à celles de gestion de la présentation du diagramme et il permet également d'accéder aux palettes d'outils décrites précédemment.

Ses rubriques principales sont :

- **Palettes** : chargement des palettes d'outils. Après avoir sélectionné cette option une fenêtre de dialogue vous permet de sélectionner l'une des 7 palettes. Il suffit de cliquer sur celle de votre choix puis sur OK. Elle apparaît alors à

l'écran. Lorsque vous cliquez sur un bloc, la palette part en barre des tâches et le diagramme principal redevient actif. Il ne reste plus qu'à cliquer à l'endroit où vous voulez placer le bloc pour qu'il y apparaisse.

- **Context** : édition du contexte de la simulation. On y définit les valeurs des constantes rentrées sous forme de lettre dans les blocs du programme.
- **Smart Move** : permet de sélectionner un bloc et de le déplacer sur le diagramme. Après le premier déplacement, la fonction est automatiquement désactivée.
- **Move** : permet de sélectionner un bloc et de le déplacer sur le diagramme. La fonction reste alors active tant que l'on n'en sélectionne pas une autre.
- **Copy** : permet de sélectionner un bloc et de le copier. La copie aura les mêmes caractéristiques que le modèle, même si celles-ci sont modifiées à posteriori.
- **Copy Region** : permet de sélectionner un ensemble de blocs et de le copier. La copie aura les mêmes caractéristiques que le modèle, même si celles-ci sont modifiées à posteriori. La sélection se fait en cliquant sur le diagramme et en développant à la souris un rectangle qui englobe les éléments à copier.
- **Link** : permet de relier les blocs entre eux. En cliquant à un endroit quelconque sur un lien on crée une ramification qui permet d'envoyer les données vers plusieurs blocs simultanément.
- **Delete** : permet d'effacer un bloc ou un lien sélectionné par un clic de souris. La fonction est automatiquement annulée après la sélection.
- **Delete Region** : permet d'effacer un ensemble de blocs et de liens sélectionnés en développant à la souris un rectangle qui englobe les éléments à effacer la fonction est désactivée après la sélection.
- **Undo** : annule la dernière action.

### 1.5.3. Le menu « Simulate » :

Ce menu regroupe les fonctions de paramétrage et d'exécution du programme réalisé dans le diagramme.

- **Setup** : permet de paramétrer la simulation. On y gère notamment la durée de la simulation, la précision des valeurs, le pas d'intégration,...
- **Run** : lance l'exécution du programme et active le menu Stop.
- **Compile** : permet de compiler le programme graphique. La compilation se fait automatiquement lorsque l'on sauve le diagramme ou lorsqu'on lance l'exécution. Toutefois il se peut qu'une anomalie survienne et l'on peut alors forcer la compilation avec cette fonction.

### 1.5.4. Le menu « Object » :

Il regroupe les opérations que l'on peut réaliser sur les objets. Aussi bien le réglage des paramètres associés que leur dimensionnement, leur couleur, ...

Les fonction principales sont les suivantes :

- **Open/set** : permet d'accéder au paramétrage des blocs. Cette fonction est active par défaut.
- **Resize** : permet de redimensionner la taille des blocs à l'affichage.
- **Icon** : permet de modifier l'icône associée aux blocs.
- **Color** : permet de colorier les blocs. Cela peut être intéressant dans un diagramme complexe de colorier les différentes fonctions ou les opérateurs associés à différentes lignes de calcul de sorte qu'on les distingue plus facilement.
- **Label** : permet d'associer une étiquette aux différents blocs.
- **Identification** : ajoute une identification sous l'icône des blocs.
- **Documentation** : permet d'accéder à la documentation en ligne associée à un bloc s'il y en a une particulière.

### 1.5.5. Le menu « Misc » :

Ce menu donne accès aux fonctions permettant de gérer les affichages de la fenêtre du diagramme, les polices, les tailles, les couleurs,...

