

# L'AUTOMATIQUE DES MODÈLES LINÉAIRES REVISITÉE À LA LUMIÈRE DU CALCUL SYMBOLIQUE

Patrick Sibille

*Nancy-Université, Université Henri Poincaré  
Centre de Recherche en Automatique de Nancy (CRAN), CNRS UMR 7039  
BP 239, F-54506 Vandœuvre Cedex, France,  
Tél : (33) 3 83 68 44 61 - Fax : (33) 3 83 68 44 62  
Mél : Patrick.Sibille@cran.uhp-nancy.fr*

**Résumé** Cet article est dédié, d'une part, à l'utilisation d'un logiciel de calcul symbolique pour l'enseignement des fondements de l'automatique, et, d'autre part, à la présentation des nouveaux développements d'un outil de calcul formel. Cette boîte à outils, baptisée : **AutoSLI** pour Automatique des Systèmes Linéaires Invariants, a trait à la manipulation de modèles linéaires décrits sous formes de transfert ou d'état. Ces fonctions donnent des solutions analytiques à des problèmes liés au maniement de ces représentations, en exploitant efficacement un logiciel de calcul formel, en l'occurrence Maple. Face à la désaffection des étudiants pour nos disciplines et plus généralement pour les études scientifiques, il est nécessaire d'aborder l'automatique sous un autre angle. Or, ces outils permettent de traiter non seulement des représentations simples, de type académique, mais également des modèles de processus physiques qui comportent un grand nombre de paramètres ou d'ordres élevés sans que pour autant l'aspect calcul n'occulte l'interprétation des résultats. Ils sont obtenus sans effort vis à vis de calculs qui sont quelquefois fastidieux et sans intérêt pour être menés à la main. Cette bibliothèque permet, sans aucun développement logiciel, à des étudiants non spécialistes du calcul symbolique de résoudre les problèmes fondamentaux de l'automatique. Ce travail s'inscrit dans le cursus de la licence de Sciences et Technologies, mention «Électronique, Électrotechnique, Automatique et Réseaux (EEAR)».

**Mots clés** Calcul formel, calcul symbolique, automatique, fonction de transfert, représentation d'état, Maple.

## 1. INTRODUCTION

Les outils employés par la grande majorité des automaticiens, pour résoudre les problèmes qui leur sont posés, sont des logiciels scientifiques dédiés au calcul numérique comme : Scilab (logiciel libre développé par l'Institut National de Recherche en Informatique et en Automatique (INRIA) et l'École Nationale des Ponts et Chaussées (ENPC)) ou Matlab (logiciel commercialisé par Mathworks). Ce sont également ces produits qui sont principalement introduits comme supports pédagogiques pour l'enseignement de l'automatique. Dans cet article, nous présentons l'utilisation d'un système de calcul formel, moyen indispensable et complémentaire aux outils de calcul numérique. Le logiciel exploité, pour illustrer la mise en œuvre de connaissances de base de l'automatique, est ici Maple. Tout comme pour les logiciels de simulation numérique, il existe un choix important de systèmes généraux de calcul formel, on peut citer les principaux, outre Maple : Axiom, Mathematica, MuPad et Reduce.

Maple a été choisi pour mettre en valeur l'intérêt du calcul symbolique dans l'enseignement de l'automatique

car il offre des passerelles avec Matlab et Scilab. Il existe, aussi sous Matlab, une boîte à outils intitulée «Symbolic Math Toolbox» qui s'appuie sur le noyau de Maple. De surcroît, celui-ci est très répandu dans diverses communautés scientifiques. Contrairement aux logiciels «numériques» précédemment cités, ce système de calcul formel n'intègre pas de bibliothèques spécifiques à l'automatique. Néanmoins, il inclut un ensemble d'outils de calcul scientifique appelés «packages» dédiés ([1,2,3,4]) : à l'algèbre linéaire, aux opérations sur les polynômes et fractions rationnelles, au calcul différentiel, à la résolution d'équations différentielles ordinaires ou partielles, aux calculs des transformées en Z et de Laplace, aux traitements statistiques, ... L'ensemble de ces bibliothèques renferme les constituants nécessaires au développement d'une boîte à outils propre à l'automatique ([5]).

Les logiciels de calcul symbolique sont amplement utilisés par les communautés mathématicienne et informaticienne ([6,7,8,9,10,11]) mais également par celle des physiciens ([12,13,14,15]) et d'autres encore ([16,17,18,19,20]). En outre, des ouvrages dévolus aux ingénieurs sont disponibles ([21,22,23,15,24]). Curieuse-

ment, les systèmes de calcul formel sont très peu utilisés en automatique tant en recherche qu'enseignement ([25,26,27]). Le recours à ce logiciel qui rend un résultat fiable en une fraction de seconde, évite de manipuler, laborieusement, à la main, de longues expressions littérales. De cette façon, grâce à une utilisation judicieuse de cet outil efficient, l'étudiant s'affranchit de la tâche ingrate des calculs, pour concentrer toute son attention sur les concepts et l'interprétation des résultats fournis par le système symbolique. C'est pourquoi dans cet article, nous présentons une bibliothèque de programmes qui résout des problèmes usuels de l'automatique comme le calcul, sous une forme littérale, d'une fonction de transfert ou d'une forme d'état provenant de la mise en boucle fermée d'un système.

