

RATRIX : A RAtional matRIX calculator for computer aided analysis and synthesis of linear multivariable control systems

P. Tzekis, N. P. Karampetakis and A.I.G. Vardulakis

*Department of Mathematics, Aristotle University of Thessaloniki
Thessaloniki 54006, Greece, email : karampet@ccf.auth.gr*

Abstract. The main aim of this work is to present an interactive software tool environment for analysis and synthesis for linear multivariable systems using the numeric power and the symbolic algebra capabilities of the MAPLE computer algebra package.

1. Introduction

Tedious, repetitive and complex calculations are involved in many problems in linear algebra and furthermore in control. It is in general beneficial to implement the corresponding problem computationally rather than solving the problem by hand in terms of certain governing factors. Such factors may include the time, accuracy and cost of obtaining such a solution.

The advents of complete symbolic computational packages, such as Maple, Mathematica, MACSYMA, Reduce, MuMath and Scratchpad, have made the computational implication of such problems particularly attractive. The main advantages, in regards to linear algebra, lie in the following.

Symbolic Storage - Variables can be stored in an "exact" form (i.e. $1/3$ as opposed to $0.33333\dots$) resulting in no loss of accuracy during a calculation. The main disadvantage of uses "exact numbers" are : a) the large size of memory they use, and b) the slow speed they have, such kind of programs. Thus we can easily see that symbolic computational programs may be used in case where accuracy is more important than speed and memory.

Additionally variables can be left "unassigned" (i.e.) without holding any numerical values) which enables polynomial operations to be defined, say, in an indeterminate s .

Inbuilt Procedures - The existence of hundreds of inbuilt procedures covering both general and certain specialized areas of mathematics. In Maple, for example, the package *linalg* consist of numerous procedures used in the solution of problems in linear algebra such as smith form and the inverse of a polynomial matrix.

Programming Language - The existence of unique high-level programming languages allowing specific procedures to be written. In Maple over 90% of these inbuilt procedures are written in its own procedural programming language and therefore any of them can be implemented in any new procedures written. They can therefore be viewed upon as *building blocks* for subsequently more specified and advanced procedures.

Perhaps understandably, however, neither one of these symbolic computer packages contain existing procedures or indeed packages for specialized work in the study of rational matrices and its applications in analysis, synthesis and design of control systems. The only exception is the polynomial toolbox of Matlab created by PolyX [7], which has the ability to handle polynomial matrices, using at most numerical algorithms (instead of symbolic). However all the referenced packages are clearly flexible

enough for the implementation of such work to be carried out.

An extra disadvantage of the above packages is that requires the knowledge of the procedures from the user. This however may be considered as an advantage for the experienced user in order to build his own procedures.

In order to overcome the disadvantages proposed in the last two paragraphs, we have developed a program named "RATRIX". The main aim of this program is to permit the user to execute on line control related algorithmic procedures and calculations using a symbolic algebra user friendly windows based application dealing with the use of rational matrices in the study of the analysis and synthesis of linear multivariable systems.

2. Program Description

While the program is written in Visual C++, the main procedures of the program were created in the symbolic package MAPLE, which is used internal in the program.

The main window of the program consists of : a) a menu bar, which gives direct access to the main procedures of the program, b) a toolbar with shortcut icons for the easy creation of algebraic expressions and the easy call of the main procedures of the program, c) a window where the matrices and procedures that we have already used are displayed, d) a window that displays the results of our calculations and e) a window for the syntax and use of known maple procedures (see Figure 1).

