Découverte du langage graphique SCICOS
Simulation d'oscillateurs unidimensionnels

par **Claire ALEXANDRE** et **Frédéric WILLOT**
Lycée Gustave Eiffel – 59427 Armentières Cedex
calexandr@infonie.fr
fredpro.willot@cegetel.net

Le langage graphique SCICOS est une partie intégrante du langage scientifique SCILAB. Ce logiciel libre développé et distribué gratuitement par l'I.N.R.I.A, fonctionne sous des environnements très divers (Windows, Linux, Mac OS, …). Il permet de programmer de façon simple des calculs complexes grâce à ses nombreuses fonctions mathématiques et graphiques très avancées, sa capacité à reconnaître des variables spécifiques aux applications scientifiques (comme les vecteurs et les matrices) et une syntaxe simple. Il évolue constamment grâce à une grande communauté d'utilisateurs qui peuvent apporter leur contribution à son développement. Il a atteint aujourd'hui un haut niveau de performances et est utilisé par des industriels tels que RENAULT.

Ce logiciel est analogue à MATLAB et procède par calcul numérique et non par calcul analytique. Tout comme celui-ci il contient un module de programmation graphique : SCICOS. SCICOS permet de programmer de façon très simple les calculs liés à bon nombre de problèmes scientifiques. **Il permet notamment une résolution numérique des systèmes différentiels ayant ou non une solution analytique.**

Cet article vous présentera, dans une première partie, des exemples de problèmes pouvant être simulés sous SCICOS, pour vous donner une idée des ses possibilités, ainsi que les principes de base de ce langage. La deuxième partie expose le principe de résolution numérique d'une équation différentielle et détaille la simulation d'un oscillateur harmonique.

I Exemples d'applications et principes de SCICOS

I.1 Exemples d'applications :

Dans les programmes de l'enseignement post-bac (et même « pré-bac ») les exemples d'applications que l'on peut traiter sous SCICOS sont nombreux :

Mécanique :
- Oscillateurs libres uni ou multidimensionnels avec ou sans frottements (fluides,visqueux,…).
- Oscillateurs forcés (excitation sinusoïdale, triangulaire, …).
- Mouvements dans les champs de forces.

Electrocinétique :
- Régimes transitoires des circuits linéaires et non linéaires.
- Régimes forcés des circuits linéaires.

Chimie :
- Evolution des concentrations lors d'une réaction chimique.
- …

Et d'autres auxquels vous ne manquerez pas de penser.

Sur le site du lycée Gustave Eiffel d'Armentières (http://www2.ac-lille.fr/eiffel/CPGE/index.htm) vous trouverez :
- une notice d'initiation au module de programmation SCICOS,
- des simulations d'oscillateurs, de tirs balistiques et de cinétique chimique,
- des supports de séances réalisées en BTS T.P.I.L. et en P.T.S.I. sur la simulation des oscillateurs.

Vous trouverez ci-dessous quelques résultats obtenus lors de l'étude d'oscillateurs unidimensionnels avec une classe de PTSI et une classe de BTS TPIL. Lors de la séance les élèves ont réalisé le diagramme d'un oscillateur libre avec frottements visqueux en suivant les instructions de leur professeur (*fig.1*). Ceci a permis de les initier aux principes de base de SCICOS. Ils ont ensuite apporté les modifications nécessaires au programme pour étudier les portraits de phase (*fig.2*), l'aspect énergétique de l'oscillateur (*fig.3*), le passage au régime forcé (*fig.4*),…