Dans cet article, nous avons choisi d'illustrer quelques thématiques fondamentales relatives à la manipulation de modèles linéaires sous formes d'état ou de transfert. Il est bien évident qu'un outil de calcul symbolique ne se réduit pas à la résolution des problèmes exposés ici : il permet de résoudre bien d'autres aspects. Des problèmes parfois épineux de discrétisation, de dérivation, d'intégration, de synthèse de régulateurs, ... peuvent être résolus au moyen de ce type de logiciels. Les quelques exemples caractéristiques, exposés ici, ont essentiellement pour but de montrer comment tirer profit du calcul symbolique en automatique. Dans le même état d'esprit, des fonctionnalités nouvelles pourraient être facilement étendues par l'utilisateur, pour constituer une boîte à outils plus complète pour l'analyse des modèles linéaires. Par ailleurs, les résultats exhibés sont sous forme littérale, ils pourraient bien évidemment être de nature numérique.

Ce travail est structuré en 2 parties principales. Dans la première partie, l'organisation et les fonctionnalités de la bibliothèque de fonctions, baptisée **AutoSLI** (pour Automatique des Systèmes Linéaires Invariants), sont décrites. Dans cette nouvelle version de la bibliothèque de nouvelles fonctionnalités ont été développées pour répondre, par rapport à une fonction de transfert : à la question de la stabilité, aux calculs des pôles et zéros, aux calculs des module et argument. La linéarisation d'un modèle est relativement souvent pénible, ainsi la fonction **linmod** offre une solution sûre à ce problème. **linmod** fonctionne en multi-entrées/multi-sorties, elle rend une représentation d'état ou une matrice de transfert. Enfin, une procédure assurant la conversion d'un modèle à temps discret vers un modèle à temps continu a été ajoutée. Des illustrations didactiques sont ensuite mises à profit pour juger de l'efficacité de quelques uns des algorithmes de la boîte à outils proposée. Enfin, un bilan pédagogique est formulé.

## 2. PRÉSENTATION DE LA NOUVELLE VERSION DE LA BOÎTE À OUTILS AUTOSLI

La boîte à outils développée permet d'effectuer quelques uns des traitements élémentaires demandés dans un cours d'automatique de base. Pour opérer ces tâches, la formulation sous forme d'état a été choisie car, même si cette bibliothèque est formulée dans le cas monovariante, la plupart des programmes peut être aisément réécrite dans un contexte multivariable. Les fonctions admettent donc des arguments de type représentation d'état, mais elles acceptent bien évidemment, également, des modèles sous forme de fonctions de transfert.

Le logiciel de calcul numérique Matlab étant largement répandu dans notre communauté, pour simplifier et rendre confortable l'emploi de cette bibliothèque, les fonctions développées portent des dénominations identiques ou similaires à celle de Matlab lorsque cela est possible. En outre, pour rendre plus simple l'utilisation de ces fonctions, leur appel peut se réaliser, tout comme sous Scilab ou Matlab, avec un nombre de paramètres variables et optionnels. Cette boîte à outils permet :

- de donner une solution analytique aux réponses impulsionnelles et indicelles de modèles décrits sous formes de fonction de transfert ou d'état à temps continu ou échantillonné ;
- d'obtenir une représentation d'état canonique : commandable, observable ou de Jordan. Cette même fonction assure également le changement de base d'une forme d'état ;
- d'analyser un modèle en calculant, par exemples : les matrices de commandabilité et d'observabilité d'une représentation d'état, les pôles et les zéros, le module et l'argument d'une fonction de transfert. En outre, la stabilité peut être obtenue en construisant le tableau de Routh, ...
- d'engendrer la fonction de transfert ou la forme d'état découlant de l'association d'éléments de type «transfert» ou «état». Ainsi, il est possible de calculer le résultat de la mise en cascade de deux éléments d'un même type ou non, ou de celui résultant de la mise en boucle fermée de trois éléments quelconques, les deux premiers étant relatifs à la description de la chaîne directe et le dernier représentant la chaîne de retour. La nature des résultats («transfert» ou «état») fournis par les fonctions, est choisie par l'utilisateur ;
- de convertir un modèle décrit sous forme de transfert en une représentation d'état canonique observable et vice versa. Il est également possible d'obtenir le modèle d'état linéarisé tangent à partir d'un modèle d'état non linéaire ;
- de discrétiser, sous différentes hypothèses, un modèle à temps continu pour obtenir une fonction de transfert ou représentation d'état échantillonnées. Six méthodes de discrétisation sont implantées : approximations sous l'hypothèse d'un bloqueur

d'ordre 0 ou d'un bloqueur d'ordre 1 non causal, discrétisation par la méthode de Tustin avec ou sans pré-compensation de distorsion de fréquence, approximations rectangulaires par les opérateurs d'Euler (ou «avance») et de «retard». La fonction inverse qui assure la conversion d'un modèle à temps discret vers un modèle à temps continu est également disponible.

Un aperçu des fonctionnalités de la bibliothèque de programmes est donné à la figure (1). Cette bibliothèque est divisée en sept champs thématiques. Le premier concerne les réponses temporelles de modèles linéaires aux signaux impulsions ou indiciels. Les techniques d'obtention des différentes formes canoniques sont rassemblées dans le champ nommé «changement de forme d'état». Le troisième groupe est intitulé «analyse d'un modèle», on y trouve les fonctions élémentaires d'analyse. Le quatrième item a trait à l'association de modèles : en cascade ou en boucle fermée. La conversion de modèles transfert/état et vice versa constitue le cinquième champ. Le sixième réunit l'ensemble des fonctions relatives à la discrétisation de modèles sous forme : état ou transfert. Enfin, le dernier item permet d'effectuer l'opération inverse du précédent.

