Applied Mathematical Sciences, Vol. 1, 2007, no. 12, 551 - 570

A Matlab-Based Software to Obtain the Time Course Equations of Dynamic Linear Systems

E. Arias[(a)], M. Picazo[(a)], F. Garcia-Sevilla[(b)], A. Alberto[(c)], E. Arribas[(d)] and R. Varon[(b,*)]

[a] *Computer Systems Department, University of Castilla-La Mancha, Albacete, Spain*

[b] *Chemical Physics Department, University of Castilla-La Mancha, Albacete, Spain*

[c] *Albacete Research Institute of Informatics, University of Castilla-La Mancha, Albacete, Spain.*

[d] *Applied Physics Department, University of Castilla-La Mancha, Albacete, Spain.*

---

[*] Corresponding author, Chemical Physics Department, University of Castilla-La Mancha, Avenida de España s/n, E-02071 Albacete, Spain. Tel.:+34 967599200, ext. 2480; Fax:+34 967599224; e-mail: ramon.varon@uclm.es

Abstract. Recently [1] a general solution of the set of differential equation in order to describe time invariant linear dynamic systems has been derived. In this paper software, easy to use and with a user-friendly format for the input of data and output of results, which is able to give the above general solution, has been implemented. The implementation is based on Matlab^{TM} [2], widely used scientific software. The algorithms and methodology used are of immediate application to any dynamic linear system to obtain the time course of any of its state variables. Finally, we apply the results, as an example, to a biological important example of dynamic linear systems: the enzyme systems, including the zymogen activation.

Mathematics Subject Classification: 34A30

Keywords: Linear, dynamic systems, software, graphic interface, differential equations

1. INTRODUCTION

The kinetic behavior of a dynamic, linear time invariant systems is described by a set of ordinary, linear, with constant coefficients, differential equations. This set of differential equations can be non-homogeneous or homogeneous. If the set is non-homogeneous then its solution is a linear combination of the general solution corresponding to the set assumed homogeneous and the particular solution of the non-homogeneous set. When the set of differential equations is homogeneous, as in the kinetic analysis of tracers or in enzyme kinetics, its solution coincides with the general solution. Recently Varon et al. [1] derived, by using the Laplace transformation method and some others algorithms developed by these authors, the general solution assuming that the non-null eigenvalues of the system matrix are simple, i.e. not repeated, what is the most frequent practical situation. This general solution consists in a sum of a polynomial part, corresponding to the null eigenvalues of the matrix of the system, and a multiexponential one, corresponding to the non-null eigenvalues. In the mentioned reference the coefficients involved in the polynomial part in terms of the entries of the matrix of the system better than in terms of the non-null eigenvalues were given. The results obtained can be applied to any time-invariant linear dynamics system. Biological important examples of dynamic linear systems are those including enzyme catalyzed reactions in which the interconversions between the different enzyme forms involved are of first or pseudofirst order.

In this contribution, as an extension of contribution in reference [1], we implement a program, DYNSYS named, which is able to give us the time course of any of the *n* state variables involved in a dynamic linear time invariant system due to a non-zero initial value of any of the *n* state variables. This software, valid for any dynamic linear time invariant system, is applied to a concrete example of a biological important example of dynamic linear time invariant systems: the enzyme systems, including the zymogen activation, in which the connections between the different enzyme forms involved in its reaction mechanism are of first or pseudofirst order.

2. MATERIALS AND METHODS

To implement the software called DYNSYS we have used Matlab 5.3 programming language for Windows. The examples described in this article have been solved on a computer based on a Pentium III 600 MHz with 128 Mbytes of RAM and 20 Gbytes IDE hard disk.

3. GENERAL SOLUTION

In this section, in order to readability of the present contribution, a summary of previous results obtained for the general solution of the set of differential equations refered in [1].

Let y_1, y_2,…,y_n be the *n* state variables of a system related by means of a set of *n* different ordinary, first order, linear, with constant coefficients differential equations and **K** the corresponding square *n* x *n* matrix of the system, i.e.:

$$\mathbf{K} = \begin{bmatrix} K_{1,1} & K_{2,1} & \cdots & K_{n,1} \\ K_{1,2} & K_{2,2} & \cdots & K_{n,2} \\ \vdots & \vdots & & \vdots \\ K_{1,n} & K_{2,n} & & K_{n,n} \end{bmatrix} \qquad (1)$$

In the following we will denote as *c* an integer number such that $0 \le c < n$ and as u another integer number such that $1 \le u \le n$. Between c, u and n the following relationship exists: $c + u \le n$.