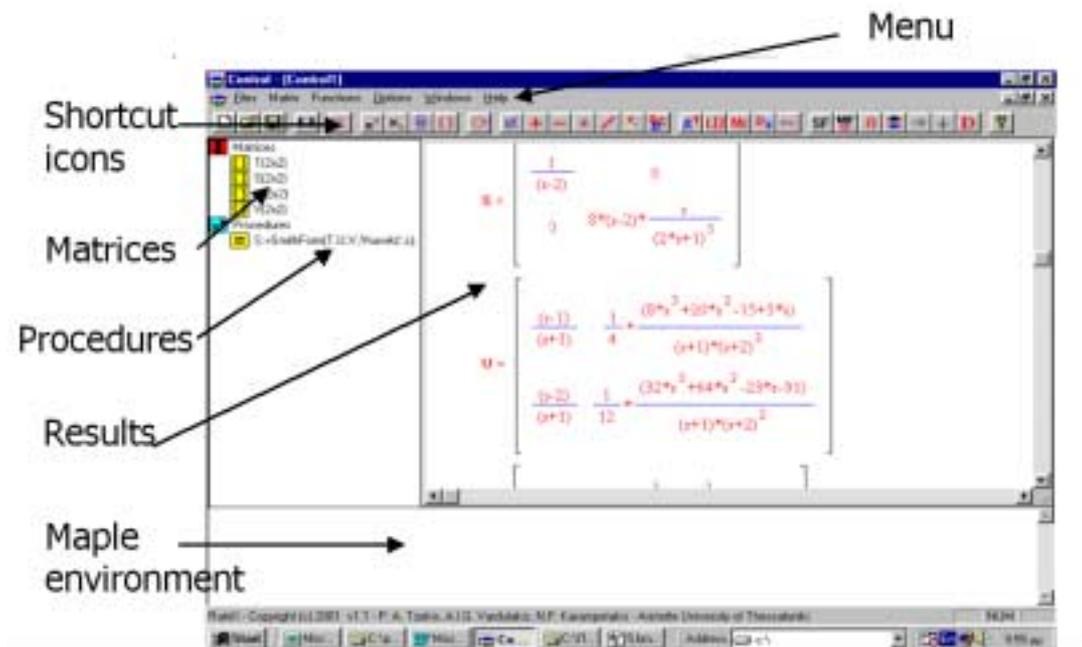


Figure 1. Description of the main window.

Some of the main procedures that the program supports are:

Simple Matrix operations.

Simple matrix operations such as addition, subtraction, multiplication and simplification of rational matrices.

Advanced rational matrix procedures.

Greatest common divisors (left, right), generalized inverse, left-right inverse, Laurent expansion at infinity, McMillan Degree, decoupling zeros of the Rosenbrock system

matrix, limit at infinity, determinant, eigenvalues, and characteristic polynomial.

Additional procedures

Smith McMillan forms, matrix fraction descriptions, solution of rational

matrix diophantine equations over different rings and their applications to control i.e. computation and parameterization of stabilizing compensators for the solution of the “tracking problem”, “model matching problem” and “decoupling problem”.