En ce qui concerne l'utilisation de cette bibliothèque, à chaque fonction est associé un fichier d'aide. Dans chaque page d'aide, les informations suivantes sont affichées :

- un descriptif de la fonction est donné ;
- les différents arguments d'entrée et de sortie de la fonction sont listés et définis, avec le cas échéant les paramètres optionnels ;
- les diverses variantes d'appel de la procédure sont exhibées ;
- des exemples d'utilisation instructifs et diversifiés sont exposés.

Ces éléments sont indispensables pour une utilisation conviviale et efficace de l'outil. Dans cet article, bien que cet aspect soit fondamental, aucun exemple de page d'aide ne sera reproduit pour des raisons de concision.

### 3. ILLUSTRATION DE QUELQUES UNES DES NOUVELLES FONCTIONNALITÉS DE LA BIBLIOTHÈQUE AUTOSLI

Ce paragraphe a pour objectif de répondre à 2 questions élémentaires : Que peut-on faire avec cette boîte à outils AutoSLI ? Comment l'utiliser ?

Les exemples choisis sont d'une complexité limitée de manière à minimiser l'encombrement des expressions rendues. Pour ce même motif, bien que parfaitement réalisable, aucun résultat numérique et graphique ne sera présenté. L'exportation sous Matlab, des expressions lit-

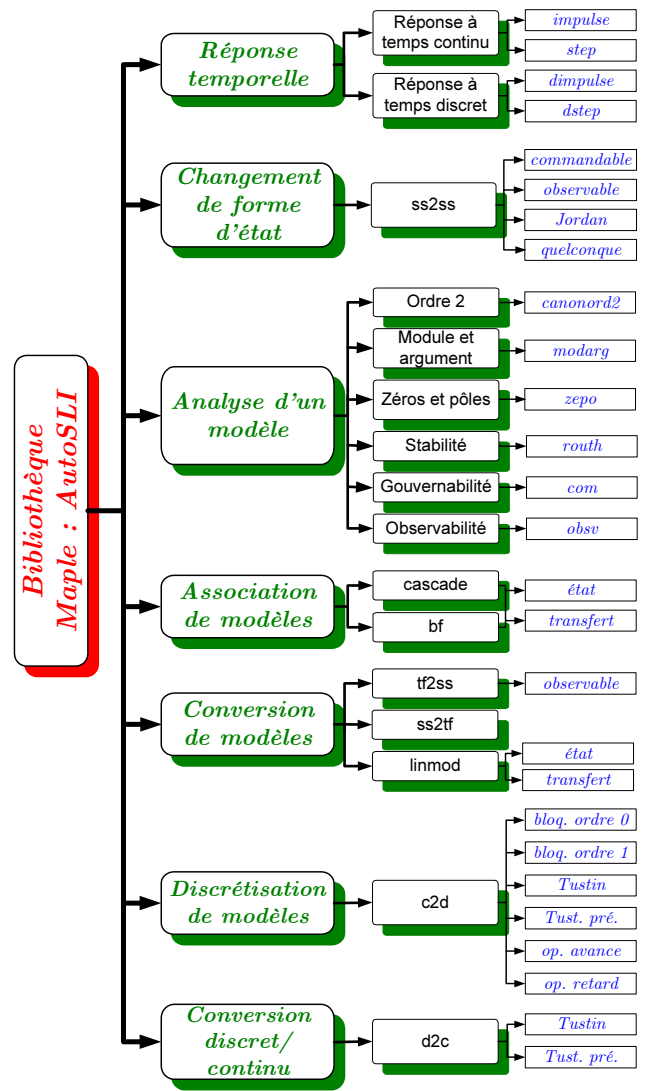


Fig. 1. Fonctionnalités de la bibliothèque Maple : AutoSLI (Automatique des Systèmes Linéaires Invariants)

térales produites par Maple, ne sera également pas traitée.

Pour illustrer les fonctionnalités de la bibliothèque, considérons un banc électro-mécanique (Cf. figure (2)). Le dispositif expérimental est constitué d'un moteur relié à une génératrice par un accouplement rigide. L'entrée et la sortie sont respectivement la tension d'alimentation et la vitesse de rotation du moteur.

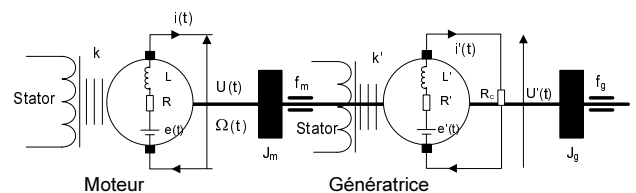


Fig. 2. Schéma de l'ensemble moteur-génératrice

Afin d'illustrer, de manière didactique, les fonctionnalités du logiciel nous poserons quelques hypothèses simplificatrices réalistes.