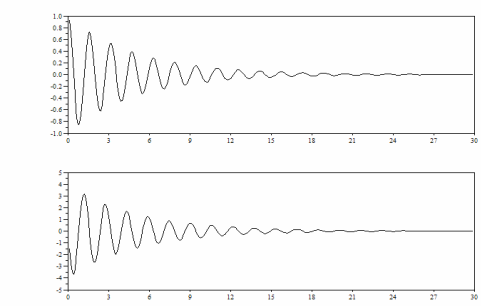

fig.1 : Oscillateur libre, position et vitesse

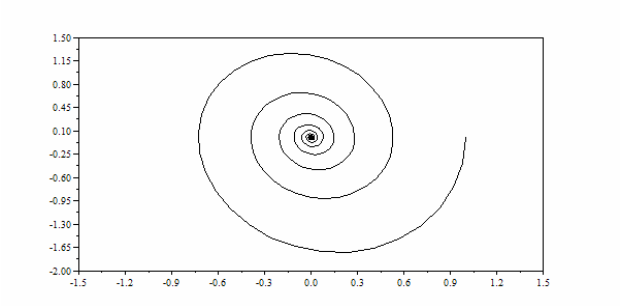

fig.2 : Portrait de phase

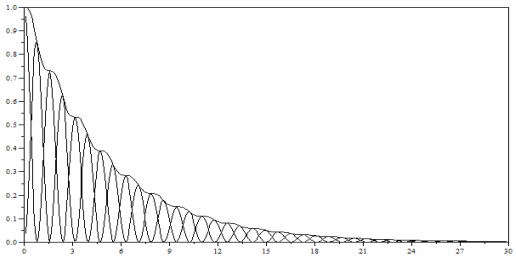

fig.3 : Etude énergétique

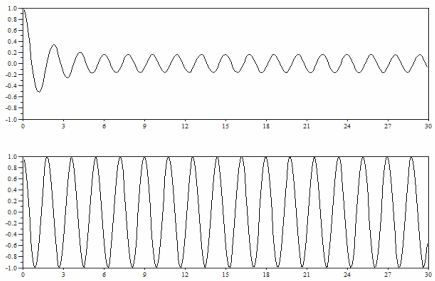

fig.4 : Oscillations harmoniques forcées,réponse et excitation

On peut aussi envisager des oscillations forcées puis comparer l'excitation (sinusoïdale, rectangulaire, …) et la réponse de l'oscillateur ou de nouveau s'intéresser au portrait de phase ou tout autre petit divertissement du même ordre.

I.2 Structure d'un programme sous 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 plus qu'à le disposer correctement dans la fenêtre de programmation graphique. Un programme SCICOS se présente sous la forme suivante (fig.5) :

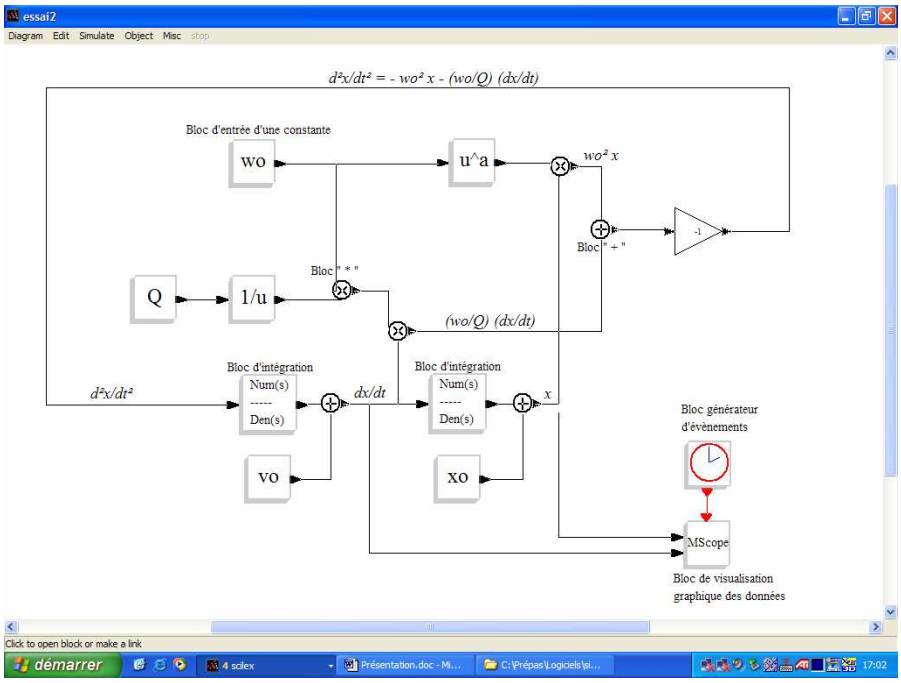

fig.5 : Simulation d'un oscillateur unidimensionnel amorti

Pour réaliser les opérations de calcul sur les données ainsi que différentes tâches (définition de variables, opérations mathématiques, lecture et écriture dans des fichiers, affichage graphique, etc…), l'utilisateur dispose de blocs (qui sont en fait des petits programmes) qu'il doit relier entre eux par des fils de données pour réaliser son programme. Le programmeur sélectionne ces blocs dans des palettes (*fig.8*) d'outils accessibles par une barre de menus (*fig.7*) et les dispose dans la fenêtre où il écrit ainsi son programme. Il doit ensuite relier les blocs entre eux pour que les données circulent d'un bloc à l'autre en y subissant les opérations prévues.