If it is assumed that only one of the initial values y_i^0 (*i*=1,2,….,*n*) is non-null, e.g. y_e^0 (*e* being one of the subscripts 1,2,…,*n*), the time course of y_f (f =1,2,..., *n*) is given, when c >0, by:

$$y_f = A_0 + A_1 t + \cdots + A_{c-1} t^{c-1} + B_1 e^{s_1 t} + B_2 e^{s_2 t} + \cdots + B_u e^{s_u t} \qquad (c > 0) \qquad (2)$$

and when c = 0 by:

$$y_f = B_1 e^{s_1 t} + B_2 e^{s_2 t} + \cdots + B_u e^{s_u t} \qquad (c = 0) \tag{3}$$

In Eqs. (2) and (3) s_1, s_2, …s_u non-null quantities which are the zeros of the following polynomial F(s):

$$F(s) = s^u + F_1 s^{u-1} + \cdots + F_u \tag{4}$$

where the coefficients Fj (j=1,2,...,u) are sums of terms each of them being a product of j different entries $K_{i,j}$'s of matrix K. Hence, in Eqs. (2) and (3):

$$B_h = \frac{(-1)^{u-1} N(s_h) y_e^0}{s_h^c \prod_{\substack{p=1 \\ p \neq h}}^{u} \left(s_p - s_h\right)} \qquad (h=1,2,\ldots,u) \tag{5}$$

where $N(s_h)$ is given by:

$$N(s_h) = G_0 s_h^w + G_1 s_h^{w-1} + \cdots + G_w \tag{6}$$

w being an integer number lying between 0 and c+u-1, i.e. $0 \leq w \leq c+u-1$ and the coefficients G_0, G_1,....G_w are functions of the entries $K_{i,j}$'s of the matrix K.

The coefficients A_0, A_1, A_{c-1} involved in Eq. (2) are given by:

$$A_0 = y_f^0 + (-1)^u y_e^0 \left\{ \sum_{j=0}^{w} G_j \left[\sum_{h=1}^{u} \frac{1}{s^{c-w+j} \prod_{\substack{p=1 \\ p \neq h}}^{u} \left(s_p - s_h\right)} \right] \right\} \qquad (\, y_f^0 = 0 \text{ if } f \neq e \,) \tag{7}$$

$$A_q = \frac{1}{q!}\left(\frac{d^q y_f}{dt^q}(0) + (-1)^u y_e^0 \left\{\sum_{j=0}^{w} G_j \left[\sum_{h=1}^{u} \frac{1}{s^{c-w-q+j}\prod\limits_{\substack{p=1\\ p\neq h}}^{u}\left(s_p - s_h\right)}\right]\Bigg]\right\}\right) \quad (q = 1,2,\ldots,c\text{-}2) \qquad (8)$$

$$A_{c-1} = \frac{G_w y_e^0}{F_u} \qquad (9)$$

In Eq. (8) $\frac{d^q y_f}{dt^q}(0)$ (f = 1,2,….,n; q = 1,2,…) is the value at t = 0 of the instantaneous q-th derivative with respect time of $y_{f.}$ Therefore, the value of $\frac{d^q y_f}{dt^q}(0)$ (f = 1,2,….,n; q=1,2,…) when only the state variable y_e (e being one of the values 1,2,…,n) is non-null at t=0, i.e. when, as assumed in this paper, $y_e^0 \neq 0$ and $y_j^0 = 0$ (j=1,2,…,n; $j \neq e$) can be merely obtained by multiplying by y_e^0 the entry of matrix $\mathbf{K}^q$ on f-th row and k-th column *(i*=1,2,…,*n),* $\mathbf{K}^q$ (q=1,2,…) being the *qth*-power of the square n x n matrix $\mathbf{K}$.

4. SOFTWARE TO OBTAIN THE TIME COURSE OF $\mathbf{y_f}$

Based on the analysis carried out in [1], we have developed a software, we call as DYNSYS running in Windows, which give the parameters and coefficients involved in Eq. (21) of [1] corresponding to the time course of the state variable y_f of any dynamic linear time invariant system with input zero due to a initial value of y_e equal to y_e^0. This program, which is available to the interested readers on demand, has an easy input of data, process rapidly and results are collected in an output file which can be handled with any text processor apart of its visualization on the screen. As far as we know, this is the first computer program with this objective giving the results with the security that no error are possible in the process. The only error source is that resulting of mistakes in the input of the data.