**Notations :**

- $k$  : Constante de flux ou de couple,
- $R, Rc$  : Résistances de l'induit et de charge,
- $L$  : Self de l'induit,
- $J$  : Moment d'inertie (moteur+génératrice),
- $f$  : Frottements visqueux (moteur+génératrice),
- $\Omega(t)$  : Vitesse de rotation de l'axe du moteur,
- $U(t)$  : Tension aux bornes de l'induit,

**Hypothèses simplificatrices :**

- les 2 moteurs sont supposés identiques,
- l'accouplement entre les 2 moteurs est rigide,
- les frottements secs sont négligeables,
- les constantes de flux et de couple sont identiques.

Maple contient des centaines de commandes qui sont opérationnelles en ouvrant une feuille de travail. D'autres commandes ne sont accessibles que si l'utilisateur a demandé explicitement le chargement des fonctions. Ces fonctions sont rangées dans des bibliothèques particulières, intitulées «packages». La boîte à outils `AutoSLI` n'est pas chargée initialement. Le code Maple correspondant à son chargement est édité ci-après. Cette commande rend, notamment, la liste des fonctions disponibles dans cette bibliothèque. La 2<sup>ème</sup> ligne de commande montre comment faire appel à l'aide. Le système considéré est enfin défini sous Maple de la manière suivante :

```
> with(AutoSLI);
      Boîte à outils : AutoSLI, version 1.2
                        Patrick Sibille
      Nancy-Université, Université Henri Poincaré
      Centre de Recherche en Automatique de Nancy
      Pour obtenir de l'aide sur la boîte à outils : ? AutoSLI
      Pour obtenir de l'aide sur une fonction
      particulière : ? <nom de la fonction>
      [Help, Version, bf, c2d, canonord2, cascade, com, d2c,
      dimpulse, dstep, impulse, linmod, modarg, obsv, routh,
      ss2ss, ss2tf, step, tf2ss, zepo]

> G := k*(R+Rc)/((2*R+Rc)*k^2+(R^2+Rc*R)*f)/(1+L*J*(R+Rc)/
((2*R+Rc)*k^2+(R^2+ Rc*R)*f)*s^2+
(L*k^2+(R+Rc)*L*f+(R^2+Rc*R)*J)/
((2*R+Rc)*k^2+(R^2+Rc*R)*f)*s);
G := k (R + Rc) / ( ((2 R + Rc) k^2 + (R^2 + Rc R) f)
(1 + (L J (R + Rc) s^2) / ((2 R + Rc) k^2 + (R^2 + Rc R) f)
+ (L k^2 + (R + Rc) L f + (R^2 + Rc R) J) s) /
((2 R + Rc) k^2 + (R^2 + Rc R) f) )
```

**3.1 Passage d'une fonction de transfert à une forme d'état**

Exploisons la fonction de transfert de ce système électromécanique pour illustrer le passage d'une fonction de transfert à une forme d'état. Cette conversion s'effectue grâce à la fonction `tf2ss`. Cette procédure renvoie un quadruplet  $(Ac, Bc, Cc, Dc)$  qui constitue les éléments

de la représentation d'état. La représentation rendue est une forme canonique observable. Cette forme d'état pourrait être convertie, sous une autre représentation d'état canonique, en utilisant la fonction `ss2ss`.

```
> tf2ss(G,s,'Ac','Bc','Cc','Dc'):print(Ac,Bc,Cc,Dc);
```

Le système est supposé observable

$$\begin{bmatrix} 0 & -\frac{2 R k^2 + f R^2 + R f R c + k^2 R c}{L J R + L J R c} \\ 1 & -\frac{J R^2 + R J R c + L k^2 + L f R c + L f R}{L J R + L J R c} \end{bmatrix}, \begin{bmatrix} (R + R c) k \\ L J R + L J R c \\ 0 \end{bmatrix}, [0 \ 1], [0]$$

**3.2 Utilisation de la fonction bf sur le modèle du banc électro-mécanique**

Supposons que ce système  $(G(s))$  soit monté en cascade avec un correcteur PI  $(Cor(s))$  de structure parallèle et bouclé par un retour unitaire (Cf. schéma (3)). Calculons la représentation d'état canonique observable et la fonction de transfert en boucle fermée de l'ensemble à l'aide de la procédure `bf`. A noter que `bf` renvoie par défaut une forme d'état, mais, l'utilisateur peut spécifier la nature du résultat désiré, par exemple transfert, en ajoutant un paramètre d'entrée affecté à la valeur «ft».

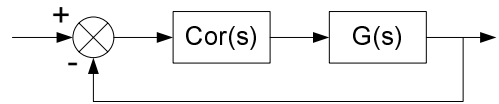


Fig. 3. Schéma de la boucle fermée du modèle du banc électro-mécanique

```
> Cor:=K+1/(T[i]*s);
```

$$Cor := K + \frac{1}{T_i s}$$

```
> bf(Cor,[Ac,Bc,Cc,Dc],1,'sys',s):
print(sys[1],sys[2],sys[3],sys[4]);
```