I.3 Les « blocs programmes » de SCICOS :

Les blocs disposent de deux 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 le traitement des données effectué.
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, …
Les blocs sont répartis dans des palettes auxquelles on accède par le menu **Edit** de la barre de menus (*fig.7*). Les blocs y sont regroupés par grandes familles (*fig.8*) :

- **Sources** : regroupe les blocs permettant de gérer les entrées (par exemple l'entrée des constantes qui paramètrent le problème.
- **Sinks** : regroupe les blocs permettant de gérer les sorties de données dans le diagramme.
- **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.
- Etc…

Les utilisateurs peuvent créer leurs propres blocs et leurs palettes. L'éventail disponible s'enrichit donc de version en version.

I.4 Gestion des programmes :

La gestion des fonctions (sauvegarde, édition, paramétrage, exécution, etc…) permettant d'écrire, de sauvegarder et d'exécuter les programmes SCICOS se fait à l'aide des menus de la barre de menus. Celle-ci contient six menus (*fig.7*) :

- **Diagram** : pour ouvrir, fermer ou enregistrer un diagramme. C'est l'analogue du menu **Fichier** des logiciels sous Windows ;
- **Edit** : pour accéder aux palettes contenant les blocs fonctionnels classés selon leur fonction ;
  pour déplacer, copier, supprimer un objet ou pour en lier deux ;
  pour fixer une valeur aux constantes du programme ;
- **Simulate** : pour paramétrer, tester et lancer la simulation ;
- **Object** : pour modifier les blocs ;
- **Misc** : pour accéder à des fonctions diverses ;
- **Stop** : pour interrompre le programme en cours d'exécution.

I.5 Utilisation des résultats :

Les données issues de la résolution numérique peuvent être présentées graphiquement par l'intermédiaire de différents blocs d'affichage qui fonctionnent de façon analogue aux oscilloscopes. Elles peuvent aussi être récupérées en tant que données numériques pour subir par exemple un traitement supplémentaire dans un tableur (Excel, …) ou dans un logiciel de modélisation (CurveExpert, …) ou encore dans un script écrit en langage SCILAB.

I.6 Se procurer SCILAB et l'installer :

Pour vous procurer le logiciel, vous pouvez le télécharger sur le site www.scilab.org. L'exemple traité dans la suite de l'article a été développé sous la version 3.1.1. La version 4 est actuellement disponible. La compatibilité des programmes développés sous la version 3 devrait être assurée.

Pour l'installer, il suffit de lancer l'exécutable et de suivre les étapes. Un raccourci apparaîtra sur le bureau pour un lancement simple par « double clic » sur l'icône. Sinon c'est le passage par la séquence « Démarrer → Tous les programmes → Scilab-x.x.x → Scilab-x.x.x.exe » qui permettra de lancer le logiciel.

Le module SCICOS se lance en tapant « scicos() » derrière la flèche de la ligne active, puis « Entrée ».

Si vous êtes prêt à vous lancer, laissez-vous guider dans l'exemple de la partie suivante pour démarrer.

II EXEMPLE DE REALISATION : SIMULATION D'OSCILLATEURS HARMONIQUES

II.1 Principe de résolution numérique d'une équation différentielle

II.1.1 Principe basique d'intégration numérique

L'intégration numérique en soi est un vaste sujet et il existe diverses méthodes plus ou moins sophistiquées que nous ne saurions développer ici. Disons simplement que le principe repose sur la connaissance des conditions initiales et des équations du système. Les valeurs des grandeurs à un instant t permettent alors le calcul des valeurs à l'instant $t + dt$. Le calcul réalisé en boucle permet d'obtenir l'évolution temporelle des variables du système.

II.1.2 Schématisation de l'équation

Il s'agit d'intégrer numériquement l'équation différentielle d'un pendule élastique : $\frac{d^2x}{dt^2} = -\frac{k}{m}x$.

Les variables à entrer seront la raideur du ressort « k », la masse du pendule « m » ainsi que les conditions initiales sur la position « xo »et la vitesse « vo ». Les variables à calculer sont l'accélération, la vitesse et la position. Il est conseillé de réaliser un diagramme sur papier du programme à écrire (*fig.6*).

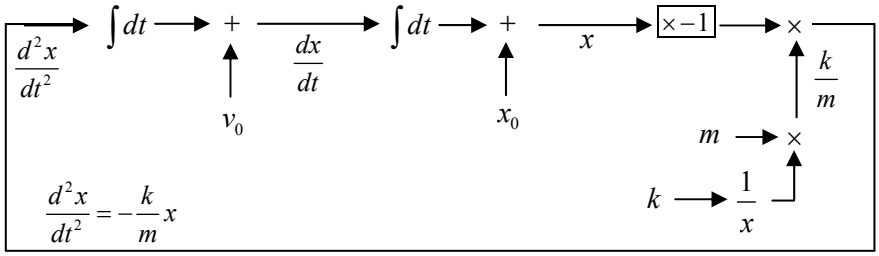

fig.6 : Diagramme sur papier

II.2 Ecriture du programme SCICOS

II.2.1 Lancement de SCICOS

« Double cliquez » sur l'icône SCILAB présente sur le bureau ou exécutez la séquence « Démarrer → Tous les programmes → Scilab-x.x.x → Scilab-x.x.x.exe » qui permettra de lancer le logiciel.