4.1. Software development

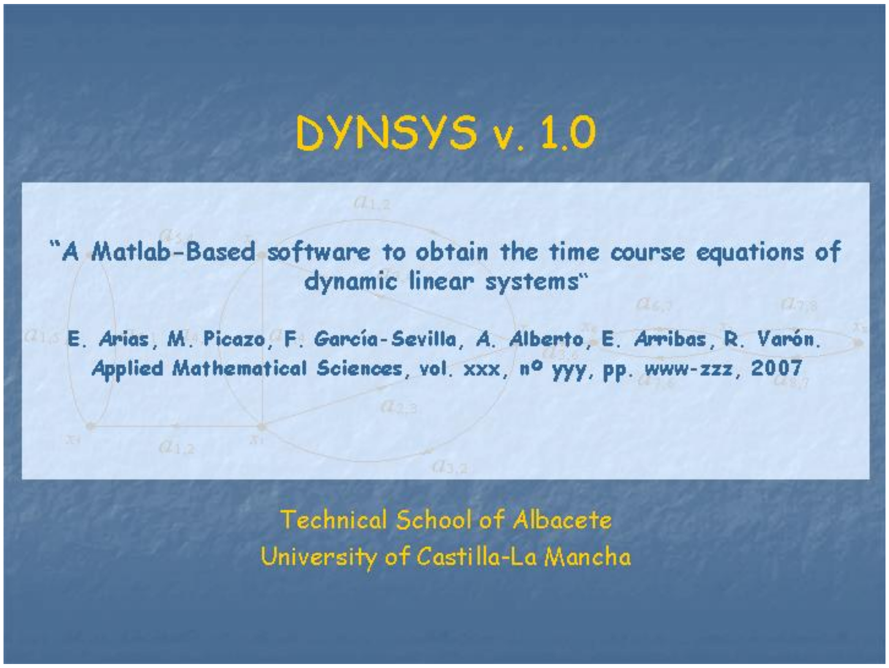
The computer program has been written in Matlab (**Mat**rix **Lab**oratory) programming language, version 5.3, under Windows (32 bits) Operating System. This software has been developed in Matlab due to the fact that:

- Matlab is a user-friendly environment
- Matlab is widely used by the scientific community
- Matlab provides a lot of toolboxes of interest for this work: numerics, graphics, symbolic, graphical user interfaces, etc.
- Matlab provides portability of the implementation, that is, the resulting implementation could be run, without a few o no modifications, on different Operating Systems such a Windows and Linux.
- Compatibility due to the standardization of Matlab implementation, it is possible used the developed program with different version of Matlab, if the necessary toolboxes are includes in it.

According to the previous good features, the resulting code consists on a Graphical User Interface (GUI) through which the user interacts easily with the program (see Figure 1, the cover of the program and Figure 2, the presentation window).

In the presentation window the user has some information related to the DYNSYS program: How it functions, some details of implementation, etc.

By pressing the *Continue* button, the user will choose the problem to solve (to obtain the time course equations), will introduce the desired input data, and will obtain the required results in a file and in the different windows. A demo of that will be shown at the example section.



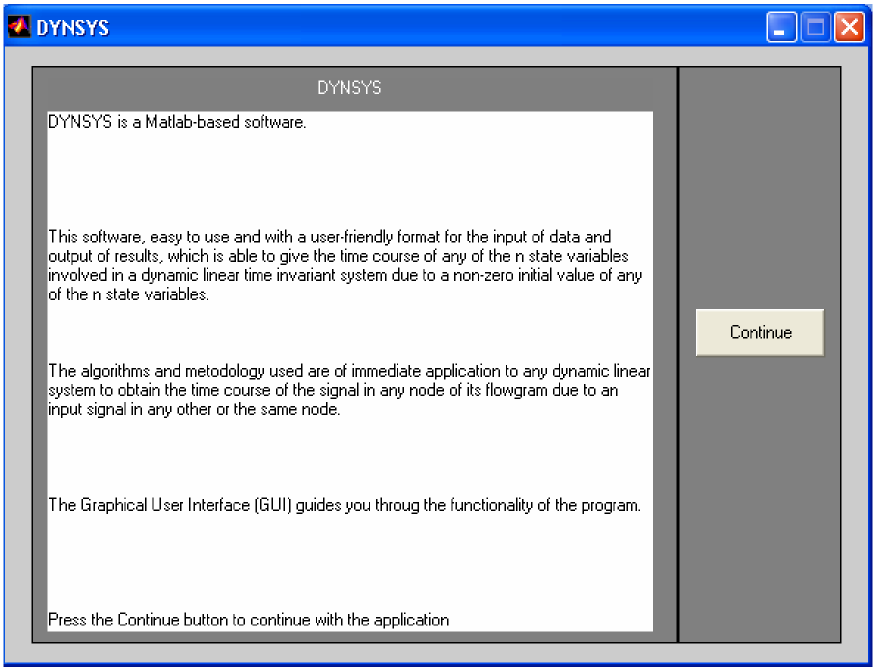
DYNSYS v. 1.0

"A Matlab-Based software to obtain the time course equations of dynamic linear systems"

E. Arias, M. Picazo, F. García-Sevilla, A. Alberto, E. Arribas, R. Varón.
Applied Mathematical Sciences, vol. xxx, nº yyy, pp. www-zzz, 2007

Technical School of Albacete
University of Castilla-La Mancha

Figure 1: Cover of DYNSYS program



DYNSYS

DYNSYS is a Matlab-based software.

This software, easy to use and with a user-friendly format for the input of data and output of results, which is able to give the time course of any of the n state variables involved in a dynamic linear time invariant system due to a non-zero initial value of any of the n state variables.