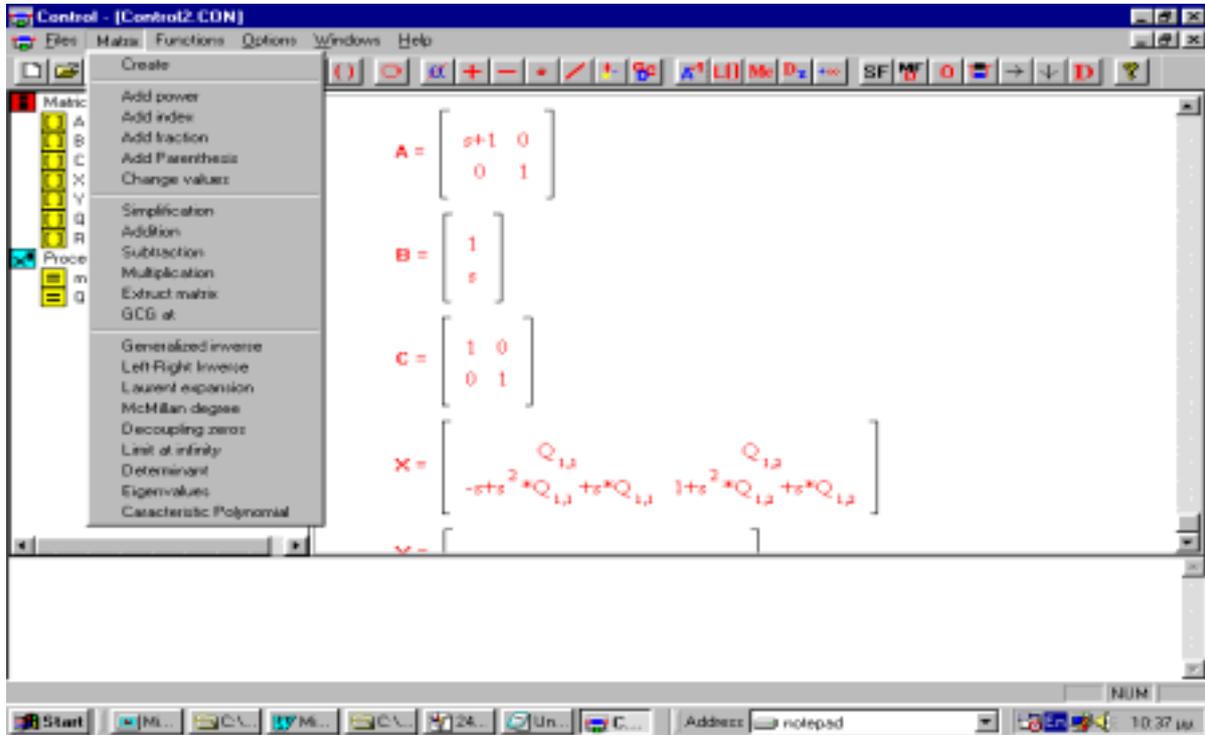


Figure 2. You select from the menu bar the procedure that you want.

The shortcut menu is used in order to help you to : a) create your own matrices, and b) execute the main procedures of the program.

You can create a matrix, either by clicking on the icon [] or by selecting *Create* from the drop down menu of the menu option *Matrix*. While you have created a matrix, this matrix is included on the left window under the title “Matrices”. You can see this matrix just by clicking on the name of the matrix or you can delete this matrix by clicking on the name of the matrix and then pressing the button Delete. After the creation of the matrices you can use the procedures from the shortcut icons or the menu bar. In any case the program asks you for the name of the matrices that you will use and the options that

needs in order to solve your problem. A simple example is given in the next section.

Some of the benefits of the above program are:

- a) the user friendly Windows based interface (e.g. use of icons, drop down menus etc. for the use of the various procedures and algorithms),
- b) internal use of the powerful kernel of the Maple symbolic computer algebra package,
- c) all procedures and calculations are build to deal with matrices having elements from the four well known rings of polynomials, proper rational functions, proper and Shur stable rational functions

and proper and Hurwitz stable rational functions, therefore allowing the user firstly to deal with the algebraic analysis of both continuous and discrete time systems and secondly design of stabilizing controllers in both the Laplace and Z-domains,

- d) the user may use for its calculations both the windows application environment and the standard Maple environment, thus the program satisfies both the beginner and advanced user.

3. Example of use

Example 1. Suppose that we are interested to calculate the Smith form at $s = \infty$ of the polynomial matrix

$$A(s) = \begin{bmatrix} 1 & s & 0 \\ 0 & 1 & s \\ 0 & 0 & 1 \end{bmatrix}$$

First we create the matrix $A(s)$ by first clicking on the shortcut icon [], secondly giving the name and the dimension of $A(s)$ and finally by completing the elements of $A(s)$.

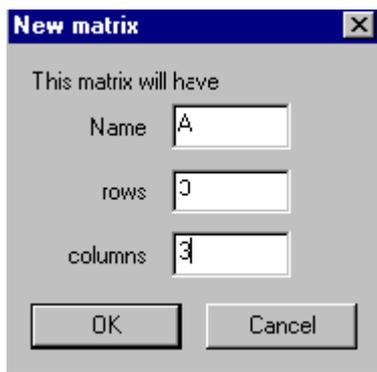


Figure 3. Give the properties of $A(s)$.



Figure 4. Entering the elements of $A(s)$.

In the sequel we select either the shortcut icon SF or select *Smith Form* from the drop down menu of the option *Functions* in the menu bar.