Le module SCICOS se lance en tapant « scicos() » derrière la flèche de la ligne active, puis « Entrée ».

Une fenêtre nommée **Untilted** s'est ouverte. C'est là que nous écrirons le programme.

Avant toute chose sauvegardons notre programme vide. Dans la barre de menu cliquez sur **Diagram** puis **Save as**. Dans la fenêtre, sélectionnez le répertoire de sauvegarde du fichier puis dans l'espace réservé à cet effet tapez le nom du fichier **suivi de .cos.** Il est indispensable de préciser l'extension pour que le fichier puisse être reconnu et exécuté par la suite. Pensez à sauvegarder régulièrement au fur et à mesure de la programmation.

Pour vous aider dans la construction du diagramme, vous pouvez vous référer à la figure 22.

II.2.2 Insertion des blocs d'intégration

- Dans le menu **Edit**, sélectionnez la rubrique **Palettes** (*fig.7*), puis la palette nommée **Linear** (*fig.8*).
- La palette d'outils apparaît à l'écran et vous pouvez alors sélectionner **le bloc « num(s) /den(s) »** (*fig.9*) et le disposer sur le diagramme.

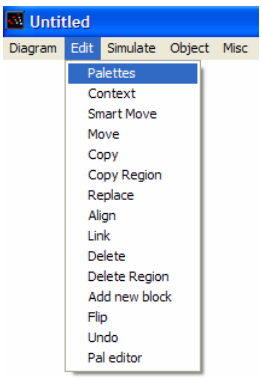

*fig.7 : Barre des menus, menu **Edit***

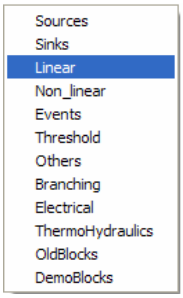

fig.8 : Liste des palettes

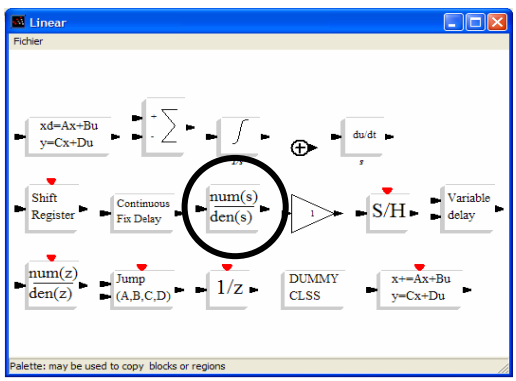

*fig.9 : Palette **Linear***

- Cliquez dessus et sélectionnez *1* pour le numérateur et *s* pour le dénominateur (*fig.10*). On réalise ainsi un intégrateur (le bloc travaille à partir de la transformation de LAPLACE).
- Insérez un deuxième intégrateur dans la fenêtre.
- Pour déplacer un bloc dans la fenêtre, dans le menu **Edit**, sélectionnez l'item **Move** (*fig.7*). Il suffit alors de cliquer sur le bloc et de la faire glisser dans le diagramme. Un nouveau clic le fixera à sa nouvelle position.

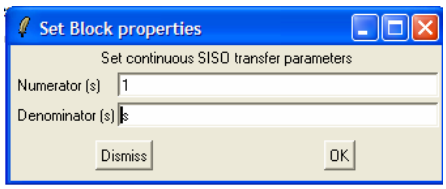

fig.10

- Pour supprimer un élément de la fenêtre, dans le menu **Edit**, sélectionnez l'item **Delete** (*fig.7*). Il suffit alors de cliquer sur l'élément en question pour le faire disparaître.

II.2.3 Introduction des paramètres de l'oscillateur

- Dans le menu **Edit**, sélectionnez la rubrique **Context** (*fig.11*) et déclarez les paramètres de l'oscillateur : « k », « m », « xo » et « vo » (*fig.12*).