On y trouve, entre autres, les fonctions suivantes :

- **Window** : permet de gérer la taille de la fenêtre du diagramme.
- **Background color** : permet de choisir la couleur du fond et de l'avant – plan.
- **Default link color** : permet de choisir les couleurs associées aux liens de données et d'évènements.
- **ID fonts** : permet de gérer les tailles et polices des différents caractères affichés.
- **Help** : permet d'obtenir l'aide associée en cliquant sur un bloc. Cette fonction est très utile et donne de bonnes réponses aux questions que l'on peut se poser sur les différents blocs lorsqu'on réalise les diagrammes...surtout quand ils ne donnent pas le résultat espéré.

### 1.5.6. Le menu « stop » :

Ce menu ne comporte qu'une seule fonction : **stop**.

Elle permet de stopper l'exécution d'un programme avant la fin du temps d'intégration initialement prévu. Cette fonction est très utile lorsqu'on oublie de paramétrer la simulation par la rubrique setup du menu Simulate.

En effet, le temps d'intégration par défaut est de 100 000 s....

## 2. REALISATION D'UN DIAGRAMME :

L'exemple traité est le très classique cas d'un oscillateur libre amorti. La simulation doit permettre de choisir la pulsation propre et le facteur de qualité ainsi que la valeur initiale de la grandeur oscillante et de sa dérivée par rapport au temps.

### 2.1. Analyse du problème :

La méthode proposée n'est pas la seule mais simplement une façon de travailler parmi d'autres.

#### 2.1.1. 1<sup>ère</sup> étape :

Ecrire l'équation qui régit les oscillations :  $\frac{d^2x}{dt^2} + \frac{\omega_o}{Q} \frac{dx}{dt} + \omega_o^2 x = 0$

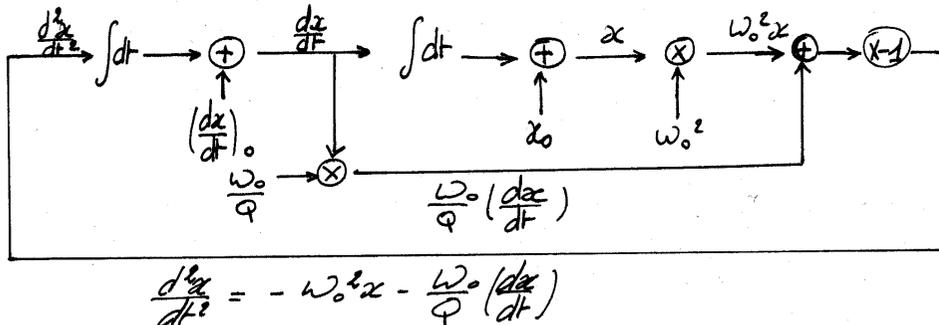
#### 2.1.2. 2<sup>ème</sup> étape :

Recenser les paramètres du problème :

- $\omega_o$  et  $Q$  pour l'oscillateur
- $x_o$  et  $v_o (=dx/dt)_o$  pour les conditions initiales

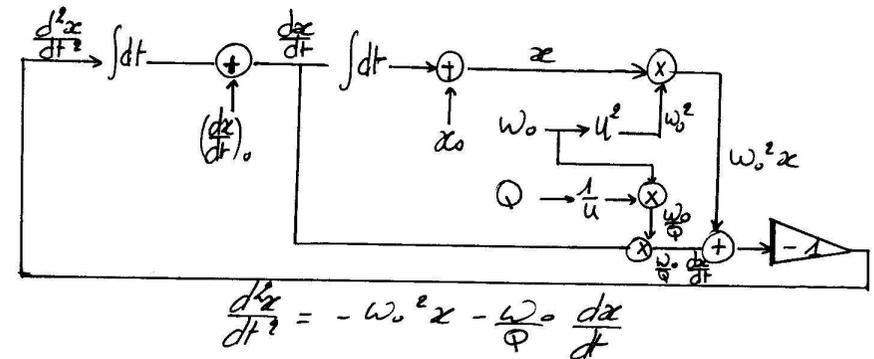
#### 2.1.3. 3<sup>ème</sup> étape :

Ecrire un premier schéma sur papier pour représenter l'équation différentielle.