Le système est supposé observable

$$\begin{bmatrix} 0 & 0 & -\frac{R k + k R c}{\%1} \\ 1 & 0 & -\frac{k R K T_i + k R c K T_i + 2 T_i R k^2 + T_i f R^2 + T_i R f R c + T_i k^2 R c}{\%1} \\ 0 & 1 & -\frac{T_i J R^2 + T_i R J R c + T_i L k^2 + T_i L f R c + T_i L f R}{\%1} \end{bmatrix}, \begin{bmatrix} (R + R c) k \\ \%1 \\ (R + R c) k K T_i \\ \%1 \\ 0 \end{bmatrix}, [0 \ 0 \ 1], [0]$$

$$\%1 := T_i L J R + T_i L J R c$$

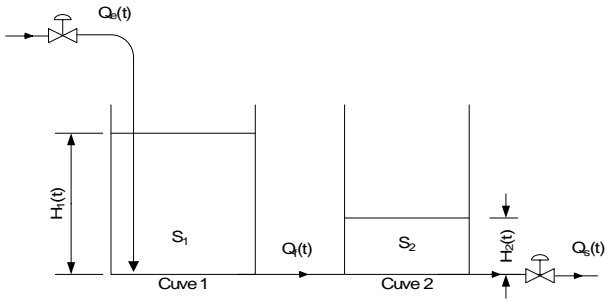


Fig. 4. Schéma du système des 2 cuves

Bien que le modèle initial soit simple et que le correcteur choisi soit des plus classiques, la lecture des divers résultats renvoyés par cette bibliothèque de calcul symbolique est quelque peu fastidieuse. Nul doute que ce type de calcul, conduit à la main, n'enthousiasmerait que modérément les étudiants !

### 3.3 Illustration de la fonction linmod sur un système hydraulique

Considérons maintenant, le modèle physique, non linéaire, d'une installation constituée de deux cuves couplées (Cf. figure (4)). Ce banc comporte :

- 2 vannes motorisées permettant la commande des débits d'entrée ou de sortie d'eau,
- 2 sondes de niveau,
- 2 débitmètres assurant les mesures des débits d'entrée et de sortie du fluide.

Dans l'objectif de synthétiser une loi de commande fondée sur un modèle linéaire, les étudiants cherchent à obtenir une représentation linéarisée de ce système. Cette installation doit assurer la régulation de la hauteur dans la cuve 1 et la régulation du débit en sortie de la cuve 2 autour d'un point de fonctionnement.

Ce système peut-être considéré, d'un point de vue de l'automatique, comme un système à 2 entrées (commande des vannes) et 2 sorties (hauteur dans la cuve 1 et débit en sortie de la cuve 2). Les caractéristiques non linéaires de l'installation sont données par les lois de variation des débits d'entrée  $Q_e(t)$ , de sortie  $Q_s(t)$  et de fuite  $Q_f(t)$  entre les deux cuves :

$$Q_e(t) = K_e Y_e(t) \sqrt{\alpha P - H_1(t)} \quad \text{avec} \quad \alpha P \gg H_1(t)$$

$$Q_s(t) = K_s Y_s(t) \sqrt{H_2(t) + H_0 - f Q_s^2(t)}$$

$$Q_f(t) = K_f \text{sign}(H_1(t) - H_2(t)) \sqrt{|H_1(t) - H_2(t)|}$$

Avec,

- $K_e$  : coefficient de débit caractéristique de la vanne 1,
- $K_f$  : coefficient de débit caractéristique du tuyau inséré entre les cuves,
- $K_s$  : coefficient de débit caractéristique de la vanne 2,
- $f$  : coefficient de perte de charge,
- $Y_e(t)$  : position d'ouverture de la vanne d'entrée (unité : %),
- $Y_s(t)$  : position d'ouverture de la vanne de sortie (unité : %),
- $\alpha$  : coefficient (unité :  $m/\text{bar}$ ),
- $P$  : pression du réseau d'alimentation en eau (unité :  $\text{bar}$ ),
- $H_1(t)$  : hauteur d'eau dans la cuve 1 (unité :  $m$ ),
- $H_2(t)$  : hauteur d'eau dans la cuve 2 (unité :  $m$ ),
- $H_0$  : différence de hauteur entre le fond de la cuve 2 et l'égout (unité :  $m$ ),
- $Q_e(t)$  : débit d'entrée (unité :  $m^3/s$ ),
- $Q_f(t)$  : débit de fuite (unité :  $m^3/s$ ),
- $Q_s(t)$  : débit de sortie (unité :  $m^3/s$ ).

Les étudiants doivent mettre ce système sous la forme standard :

$$\begin{cases} \dot{X}(t) = F(X(t), U(t)) \\ Y(t) = G(X(t), U(t)) \end{cases}$$

Où,

$$X(t) = \begin{bmatrix} H_1(t) & H_2(t) \end{bmatrix}^T$$

$$U(t) = \begin{bmatrix} Y_e(t) & Y_s(t) \end{bmatrix}^T$$

$$Y(t) = \begin{bmatrix} H_1(t) & Q_s(t) \end{bmatrix}^T$$

Il suffit alors de définir, sous Maple, les fonctions  $F(X(t), U(t))$ ,  $G(X(t), U(t))$  et le point de fonctionnement de la façon ci-après :

```
> F:= [1/S[1]*(K[e]*sqrt(alpha*P-H[1])*Y[e]
-K[f]*signum(H[1]-H[2])*sqrt(abs(H[1]-H[2]))),
1/S[2]*(K[f]*signum(H[1]-H[2])*sqrt(abs(H[1]-H[2]))
-K[s]*Y[s]*sqrt(H[2]+H[0])/sqrt(1+K[s]^2*Y[s]^2*f))];
```