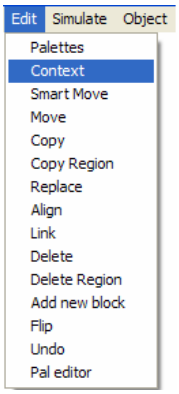

fig.11

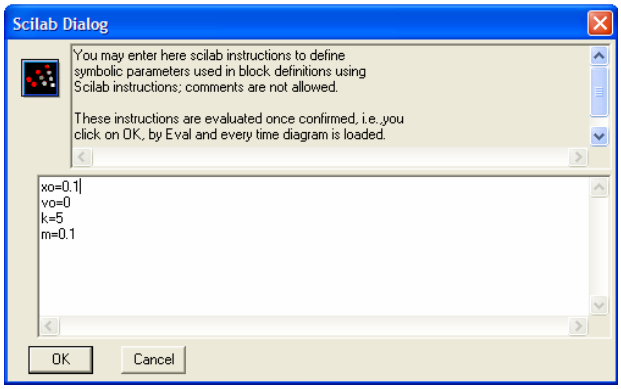

*fig.12 : Renseignement du **Context***

- Dans le menu **Edit**, sélectionnez ensuite la rubrique **Palettes**. Choisissez alors celle nommée **Sources** et validez. La palette d'outils (*fig.14*) apparaît à l'écran et vous pouvez alors sélectionner **le bloc d'entrée de constante « 1 »** en cliquant dessus et l'insérer dans le diagramme.

fig.13

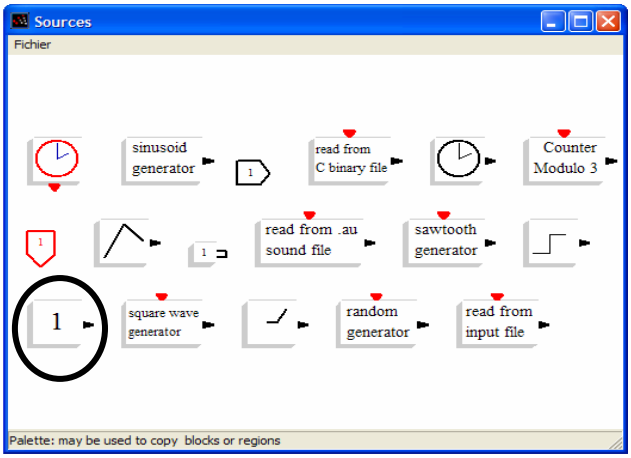

*fig.14 : Palette **Sources***

- Cliquez sur le bloc et entrez le nom de la constante désirée, par exemple vo (*fig.13*). Comme elle est définie dans le contexte de la simulation sa valeur lui sera automatiquement affectée.
- Recommencez l'opération pour les trois autres paramètres.

II.2.4 Insertion des opérateurs

- Dans la palette **Linear** (*fig.16*), sélectionnez le **bloc d'addition** et placez-le sur le diagramme. Recommencez l'opération pour placer l'autre bloc d'addition nécessaire.
- Dans la même palette, sélectionnez **le bloc amplificateur**, placez le sur le diagramme. Cliquez dessus pour accéder à son paramétrage (*fig.15*). Choisissez *–1* pour le gain.