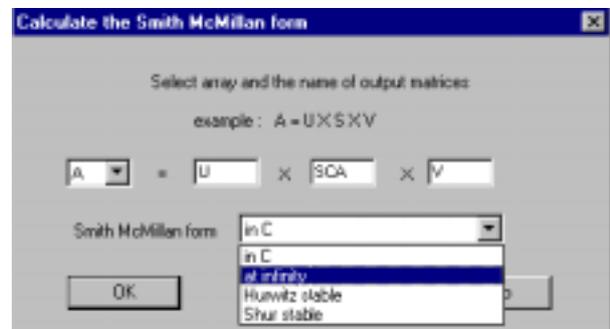


Figure 5. Complete the entries.

We complete the entries in the window that appears i.e. A is our matrix, U and V are the left and right transforming matrices that will be calculated by the program as well as SCA which is the Smith Form matrix that we are looking for. We select also from the drop down menu the word “at infinity” as we are looking for the Smith Form of $A(s)$ at $s = \infty$. Then the results are displayed on the right window.

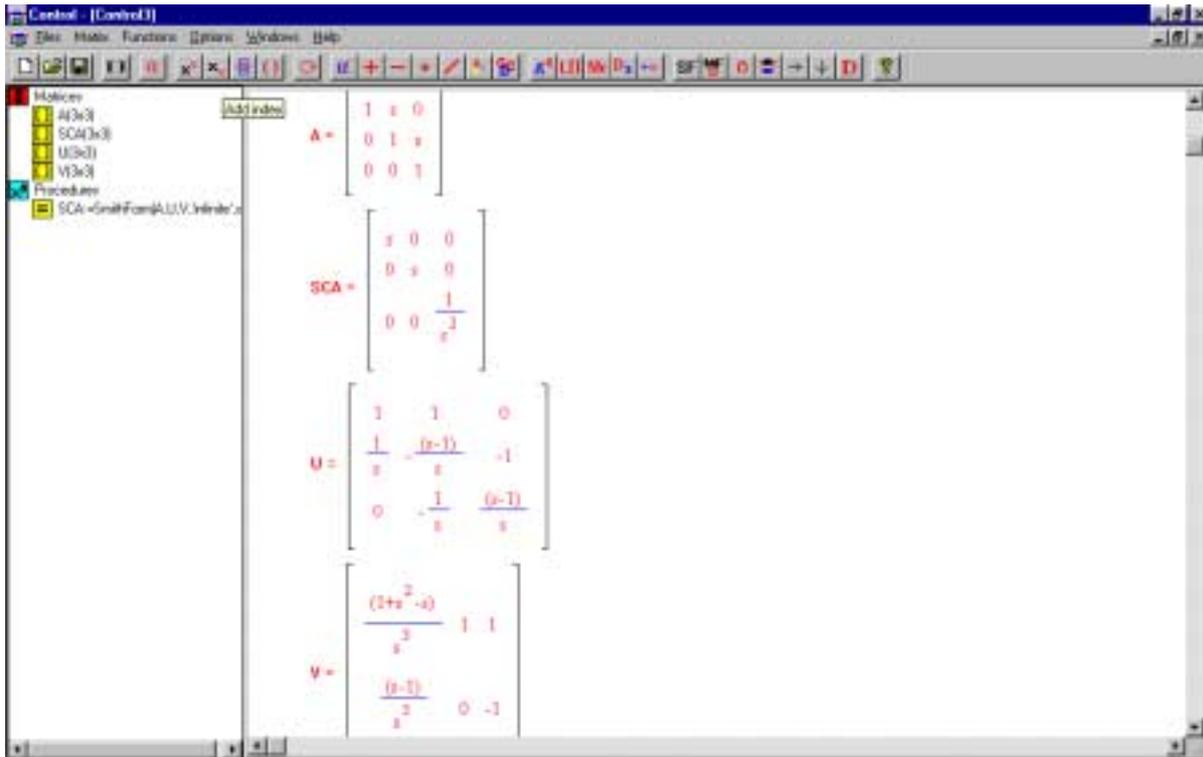


Figure 6. The results are displayed on the right window.