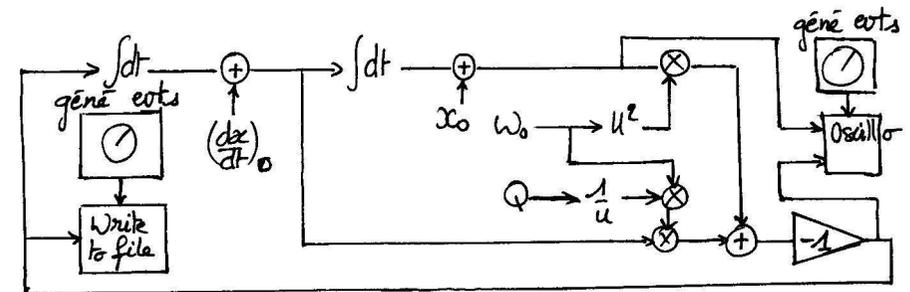
#### 2.1.4. 4<sup>ème</sup> étape :

Affiner le schéma pour introduire les paramètres de la simulation par des blocs d'entrée de constante. Cela nécessite l'ajout d'opérations sur ces données.



#### 2.1.5. 5<sup>ème</sup> étape :

Ajouter les sorties de données vers un fichier ou vers un dispositif d'affichage.



Une fois le schéma sur papier complété et correctement agencé, il faut maintenant écrire le programme dans la fenêtre graphique.

## 2.2. Ecriture du programme :

Le diagramme réalisé est représenté en page 11.

### 2.2.1. Renseignement du « contexte » :

Dans le menu **Edit**, cliquer sur la rubrique **Context**. Il apparaît alors une fenêtre de dialogue dans laquelle on peut introduire données à entrer dans la simulation. On tapera, par exemple :

**Wo=10** ( pulsation propre de 10 rad/s)  
**Q=5** ( facteur de qualité de 5)  
**Xo=0.1** ( valeur initiale de la grandeur oscillante de 0,1 SI)  
**Vo=0** ( valeur initiale de sa dérivée par rapport au temps)

### 2.2.2. Création des entrées de données :

Dans le menu **Edit**, sélectionner la rubrique **Palettes**. Choisir alors celle nommée **Inputs/Outputs** et valider.

La palette d'outils apparaît à l'écran et vous pouvez alors sélectionner le **bloc d'entrée de constante** (cf. p 2, 1<sup>er</sup> outil de la 1<sup>ère</sup> colonne) en cliquant dessus.

La palette est alors envoyée en barre des tâches et il suffit de cliquer à l'endroit où l'on veut placer le bloc dans le diagramme pour qu'il apparaisse. En cliquant dessus à nouveau on peut entrer la constante désirée. Par exemple on tape Wo. Comme elle est définie dans le contexte de la simulation sa valeur lui sera automatiquement affectée.

Recommencer l'opération pour les trois autres paramètres.

A l'aide des outils **Move** ou **Smart Move** (menu **edit**), on peut les disposer à l'endroit voulu sur le diagramme.

### 2.2.3. Création des opérations sur les données :

Dans le menu **Edit**, sélectionner la rubrique **Palettes**. Choisir alors celle nommée **Non linear** et valider.

La palette d'outils apparaît à l'écran et vous pouvez alors sélectionner le **bloc d'inversion de donnée** (cf. p 4, 1<sup>er</sup> outil de la 2<sup>ème</sup> colonne) et le disposer sur le diagramme.

Activer la palette dans la barre des tâches et sélectionner le **bloc d'élévation à la puissance n** (cf. p 4, 2<sup>ème</sup> outil de la 2<sup>ème</sup> colonne) pour le placer sur le diagramme. Cliquer dessus et taper 2 pour la puissance.

### 2.2.4. Création des intégrateurs :

Dans le menu **Edit**, sélectionner la rubrique **Palettes**. Choisir alors celle nommée **Linear** et valider.

