THE USE OF MAPLE IN LINEAR SYSTEMS ANALYSIS AND SYNTHESIS*

N. P. Karampetakis†, P. Tzekis† and J. Jones‡

† *Department of Mathematics, Aristotle University of Thessaloniki, Thessaloniki 54006, Greece, Fax : ++31 997951, email : karampetakis@ccf.auth.gr*

‡ *Department of Mathematical Sciences, Loughborough University of Technology, Loughborough LE11 3TU, England, U.K. Fax : ++1509 211867, email : J.Jones1@lut.ac.uk*

Abstract

The main purpose of this work is to present four new packages in the computer algebra system MAPLE for analysis and synthesis of linear, time invariant, multivariable systems. More specifically we examine a) the structure of rational matrices, b) the solution of ARMA-Representations, c) generalized state space representations of Polynomial Matrix Descriptions (PMDs) and d) solutions of diophantine equations with applications to linear systems synthesis problems e.g. model matching problem, disturbance rejection problem e.t.c.

Keywords : symbolic packages, linear systems, analysis, synthesis.

1 Introduction

The symbolic computational packages, such as Maple [1], Mathematica, MACSYMA, Reduce, MuMath and Scratchpad, are computer environments for doing Mathematics. They are particularly applicable to "computationally attractive" problems because they work with symbolic, numerical and graphical computations. 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 $0.33333...$) resulting in no loss of accuracy during a calculation (i.e. $(1/3) * 3 = 1$ while $(1/3.0) * 3 = 0.999999$). 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.

The three main reasons which had lead us to implement our work in the symbolic computational package MAPLE are a) the existence of the linear algebra package *linalg* which consists of numerous procedures used in the solution of problems in linear algebra (e.g. i) inverse - computes the inverse of a non-singular matrix, ii) smith - computes the smith form at C of a polynomial matrix, iii) linsolve - solves a set of linear equations), b) the ability to perform 'polynomial operations' and c) the existence of facilities for the creation of additional packages.

Perhaps understandably, however, neither one of these symbolic computer packages contain existing procedures or indeed packages for specialized work in linear system analysis and synthesis. However they are clearly flexible enough for the implementation of such work to be carried out. This is the motivation of this paper and we present therefore four packages of procedures, developed for use in Maple, for the analysis and synthesis of linear, time invariant, multivariable systems. More specifically in Section 2 we present a package of procedures concerning the structure of rational matrices while in Section 3 we present a package for solving discrete time ARMA-representations [4] and continuous time nonregular AR-Representations [2]. In section 4 we use MAPLE for the construction of generalized state space representations for PMDs using known algorithms presented in [3] while in Section 5 we give a MAPLE package for solving matrix diophantine

*The work is supported by the Greek General Secreteriat of Industry, Research and Technology.

equations over different rings [9] with some applications in synthesis problems [10], [11]. Finally the way to implement these packages is presented in Section 6. The whole work is illustrated via an example in Section 7.

2 Matrix Structures Package

The Maple name for this package is *linstruct*. It contains procedures concerned with determining the structure of rational matrices. It also include several matrix operation procedures which can be seen as extensions to the existing linear algebra package *linalg*. It requires the Maple procedure **choose** and comprises of 25 procedures. Several of these procedures are discussed below.