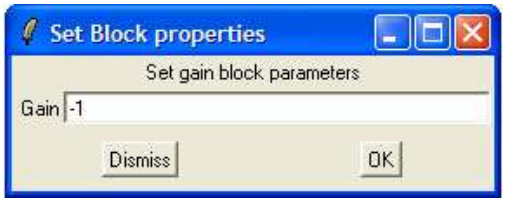

fig.15

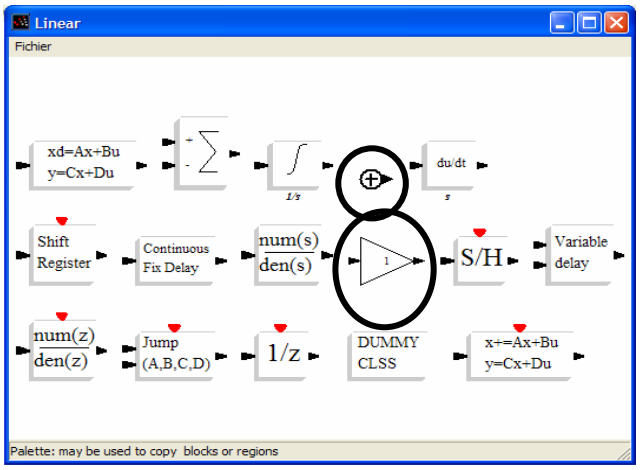

*fig.16 : Palette **Linear***

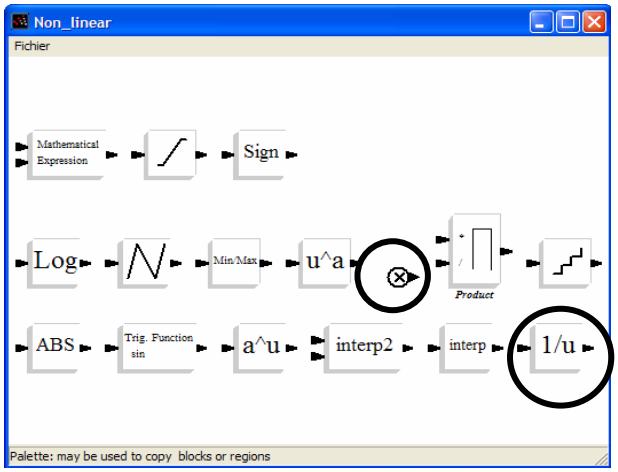

*fig.17 : Palette **Non Linear***

- Dans la palette **Non Linear** (*fig.17*), sélectionnez le **bloc de multiplication** et placez-le sur le diagramme. Recommencez l'opération pour placer l'autre bloc de multiplication nécessaire.
- Dans la même palette, sélectionnez **le bloc inverseur**, placez le sur le diagramme.

II.2.5 Liaison des blocs

- Il ne reste plus qu'à relier les blocs entre eux pour obtenir un diagramme fonctionnel (*fig.22*). Pour cela, cliquez sur la sortie d'un bloc et tirez le fil de données (noir) ou d'événement (rouge) jusqu'à l'entrée du bloc suivant. Si le fil ne se crée pas, sélectionnez la fonction **Link** dans le menu **Edit**.
- Vous pouvez alors ajuster le placement des blocs pour obtenir un diagramme plus propre et lisible.

II.2.6 Ajout d'une sortie graphique

- Dans la palette **Sinks** (*fig.18*), sélectionnez le **bloc Mscope**. Disposez-le sur le graphique et cliquez dessus pour le paramétrer (*fig.19*). En cliquant sur ce bloc, on peut choisir les couleurs et style d'affichage, la fenêtre, sa position, sa taille, les échelles, etc… On pourra prendre par exemple, pour xo = 0.1, vo = 0, k = 5 et m = 0.1, « Ymin vector » « -0.1 -1 », « Ymax vector » « 0.1 1 », « Refresh period » « 20 » et conserver les autres paramètres par défaut.

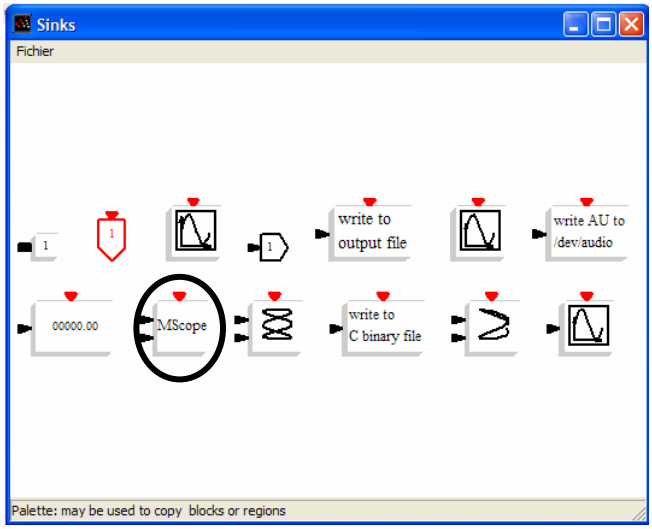

*fig.18 : Palette **Sinks***

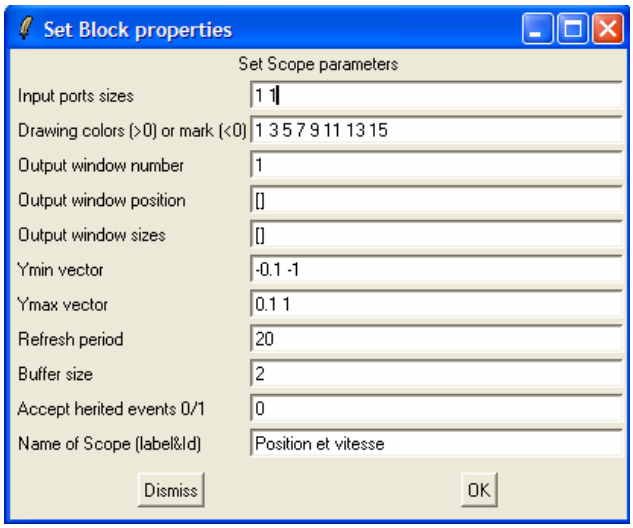

*fig.19 : Paramétrage du bloc **Mscope***

- Pour fonctionner correctement l'oscilloscope a besoin d'un **générateur d'évènements**. Celui-ci est disponible dans la palette **Sources** (*fig.14*) ou dans la palette **Events** (*fig.20*). On paramètre la période de génération des évènements (laisser 0.01s dans un premier temps) et la date du premier événement généré (mettre 0) (*fig.21*).