The algorithms and metodology used are of immediate application to any dynamic linear system to obtain the time course of the signal in any node of its flowgram due to an input signal in any other or the same node.

The Graphical User Interface (GUI) guides you throug the functionality of the program.

Press the Continue button to continue with the application

Figure 2: Presentation window

In this implementation the numeric and symbolic toolboxes have been used as convenience. This suitable combination of both toolboxes is transparent to the user who uniquely sees a window with the GUI. Also, the results are shown in the GUI or stored in a file.

With respect the functionality of the DYNSYS program, the main characteristics of the present version are: (a) It admits systems of differential equations with high number of equations and with high number of variables. In fact, due to the implementation of the algorithm, the limits are fixed by the disk space available on the computer; (b) This program needs a large amount of memory to store temporal data. Because, if the user has to solve systems with many equations, it is necessary to use NTFS as file system of the hard disk, because the length of a file in a FAT32 file systems is limited to 2 Gbytes; (c) The computation time is very short, even for complex systems. Obviously, this time depends on the computer used. Thus, with the computer described in “Systems and Methods” section, the computer time was less than 60 seconds for all examples we have tried; (d) The input of the data is very easy and versatile and; (e) It provides the results in the most simplified form.

The hardware requirements are:

CPU: 80386 or higher

Memory: This program needs only 1 MByte of free memory but, of course, it is necessary at least 16 or 32 MBytes of physical memory for the operating system.

Hard disk: The required free space depends on the complexity of the graph under study.

How we said previously, DYNSYS program allows the user to obtain the time course (Figure 3)

4.2. Description of the software

Input of the data

The input data given by the user for obtaining the time course of y_f are:

- The number of state variables of the system, *n*.

- The expression of the entries $K_{i,j}$ (*i,j=1,2,…,n*), being *K* the system matrix. The entries $K_{i,i}$'s (*i*=1,2,…,*n*) must be entered in terms of the $K_{i,j}$'s (j=1,2,…,n; $j \neq i$).

- The state variable whose time course has to be known (identified through the subindex *f*) and the state variable which initial values is different from zero (identified through the subindex *e*)

Output of the results

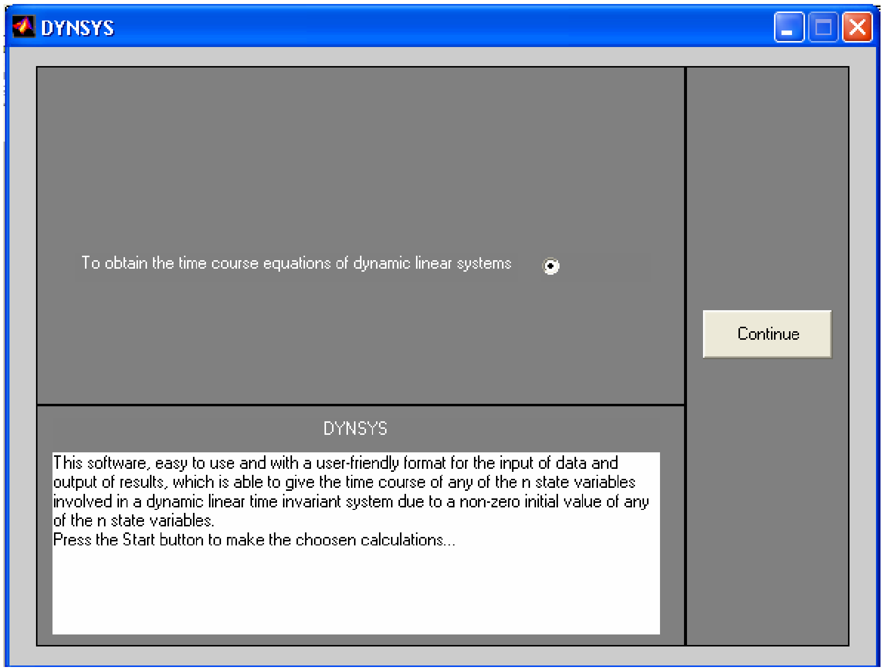
From the previous input data the software presented in this paper:

1. Determines the quotient x_f / y_e^0 = *Y*(*s*)/*U*(*s*) (Eq. (8) in reference [1]).

2. Divides both the numerator and denominator of the transfer function by b_0, being b_0 the coefficient of the term with higher degree of the polynomial *U(s)*, obtaining an equivalent expression x_f / y_e^0 = *N(s)/D(s)* (Eq. (10) in [1]).

3. Rearranges the denominator *D(s)* according to the Eq. (14) in [1].

4. Finally calculates, c >2, the powers $\mathbf{K}^q$ (*q*=1,…,*c*-2) of the system matrix *K* are calculated to be able to determine the values of $\frac{d^q y_f}{dt^q}(0)$ (*q=1,…,c-2*) by multiplying the entry ($K_{e,f}^{(q)}$) on the *f*-th row and *e*-th column of matrix $\mathbf{K}^q$) by y_e^0 .