$$F := \left[ \frac{K_e \sqrt{\alpha P - H_1} Y_e - K_f \text{signum}(H_1 - H_2) \sqrt{|H_1 - H_2|}}{S_1}, \right.$$

$$\left. \frac{K_f \text{signum}(H_1 - H_2) \sqrt{|H_1 - H_2|} - \frac{K_s Y_s \sqrt{H_2 + H_0}}{\sqrt{1 + K_s^2 Y_s^2 f}}}{S_2} \right]$$

```
> G:= [H[1], K[s]*Y[s]*sqrt(H[2]+H[0])/
sqrt(1+K[s]^2*Y[s]^2*f)];
```

$$G := \begin{bmatrix} H_1, & \frac{K_s Y_s \sqrt{H_2 + H_0}}{\sqrt{1 + K_s^2 Y_s^2 f}} \end{bmatrix}$$

```
> etat_0:= [H[10], H[20]]; # [H1[0], H2[0]]
u_0:= [Y[e0], Y[s0]];
```

```
etat_0 := [H10, H20]
```

```

u_0 := [Y_e0, Y_s0]
> X:=[H[1],H[2]]; U:=[Y[e],Y[s]];
      X := [H1, H2]
      U := [Y_e, Y_s]

```

Ce système peut alors être représenté, à l'aide de la fonction `linmod` de la boîte à outils `AutoSLI`, en première approximation, par la représentation d'état suivante :

```

> linmod(F,G,X,U,etat_0,u_0,'sys'):print(sys);

```

$$\begin{bmatrix}
\frac{-\frac{1}{2} \frac{K_e Y_{e0}}{\sqrt{\alpha P - H_{10}}} - \%4 - \frac{1}{2} \%3}{S_1} & \frac{-\%4 + \frac{1}{2} \%3}{S_1} \\
\frac{\%4 + \frac{1}{2} \%3}{S_2} & \%4 - \frac{1}{2} \%3 - \frac{1}{2} \frac{K_s Y_{s0}}{\sqrt{H_{20} + H_0} \sqrt{\%1}}
\end{bmatrix},
\begin{bmatrix}
\frac{K_e \sqrt{\alpha P - H_{10}}}{S_1} & 0 \\
0 & -\frac{K_s \sqrt{H_{20} + H_0}}{\sqrt{\%1}} + \frac{K_s^3 Y_{s0}^2 \sqrt{H_{20} + H_0} f}{\%1^{(3/2)}}
\end{bmatrix},
\begin{bmatrix}
1 & 0 \\
0 & \frac{1}{2} \frac{K_s Y_{s0}}{\sqrt{H_{20} + H_0} \sqrt{\%1}}
\end{bmatrix},
\begin{bmatrix}
0 & 0 \\
0 & \frac{K_s \sqrt{H_{20} + H_0}}{\sqrt{\%1}} - \frac{K_s^3 Y_{s0}^2 \sqrt{H_{20} + H_0} f}{\%1^{(3/2)}}
\end{bmatrix},$$

```

%1 := 1 + K_s^2 Y_s0^2 f
%2 := |H10 - H20|
%3 := \frac{K_f \text{signum}(H_{10} - H_{20}) \text{abs}(1, H_{10} - H_{20})}{\sqrt{\%2}}
%4 := K_f \text{signum}(1, H_{10} - H_{20}) \sqrt{\%2}

```

**Remarque :** Dans les expressions ci-dessus :  $\text{abs}(1, x)$ ,  $\text{signum}(x)$  et  $\text{signum}(1, x)$  désignent respectivement la dérivée de  $\text{abs}(x)$ , la fonction  $\text{sign}(x)$  et sa dérivée.

Sur cet exemple, il est aussi demandé aux étudiants de calculer la matrice de transfert de façon à comparer à la représentation d'état. Une brève analyse de ces 2 formes de représentation montre qu'il est beaucoup plus simple de travailler avec une forme d'état qu'avec une matrice de transfert.

D'un point de vue général, les expressions analytiques – des résultats retournés par cette boîte à outils `AutoSLI` – permettent de garder tout le sens physique aux équations et de conserver leur homogénéité dimensionnelle. En d'autres termes, les équations symboliques portent intrinsèquement la généralité qu'ont perdu les expressions numériques. Ainsi, elles garantissent une multiplicité d'interprétation sans jamais reconduire les calculs.

Néanmoins, il est clair que, sans l'aide du logiciel de calcul formel, ces manipulations sont fastidieuses et les résultats restent incertains.

**Remarque :** A noter que certaines expressions renvoyées par les fonctions testées pourraient être simplifiées ou factorisées pour être présentées au goût de l'utilisateur.

#### 4. BILAN PÉDAGOGIQUE

Après deux ans d'expérience, dans le cursus de la licence de Sciences et Technologies, mention «Électronique, Électrotechnique, Automatique et Réseaux (EEAR)», le bilan pédagogique qui suit peut être dressé.