La palette d'outils apparaît à l'écran et vous pouvez alors sélectionner le **bloc de transformation de Fourier** (cf. p 3, 1<sup>er</sup> outil de la 1<sup>ère</sup> colonne) et le disposer sur le diagramme. Cliquer dessus et sélectionner 1 pour le numérateur et s pour le dénominateur. On réalise ainsi un intégrateur.

Recommencer l'opération pour en placer un second.

### 2.2.5. Création des additions et multiplications :

Dans la palette **Linear**, sélectionner le **bloc d'addition** (cf. p 3, 2<sup>ème</sup> outil de la 2<sup>ème</sup> colonne) et le placer sur le diagramme.

Recommencer l'opération pour placer les deux autres blocs d'addition nécessaires.

Dans la même palette, sélectionner le **bloc amplificateur** (cf. p 3, 2<sup>ème</sup> outil de la 1<sup>ère</sup> colonne), le placer sur le diagramme et cliquer dessus pour accéder à son paramétrage. Choisir -1 pour le gain.

Dans la palette **Non Linear**, sélectionner le **bloc de multiplication** (cf. p 4, 1<sup>er</sup> outil de la 1<sup>ère</sup> colonne) et le placer sur le diagramme.

Recommencer l'opération pour placer les deux autres blocs d'addition nécessaires.

### 2.2.6. Création des sorties :

Dans la palette Inputs/Outputs, sélectionner le **bloc Mscope** (cf. p 2, 6<sup>ème</sup> outil de la 1<sup>ère</sup> colonne). Le disposer sur le graphique et cliquer dessus pour le paramétrer. Dans un premier temps on pourra laisser les paramètres par défaut. C'est en cliquant sur ce bloc que l'on peut choisir les couleurs et style d'affichage, la fenêtre, sa position, sa taille, les échelles, ...

Pour fonctionner correctement l'oscilloscope a besoin d'un **générateur d'évènements**. Celui-ci est disponible dans la même palette (cf. p 2, 2<sup>ème</sup> outil de la 1<sup>ère</sup> colonne). On paramètre la période de génération des évènements (laisser 0.1s dans un premier temps) et la date du premier événement généré (mettre 0).

Dans la palette Inputs/Outputs, sélectionner le **bloc Write to output file** (cf. p 2, 5<sup>ème</sup> outil de la 1<sup>ère</sup> colonne). Le disposer sur le graphique et cliquer dessus pour le paramétrer. Dans la première cas on entre la taille de l'entrée (1 pour une donnée simple, sa taille pour un vecteur) et dans la deuxième le chemin complet et le nom du fichier de sauvegarde (par exemple c:\windows\bureau\oscillateur.txt). Le format par défaut est reconnu par les éditeurs de texte.

Pour fonctionner correctement, ce bloc a également besoin d'un **générateur d'évènements**. Les données sont sauvegardées avec la même période que celle du générateur d'évènements.

### 2.2.7. Création des liens :

Il ne reste plus qu'à relier les blocs entre eux pour obtenir un diagramme fonctionnel. Pour cela cliquer sur la sortie d'un bloc et tirer le fil de données ou d'évènement jusqu'à l'entrée du bloc suivant.

Si le fil ne se crée pas, sélectionner la fonction **Link** dans le menu **Edit**. Vous pouvez alors ajuster le placement des blocs pour obtenir un diagramme plus propre et lisible.

### 2.2.8. Paramétrage de la simulation :

Dans le menu **Simulate**, choisir la rubrique **Setup**. On y gère les paramètres de la simulation. Dans un premier temps on se contentera de modifier le temps de simulation (**final integration time**) : 100 000 s étant un temps un peu trop long. Prendre un temps raisonnable (30, 60s, ...). Il sera ajusté en fonction des premiers résultats.

### 2.2.9. Lancement de la simulation :

Pour lancer la simulation il suffit de choisir l'option **Run** du menu **Simulate**.

Le diagramme réalisé et diverses courbes obtenues en jouant sur les paramètres de l'oscillateur (context), de la simulation (setup) et des sorties sont présentés dans les pages suivantes.

# Introduction à SCICOS