To obtain the time course equations of dynamic linear systems ◉

This software, easy to use and with a user-friendly format for the input of data and output of results, which is able to give the time course of any of the n state variables involved in a dynamic linear time invariant system due to a non-zero initial value of any of the n state variables.
Press the Start button to make the choosen calculations...

Figure 3: Selection of problem to solve

According to the previous calculations, the results are:

1. The value of the integer number w and the expressions of the coefficients G_i, $i=0\ldots w$.
2. The values of the integer numbers c and u and the expressions of the coefficients F_i $(i=1,2,\ldots,u)$ in terms of the transfers $K_{i,j}$'s $(i,j=1,2,\ldots,n;\ i \neq j)$.
3. The concrete expression for the case under study of the general Eq. (21) in [1].

4. If c >2, the expressions of $\frac{d^q y_f}{dt^q}(0)$ ($q=1,\ldots,c-2$) needed to have the expressions of $A_1, A_2,\ldots, A_{c-2}$.
5. Finally, for easy, the program shows the expressions entered of the transfers $K_{i,j}$'s ($i,j=1,2,\ldots,n; i \neq j$)

From this information in the output file, the user has the needed data to determine the expressions of the roots $s_1, s_2, \ldots s_u$ [roots of polynomial F(s)] and the coefficients A_0, $A_1,\ldots A_{c-1}$, B_1, $B_2,\ldots B_u$ involved in Eq. (21) in [1] (from Eqs. (29), (34), (25) and (27) in [1]) and thus, to have the complete expression of the time course y_f..

Example

Next we apply the above software to some dynamic linear time-invariant system consisting in an enzyme system involving zymogen activation. The above analysis and the corresponding software can be applied to any dynamic linear systems system which can be described by a set of ordinary, first order, linear differential equations with constant coefficient like that given by Eq. (1) in [1].

We desire to estudy a simple enzyme system corresponding to the activation of a zymogen or proenzyme into its active form by the action of a protease with release of one or more peptides according to the reaction mechanism shown in Scheme 1 where *Z* is the zymogen, *E* the activating enzyme, *EZ* the complex activating enzyme-zymogen, E_a the activated enzyme and *W* one or more peptides.

$$Z + E \underset{k_{-1}}{\overset{k_1}{\rightleftarrows}} EZ \xrightarrow{k_2} E + E_a + W$$

Scheme 1

If we consider that the initial zymogen concentration is very higher than the initial activating enzyme concentration and we consider a reaction time for which the concentration of E_a is much less than the initial concentration of *Z*, then we can consider that during this reaction time the zymogen concentration remains approximately equal to its initial one and, therefore, the kinetic behavior of the enzyme forms *E*, *EZ* and E_a, involved in the above enzyme system can be described by the following set of linear differential equations:

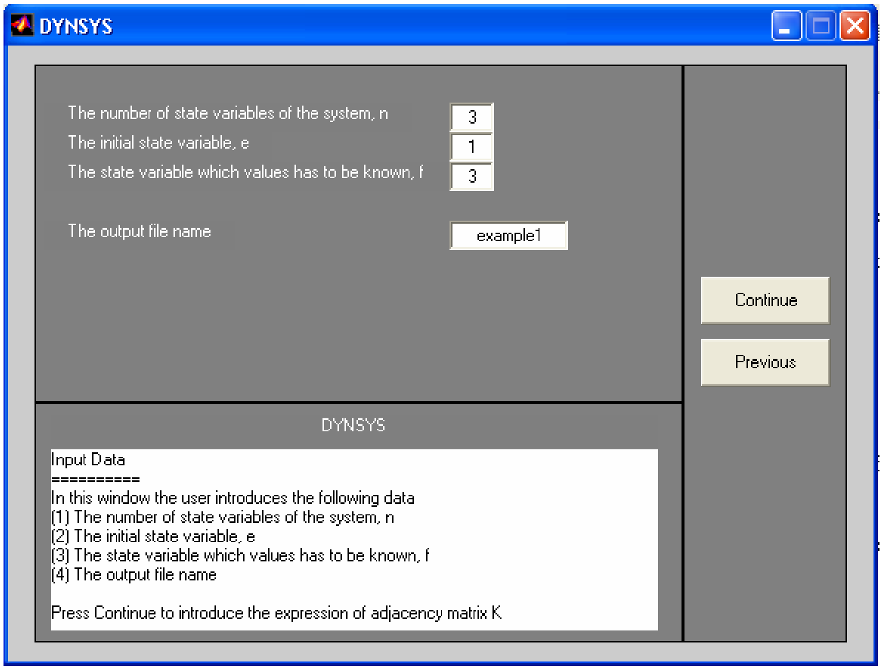
$$\left.\begin{aligned} \frac{dy_1}{dt} &= K_{1,1}y_1 + K_{2,1}y_2 \\ \frac{dy_2}{dt} &= K_{1,2}y_1 + K_{2,2}y_2 \\ \frac{dy_3}{dt} &= K_{2,3}y_2 \end{aligned}\right\} \tag{10}$$