L'usage du calcul formel en automatique est encore très insuffisant voire embryonnaire, en ce qui concerne notre filière EEAR. Cet enseignement n'est qu'une initiation et représente un volume horaire de 10 heures. Rappelons, pour mémoire, que depuis 1995 dans l'enseignement des classes préparatoires scientifiques, l'apprentissage d'un logiciel de calcul symbolique a été introduit. Aussi, pour que les étudiants des disciplines de l'EEAR soient formés à l'usage d'un système de calcul formel, l'enseignement de l'automatique doit encore indubitablement changer pour prendre en compte les progrès de ces systèmes. Même, s'il est vrai qu'en une vingtaine d'années, cette discipline s'est considérablement transformée, de façon à tirer profit des évolutions technologiques et des développements des logiciels de simulation numérique. Si ces derniers ont été abondamment insufflés pour rendre plus efficiente l'automatique, les systèmes de calcul symbolique devraient, eux aussi, contribuer à traiter analytiquement des problèmes de plus en plus complexes, ouvrant ainsi de nouveaux champs d'investigation. La boîte à outils de calcul symbolique `AutoSLI`, présentée dans cet article, apporte une modeste contribution à cette évolution.

Pour que l'étudiant automaticien puisse tirer profit de cet outil, il doit : être avisé de l'arsenal mathématique complexe dissimulé derrière la simplicité apparente du logiciel ; être conscient de ses limites et ses déficiences ; garder un sens critique vis à vis des résultats fournis. Il est indéniable que la conquête d'un tel outil passe par une maîtrise des théories mathématiques employées ce qui n'est pas sans difficulté pour nos étudiants. L'exploitation efficace de tels logiciels oblige, notamment, l'apprenant à spécifier formellement l'ensemble des hypothèses implicites qu'il effectue, subrepticement, lorsqu'il conduit, à la main, un calcul ; citons : la positivité d'une variable, d'une indéterminée ou d'un paramètre ou encore la causalité, ... De plus, une «non simplification» de termes, qui paraît flagrante, l'oblige à réfléchir !

Désormais, grâce à cette bibliothèque l'étudiant peut se consacrer complètement à l'interprétation des résul-

tats, et ne plus focaliser toute son énergie aux manipulations mathématiques que recèle inévitablement notre discipline. Des solutions analytiques sûres peuvent être fournies en évitant les problèmes liés à la manipulation fastidieuse d'imposantes expressions. Néanmoins, l'introduction de nos nouveaux outils logiciels induit, de manière quasi-inévitablement : la maîtrise de nouvelles connaissances comme les notions d'affectation, de substitution, d'évaluation, ... ; la manipulation de nouveaux objets comme les expressions, les listes, les ensembles, ... ; ou encore, elle oblige à un apprentissage syntaxique minimal. Notons encore qu'une des difficultés majeures qu'éprouvent les apprenants est relative à la domestication des fonctions de réécriture. Ces fonctions, qui permettent à l'utilisateur d'obtenir des expressions sous la forme désirée, ne sont pas toujours aisées à dominer pour aboutir à l'expression attendue. Il est vrai que la notion de mise en forme d'un résultat étant très relative, cette opération est souvent fastidieuse et laborieuse, voire beaucoup plus simple à effectuer manuellement !

Par ailleurs, une question légitime se pose : Quels sont les dangers liés à l'introduction de tels outils ? Un usage intensif d'un logiciel de calcul formel dans l'enseignement de l'automatique pourrait conduire à la dégradation des connaissances et savoir-faire mathématiques de base. Cette assertion n'est pas aussi certaine qu'elle pourrait paraître ! *A contrario*, le renoncement à ces outils, fruits des mutations technologiques, constituerait un frein au développement de l'automatique. Ce refus rendrait marginales les disciplines de l'EEA par rapport à d'autres qui ont fait le choix de les intégrer (Cf. le paragraphe 1). L'emploi de ces systèmes contribue à une transformation des savoirs et savoir-faire. Par exemple, il n'est pas nécessaire d'être, simultanément, capable de traduire le problème posé sous forme d'équations et de posséder toutes les techniques de résolutions d'un système d'équations. La résolution complète du problème peut être «sérialisée» en deux phases : l'écriture puis la résolution. La dernière phase, plus technique, pouvant être facilement confiée au logiciel de calcul symbolique qui prendra soin de trouver la ou les solutions. Dans le même état d'esprit, les scientifiques, y compris les automaticiens, ont, depuis plus de deux décennies, abandonné le calcul et la simulation numériques aux calculateurs. Alors, pourquoi ne pourrait-on, avec le développement des logiciels, déléguer les calculs symboliques aux machines ?

Bien que ces outils aient connu un essor formidable – par le développement et l'introduction de nouveaux algorithmes (par exemple, grâce à l'utilisation des bases de Gröbner) augmentant leurs possibilités ou améliorant leurs performances – il est clair que la catégorie de problèmes pour laquelle ils sont efficaces, est limitée ; les problèmes résolubles peuvent être qualifiés de

«triviaux».

L'analyse, des difficultés rencontrées par les apprenants, conduit à penser que cet enseignement doit être initié en fin de cursus de licence et approfondi dans un cycle de master. En effet, la plus grande maîtrise de l'outil informatique et des concepts fondamentaux de l'automatique sont deux ingrédients indispensables qui devraient augmenter les profits tirés du calcul symbolique et de cette boîte à outils.