Example 2. Consider the polynomial matrices

$$A(s) = \begin{bmatrix} 1 & s \\ 0 & 1 \end{bmatrix}; B(s) = \begin{bmatrix} s+1 & 1 \\ -1 & s+1 \end{bmatrix}$$

and suppose that we are interested to find out the polynomial solutions of the matrix diophantine equation :

$$A(s)X(s) + B(s)Y(s) = I_2$$

Firstly we create the matrices A(s), B(s) and $C(s) = I_2$ and in the sequel either we click on the shortcut icon  or we select *Equations* from the menu bar and then *Diophantine Equations* from the drop down menu. In the sequel we complete the entries of the boxes described in Figure 7.

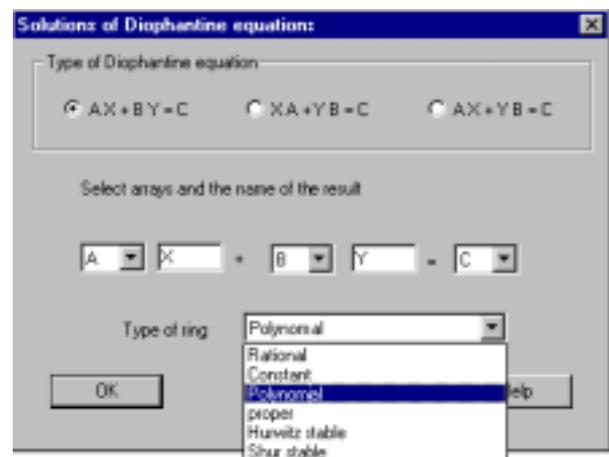


Figure 7. Complete the entries.

The results are displayed on the right window as can be easily seen in Figure 8. The entries $Q_{ij}(s)$ are arbitrary polynomials.