where y_1, y_2 and y_3 are the instantaneous concentrations of E, EZ and E_a, respectively, t is the time and $[Z]_0$ is the initial zymogen concentration and the transfer coefficients are:

$$\left.\begin{aligned} K_{1,1} &= -K_{1,2} \\ K_{1,2} &= k_1[Z]_0 \\ K_{2,2} &= -K_{2,1} \\ K_{2,1} &= k_{-1} + k_2 \\ K_{2,3} &= k_2 \end{aligned}\right\} \tag{11}$$

Let us to know the time course of y_3 due to the initial value y_1^0 of y_1. The data we must enter in the software are *n=3*, *e*=1, *f* =3 and the transfers in Eq. (11). The software window to introduce the data is shown in Figure 4.

Note that the data are introduced in *edit-box*. Once the data are introduced the user has to press *continue* button to introduce the expressions of matrix *K* as in Eq. (11). (see Figure 5). If some mistake has been introduced, the user can press *previous* button to rectify (the meaning of this button is the same in all the windows). To introduce the expressions of matrix *K*, the software asks the user entry-by-entry the matrix *K*. The entries $K_{i,i}$'s (i=1,2,…,n) must be entered as in Eq. (11), taking into account that the subindex is not considered and that superindex is represented as "^". In Figure 5, the element *K(1,2)* is required to the user. After introducing this element and pressing *next*, the user can introduced the element *K(1,3)*. Once all the elements of matrix *K* are introduced, the user has to press *continue* button.



The number of state variables of the system, n 3
The initial state variable, e 1
The state variable which values has to be known, f 3

The output file name example1

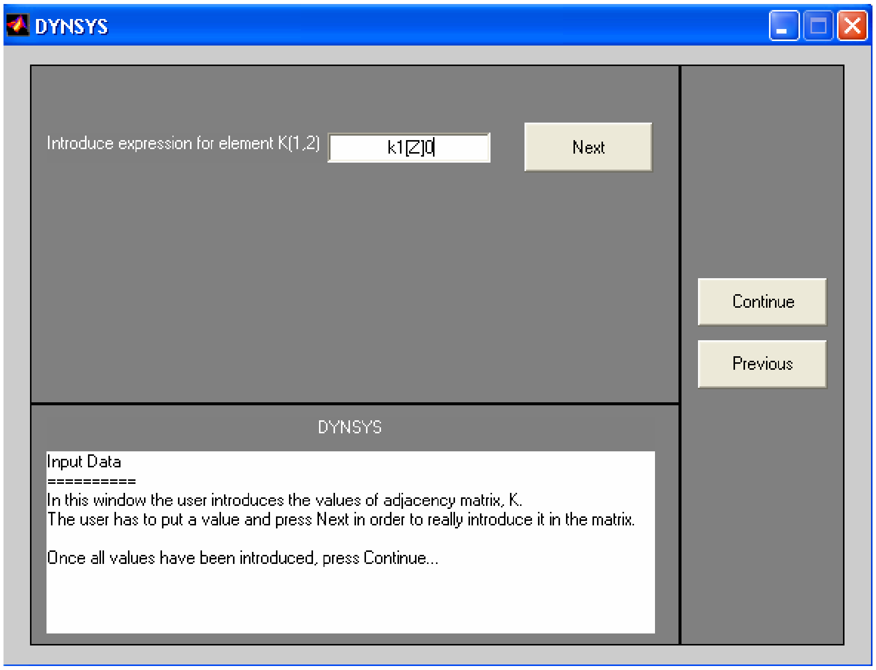
```
Input Data
==========
In this window the user introduces the following data
(1) The number of state variables of the system, n
(2) The initial state variable, e
(3) The state variable which values has to be known, f
(4) The output file name

Press Continue to introduce the expression of adjacency matrix K
```

Figure 4: First Input Data window.

Next window of DYNSYS program is shown in Figure 6. The data introduced in Figure 5 are used in DYNSYS program only as information purpose, but the data introduced in Figure 6 are used in the calculation process. In Figure 6, The entries $K_{i,i}$'s (i=1,2,…,n) must be entered as in Eq. (10), but in terms of $K_{i,j}$'s (j=1,2,…,n; $j \neq i$), e.g. $K_{1,1} = -K_{1,2}$ and $K_{1,2} = K_{1,2}$ as the Figure 6 showns. If some mistake has been introduced, *Previous* button can be used to modify the values, if not, *Continue* button is pressed. In Figure 7, the data introduced by the user which describe the dynamic system under studied are shown. In this way, the user can verify the corretness of the data introduced before continuing with the calculation process.

If some change is required, then the user can use *Modify* or *Previous* buttons. If all data are correct the calculations will start by pressing *Continue* button.