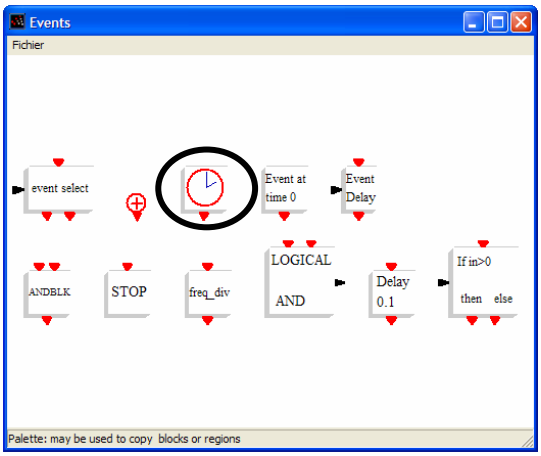

*fig.20 : Palette **Events***

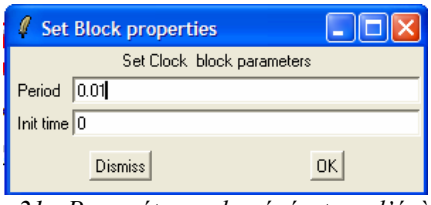

fig.21 : Paramétrage du générateur d'évènements

Vous devez maintenant avoir construit le diagramme suivant :

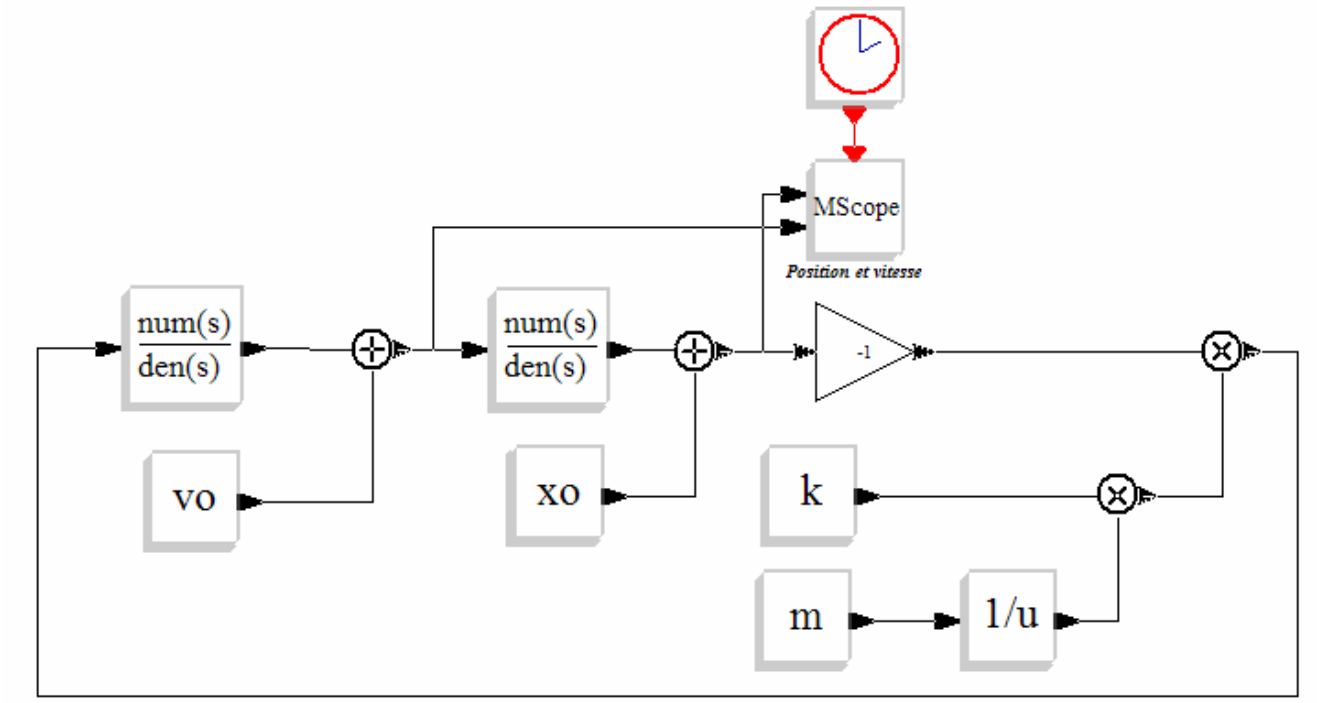

fig.22 : Diagramme permettant la simulation d'un oscillateur harmonique et la visualisation de x(t) et v(t)

II.2.7 Paramétrage et lancement de la simulation

- Dans le menu **Simulate**, choisissez la rubrique **Setup**. On y gère les paramètres de la simulation (*fig.23*). 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. Prenez un temps raisonnable (20, 60s, …). Il sera ajusté en fonction des premiers résultats.
- Pour lancer la simulation il suffit de choisir l'option **Run** du menu **Simulate**.