**Remarque :** Cette bibliothèque est disponible, à des fins non commerciales, sur simple demande.

## REFERENCES

- [1] Maplesoft. *Maple 9 Getting Started Guide*. Maplesoft, Waterloo, ON, Canada N2L 6C2, 2003.
- [2] Maplesoft. *Maple 9 Learning Guide*. Maplesoft, Waterloo, ON, Canada N2L 6C2, 2003. Based in part on the work of B. W. Char.
- [3] M. B. Monagan, K. O. Geddes, K. M. Heal, G. Labahn, S. M. Vorkoetter, J. McCarron, et P. DeMarco. *Maple 9 Introductory Programming Guide*. Maplesoft, Waterloo, ON, Canada N2L 6C2, 2003.
- [4] M. B. Monagan, K. O. Geddes, K. M. Heal, G. Labahn, S. M. Vorkoetter, J. McCarron, et P. DeMarco. *Maple 9 Advanced Programming Guide*. Maplesoft, Waterloo, ON, Canada N2L 6C2, 2003.
- [5] Patrick Sibille. Un outil de calcul symbolique au service de l'enseignement de l'automatique - application aux modèles linéaires. *Colloque sur l'Enseignement des Technologies et des Sciences de l'Information et des Systèmes (CETISIS'2005)*, 25-27 octobre 2005.
- [6] William C. Bauldry et Joseph R. Fiedler. *Calculus Laboratories with Maple : A Tool, Not an Oracle*. Brooks/Cole, Pacific Grove, CA, USA, 1991.
- [7] William C. Bauldry, Benny Evans, et Jerry Johnson. *Linear Algebra with Maple*. John Wiley, New York, NY, USA, 1995.
- [8] C. Gomez, B. Salvy, et P. Zimmermann. *Calcul Formel : Mode d'Emploi. Exemples en Maple*, volume 13 of *Logique mathématiques informatique*. Masson Editeur, Paris, France, 2e tirage avec mise à jour Maple V.4 edition, 1995.
- [9] André Heck. *Introduction to Maple*. Springer-Verlag, Berlin, Germany / Heidelberg, Germany / London, UK / etc., second edition, 1996.
- [10] Martha L. Abell et James P. Braselton. *Differential equations with Maple V*. Academic Press, New York, NY, USA, second edition, 2000.
- [11] Robert M. Corless. *Essential Maple 7 : an introduction for scientific programmers*. Springer-Verlag, Berlin, Germany / Heidelberg, Germany / London, UK / etc., 2002.

- [12] William E. Baylis. *Theoretical Methods in the Physical Sciences : an introduction to problem solving using Maple V*. Birkhäuser, Cambridge, MA, USA ; Berlin, Germany ; Basel, Switzerland, 1994.
- [13] K. Chiang et H. T. Man. *Maple for Physicists*. Springer-Verlag, Berlin, Germany / Heidelberg, Germany / London, UK / etc., 1996.
- [14] Richard H. Enns et George McGuire. *Nonlinear Physics with Maple Files and Experiments*. Birkhäuser, Cambridge, MA, USA ; Berlin, Germany ; Basel, Switzerland, 1997.
- [15] Richard H. Enns et George McGuire. *Nonlinear physics with Maple for scientists and engineers*. Birkhäuser, Cambridge, MA, USA ; Berlin, Germany ; Basel, Switzerland, second edition, 2000.
- [16] N. I. Ioakimidis et E. G. Anastasselou. Computer-based manipulation of systems of equations in elasticity problems with Gröbner bases. *Computer Methods in Applied Mechanics and Engineering*, 110(1-2) :103–111, 1993.
- [17] Bruce W. Char et Mark F. Russo. Automatic identification of time scales in enzyme kinetics models. *ISSAC '94 : Proceedings of the 1994 International Symposium on Symbolic and Algebraic Computation : July 20–22, 1994, Oxford, England, United Kingdom*, pages 74–83, 1994.
- [18] Ronald L. Greene. *Classical mechanics with Maple*. Springer-Verlag, Berlin, Germany / Heidelberg, Germany / London, UK / etc., 1995.
- [19] R. Taylor et K. Atherley. Chemical engineering with Maple. *CEE. Chemical engineering education*, 29(1) :56–66, 1995.
- [20] Bill Scott. *Maple for environmental sciences : a helping hand*. Springer-Verlag, Berlin, Germany / Heidelberg, Germany / London, UK / etc., 2001.
- [21] Christopher Tocci et Steven G. Adams. *Applied Maple for engineers and scientists*. The Artech House computer science library series. Artech House Inc., Norwood, MA, USA, 1996.
- [22] Jack-Michel Cornil et Philippe Testud. *Maple V Release 4 : introduction raisonnée à l'usage de l'étudiant, de l'ingénieur et du chercheur*. Springer-Verlag, Berlin, Germany / Heidelberg, Germany / London, UK / etc., 1997.
- [23] David I. Schwartz. *Introduction to Maple for engineers and scientists*. Prentice-Hall, Englewood Cliffs, NJ 07632, USA, 1999.
- [24] Singiresu S. Rao. *Applied Numerical Methods for Engineers and Scientists*. Prentice-Hall, Englewood Cliffs, NJ 07632, USA, 2002.
- [25] Bram de Jager. Use of symbolic computation in nonlinear control : Is it viable? *IEEE Transactions on Automatic Control*, 40(1) :84–89, January 1995.
- [26] A. B. Ogunye et A. Penlidis. Computation of system gramians and balanced realizations using Maple V. *International Journal of Systems Science*, 26(4) :899–926, 1995.
- [27] A. B. Ogunye. Process control and symbolic computation : an overview with Maple V. *MapleTech*, 3(1) :94–103, 1996.