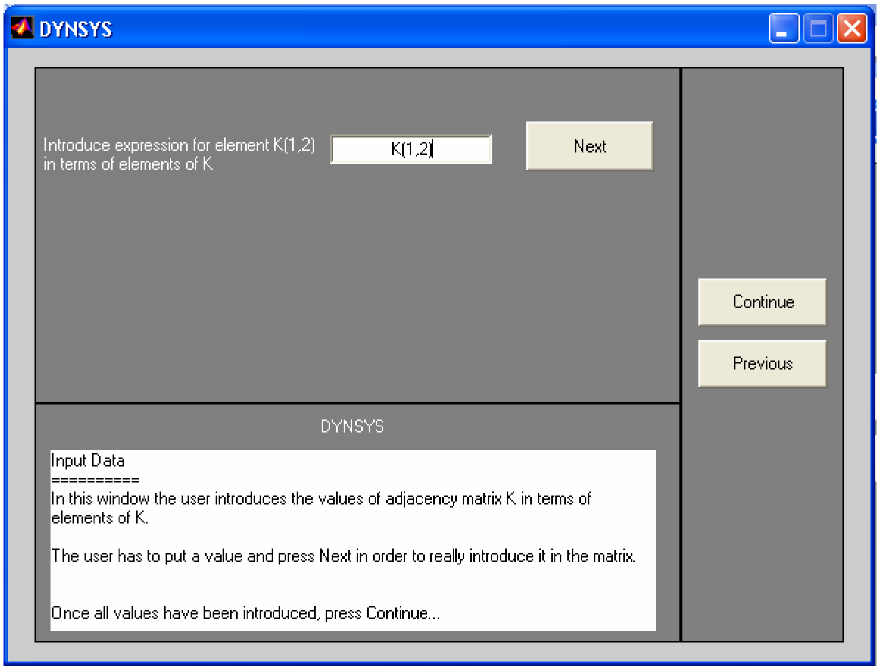


Introduce expression for element K(1,2) k1[Z]0

Input Data
==========
In this window the user introduces the values of adjacency matrix, K.
The user has to put a value and press Next in order to really introduce it in the matrix.

Once all values have been introduced, press Continue...

Figure 5: Introduction of expression of matrix K



Introduce expression for element K(1,2) in terms of elements of K

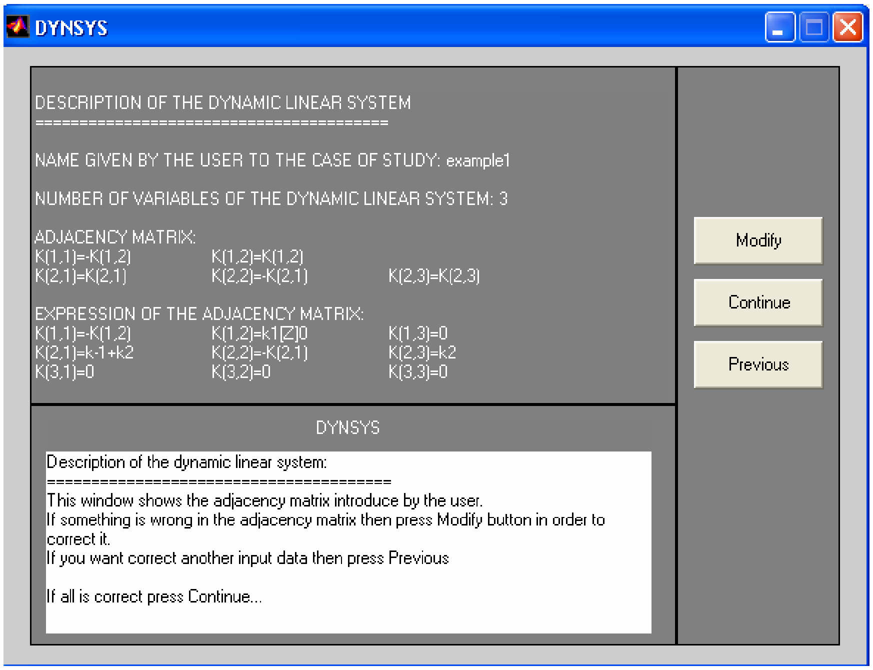
K(1,2)

```
Input Data
==========
In this window the user introduces the values of adjacency matrix K in terms of
elements of K.

The user has to put a value and press Next in order to really introduce it in the matrix.


Once all values have been introduced, press Continue...
```