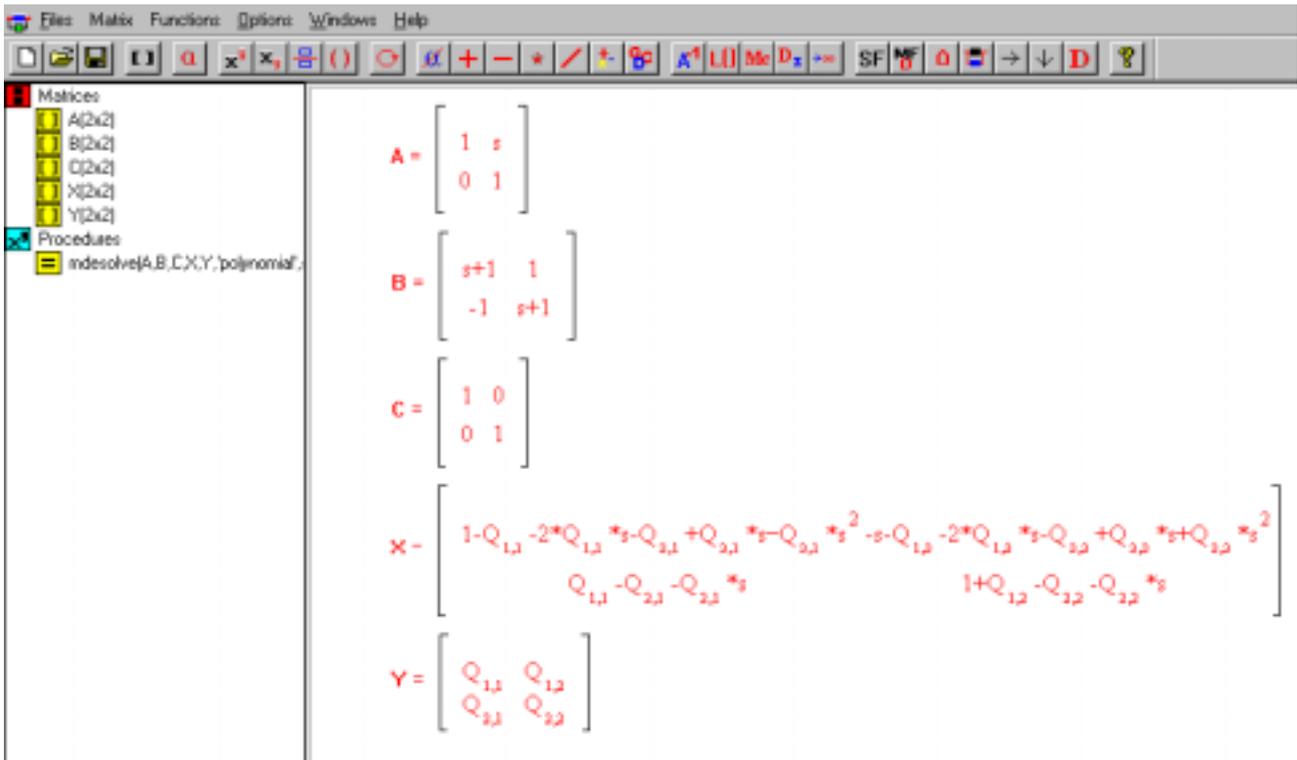


Figure 8. The solutions $X(s), Y(s)$ are displayed on the right window.

We can select specific values for the entries $Q_{ij}(s)$ in order to obtain a partial solution of our problem. In order to make these changes, we select first the matrix in which we want to change the values of $Q_{ij}(s)$ i.e. $X(s)$, and then we select *Matrix* from the menu bar and *Change values* from the drop down menu. Then we complete the following box with the values of $Q_{ij}(s)$ we want.

$$A_X = \begin{bmatrix} -2*s & -1+s^2 \\ 1 & -s \end{bmatrix}$$

$$A_Y = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

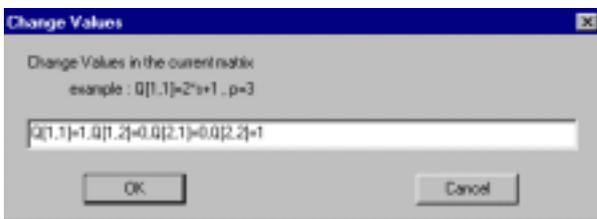


Figure 9. Changing the values of $Q_{ij}(s)$.

Finally we receive a partial solution of our problem.

Figure 10. A partial solution of our problem.

4. Conclusions

The paper has presented a user-friendly Windows based application program for the manipulation of rational matrix and the solution of basic Analysis and Synthesis problem of linear systems. This program can be used for both educational, research and industrial uses. Further information concerning this program can be found in the following URL <http://dirac.math.auth.gr/ratrix>

References

- [1] Char B.W., Geddes K.G. Gonnet G.H. and Watt S.M., 1991, *Maple V Language Reference Manual*, Springer Verlag.
- [2] J. Jones, P. Tzekis and N. P. Karampetakis, 1997, The use of MAPLE in linear systems analysis and synthesis., *Proceedings of the 4rth European Control Conference*, 1-4 July, 1997, Brussels, Belgium.
- [3] Jones J., Karampetakis N. P. and Pugh A.C., 1998, The computation and application of the generalized inverse via MAPLE., *Journal of Symbolic Computation*, **25**, No.1, pp.99-124.
- [4] Karampetakis N.P. and Tzekis P., 1999, Symbolic manipulation of rational matrices and applications, *CACSD'99*.
- [5] Kucera V., 1979, *Discrete Linear Control, The Polynomial Equation Approach*, John Wiley & Sons, Chichester
- [6] Kucera V., 1991, *Analysis and Design of Discrete Linear Control Systems*, Prentice Hall.
- [7] PolyX, The polynomial toolbox of MATLAB, <http://www.polyx.cz>
- [8] Vardulakis A.I.G., 1991, *Linear Multivariable Control : Algebraic Analysis and Synthesis Methods*, John Willey & Sons.
- [9] Vidyasagar M., 1985, *Control System Synthesis, A Factorization Approach*, MIT Press, Cambridge.