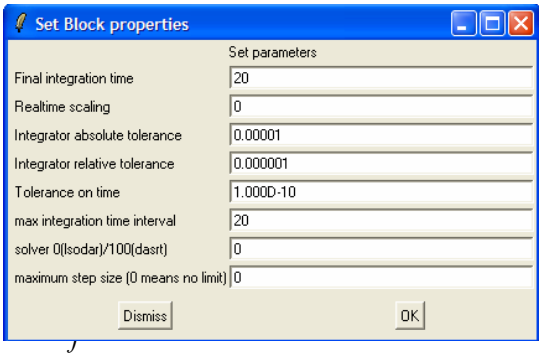

fig.23 : Paramétrage de la simulation

Le diagramme réalisé permet de visualiser les courbes de position et vitesse en fonction du temps (*fig.24*). Il est possible ensuite de jouer sur les paramètres de l'oscillateur à l'aide du menu **Context** (*fig.11 et 12*).

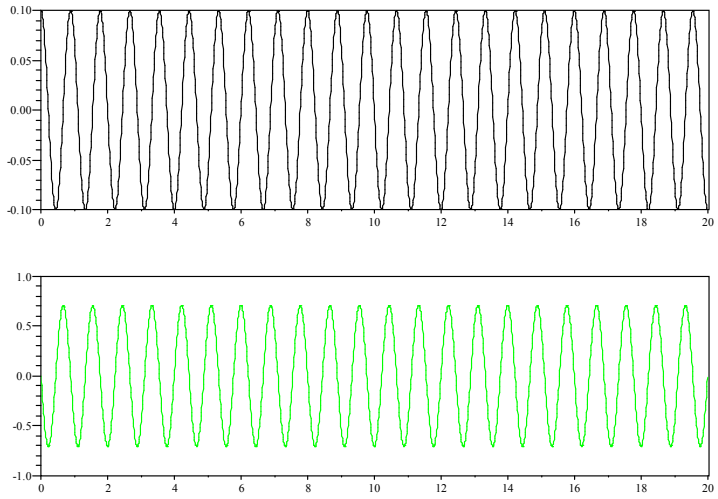

fig.24 : Evolution temporelle de la position et de la vitesse

Pour accéder ultérieurement à ce programme, il suffira de sélectionner dans le menu **Diagram** l'option **Load** et de choisir le fichier correspondant.

CONCLUSION

L'utilisation pédagogique du module SCICOS peut être envisagée de plusieurs manières :

- Production de simulation ou de courbes a exploiter comme illustration du cours.
- Séances de travaux pratiques avec des programmes de simulation fournis « prêts à l'emploi » par le professeur afin les élèves étudient l'influence des paramètres accessibles. En ces termes, le module SCICOS présente un intérêt certain lorsqu'on veut présenter des systèmes n'ayant pas de solution analytique et peut être utilisé par des élèves dès la terminale.
- Séances de travaux pratiques en proposant aux élèves de réaliser un diagramme puis un programme par eux-mêmes.

Le premier type de séance a été réalisé avec des élèves de BTS TPIL pour une étude qualitatives des oscillateurs unidimensionnels et le second type avec des élèves de P.T.S.I. qui en deux heures sont parvenus à réaliser un programme de simulation d'un oscillateur unidimensionnel amorti et à le développer. A l'issue de la séance, il apparaît que les étudiants ont apprécié ce nouveau logiciel qu'ils ont pu dompter (au moins pour les fonctions de base) assez rapidement. Il est vrai aussi que ce principe de programmation par blocs fonctionnels s'apparente à ce qui est fait en sciences industrielles la même année dans le cadre de l'automatique**.**

Pour en savoir plus :

* le site officiel Scilab :

http://www.scilab.org pour entre autres télécharger gratuitement le logiciel ainsi qu'un document pour vos premiers pas avec de Scilab (http://www.scilab.org/doc/Scilabpratique).

* *Introduction à Scilab :* Collection IRIS par JP Chancelier, F. Delebecque, C. Gomez, M. Goursat, R. Nikoukhah, S. Steer. Qui nous a permis de débuter avec ce language.