Figure 6: Introduction of expression of matrix K in terms of matriz K



```
DESCRIPTION OF THE DYNAMIC LINEAR SYSTEM
========================================

NAME GIVEN BY THE USER TO THE CASE OF STUDY: example1

NUMBER OF VARIABLES OF THE DYNAMIC LINEAR SYSTEM: 3

ADJACENCY MATRIX:
K(1,1)=-K(1,2)          K(1,2)=K(1,2)
K(2,1)=K(2,1)           K(2,2)=-K(2,1)          K(2,3)=K(2,3)

EXPRESSION OF THE ADJACENCY MATRIX:
K(1,1)=-K(1,2)          K(1,2)=k1[Z]0           K(1,3)=0
K(2,1)=k-1+k2           K(2,2)=-K(2,1)          K(2,3)=k2
K(3,1)=0                K(3,2)=0                K(3,3)=0
```

```
Description of the dynamic linear system:
========================================
This window shows the adjacency matrix introduce by the user.
If something is wrong in the adjacency matrix then press Modify button in order to
correct it.
If you want correct another input data then press Previous

If all is correct press Continue...
```

Figure 7: Description of the dynamic system introduced

The results the software furnishes are:

$w = 0$ (12)

$c = 2$ (13)

$u = 1$ (14)

$G_0 = K_{1,2}K_{2,3}$ (15)

$F_1 = K_{1,2} + K_{2,1}$ (16)

s_1: root of polynomial $s+F_1$ (i.e. $s_1 = -F_1$)

$$y_3 = A_0 + A_1 t + B_1 e^{s_1 t} \tag{17}$$

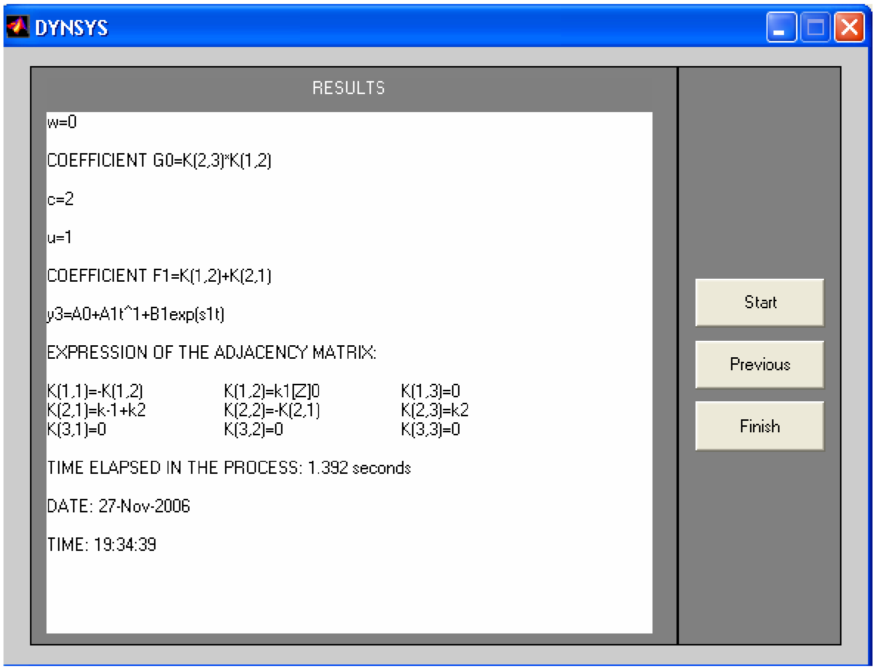
Once we dispose of the information furnished by the software and from Eqs. (25), (27), (29) and (36) in [1], we have:

$$A_0 = -B_1 = -\frac{G_0 y_1^0}{F_1^2} \tag{18}$$

$$A_1 = \frac{G_0 y_1^0}{F_1} \tag{19}$$

The result are shown in Figure 8

The way in which the results are shown in Figure 8, are according to the description of the software section. This results, obviously, corresponds to equation (12)-(17).



```
w=0

COEFFICIENT G0=K(2,3)*K(1,2)

c=2

u=1

COEFFICIENT F1=K(1,2)+K(2,1)

y3=A0+A1t^1+B1exp(s1t)

EXPRESSION OF THE ADJACENCY MATRIX:

K(1,1)=-K(1,2)           K(1,2)=k1[Z]0            K(1,3)=0
K(2,1)=k-1+k2            K(2,2)=-K(2,1)           K(2,3)=k2
K(3,1)=0                 K(3,2)=0                 K(3,3)=0

TIME ELAPSED IN THE PROCESS: 1.392 seconds

DATE: 27-Nov-2006

TIME: 19:34:39
```

Figure 8: Result for example 1

ACKNOWLEDGEMENTS. This work has been also partially supported by grants for the University of Castilla-La Mancha, Project DACIG.

References

[1] Varon R., Picazo M., Alberto A., Arribas E., Masia-Perez J., Arias E., General solution of the set of differential equations describing the time invariant linear dynamics systems. Application to enzyme systems. Applied Mathematical Sciences 1 (2007) 281-300.

[2] MATLAB®, MATLAB® Compiler, and other MATLAB family products U.S. Patent Nos. 6,857,118; 6,973,644; 6,993,772; 7,010,364; 7,051,333; 7,051,338.