smith(A, U, V, WhatIs) - Smith-McMillan form of a rational matrix

Parameters : $A(s)$ is a known rational matrix. a) If WhatIs='finite' then $U(s)$ and $V(s)$ are unimodular matrices computed from the mentioned procedure, such that $U(s)S^{C}_{A(s)}(s)V(s) = A(s)$. The output of this procedure is the Smith-McMillan form of $A(s)$ at C i.e. $S^{C}_{A(s)}(s)$. It returns also the unimodular matrices $U(s)$ and $V(s)$ which satisfy the following $U(s)S^{C}_{A(s)}(s)V(s) = A(s)$. b) If WhatIs='infinite' then $U(s)$ and $V(s)$ are biproper matrices computed from the mentioned procedure, such that $U(s)S^{\infty}_{A(s)}(s)V(s) = A(s)$. The output of this procedure is the Smith-McMillan form of A(s) at $s = \infty$ i.e. $S^{\infty}_{A(s)}(s)$. It returns also the biproper matrices $U(s)$ and $V(s)$ which satisfy the following $U(s)S^{\infty}_{A(s)}(s)V(s) = A(s)$. c) If WhatIs='Hurwitz' then $U(s)$ and $V(s)$ are proper and Hurwitz stable matrices computed from the mentioned procedure, such that $U(s)S^{C^{+}\cup\{\infty\}}_{A(s)}(s)V(s) = A(s)$. The output of this procedure is the Smith-McMillan form of $A(s)$ at $\Omega \equiv C^{+} \cup \{\infty\}$ i.e. $S^{\Omega}_{A(s)}(s)$. It returns also the proper and Hurwitz stable matrices $U(s)$ and $V(s)$ which satisfy the following $U(s)S^{\Omega}_{A(s)}(s)V(s) = A(s)$. d) If WhatIs='Shur' then $U(s)$ and $V(s)$ are proper and Shur stable matrices computed from the mentioned procedure, such that $U(s)S^{\Omega}_{A(s)}(s)V(s) = A(s)$ where $\Omega := \{s \mid |s| \geq 1\}$. The output of this procedure is the Smith-McMillan form of $A(s)$ at Ω i.e. $S^{\Omega}_{A(s)}(s)$. It returns also the proper and Shur stable matrices $U(s)$ and $V(s)$ which satisfy the following $U(s)S^{\Omega}_{A(s)}(s)V(s) = A(s)$. A theoretical algorithm proposed in [10] was used for the construction of the mentioned procedure.

ginverse(A) - generalized inverse of the rational matrix A(s)

Parameters : $A(s)$ is a rational matrix. The output of this procedure is the generalized inverse of the rational matrix $A(s)$ [2,5].

The remaining procedures concerns structural properties of rational matrices such as the computation of the forward (backward) fundamental sequence of the inverse of a polynomial matrix, the computation of unimodular matrices which make a rational matrix column or row reduced e.t.c.

3 System Solutions Package

The Maple name for this package is *linsol*. It contains procedures for determining a) forward, backward and symmetric solutions of discrete time ARMA-Representations, b) admissibility conditions for discrete time ARMA-Representations and c) solutions of nonregular continuous time AR-Representations. It uses several procedures from the package *linstruct* and these are therefore included. The package currently consists of 23 procedures. The procedures contained within the package *linsol* can be subdivided into the following areas and several procedures are discussed:

Solutions of discrete time ARMA - Representations - Procedures for finding the forward, backward and symmetric solutions and the corresponding admissibility conditions [4] of the discrete time ARMA-Representation

$$A(\sigma)\, y(k) = B(\sigma)\, u(k)$$

where σ denotes the forward shift operator i.e. $\sigma^{i} y(k) = y(k+i)$, $A(\sigma) \in R[\sigma]^{r\times r}$ with $\det[A(\sigma)] \neq 0$ and $B(\sigma) \in R[\sigma]^{r\times m}$.

solution(A,B,q,WhatIs) - Solutions of discrete time ARMA-Representations.

Parameters : $A(\sigma) \in R[\sigma]^{r\times r}$, $B(\sigma) \in R[\sigma]^{r\times m}$ are known polynomial matrices with greatest degree among its elements q. The output of this procedure is a) if WhatIs='forward' then the solution of the ARMA-Representation $A(\sigma)\, y(k) = B(\sigma)\, u(k)$ in a *forward* fashion given that a set of admissible initial output conditions exist for a given input sequence b) If WhatIs='backward' then the solution of the ARMA-Representation $A(\sigma)\, y(k) = B(\sigma)\, u(k)$ in a *backward* fashion given that a set of admissible final output conditions exist for a given input sequence while c) if WhatIs='symmetric' the solution of the ARMA-Representation $A(\sigma)\, y(k) = B(\sigma)\, u(k)$ within the interval $[0, N]$ for a combined set of admissible initial and final output conditions.

admissible(A,B,q,'L','WhatIs') - Admissible conditions of discrete time ARMA-Representations.

Parameters : $A(\sigma) \in R[\sigma]^{r\times r}$, $B(\sigma) \in R[\sigma]^{r\times m}$ are known polynomial matrices with greatest degree among its elements q. The output of this procedure is a) if WhatIs='forward' then the set of admissibility equations for the forward solution of $A(\sigma)\, y(k) = B(\sigma)\, u(k)$ in terms of the initial output conditions $\{y(0), y(1), ..., y(q-1)\}$ and input sequence b) if WhatIs='backward' then the set of admissibility equations for the backward solution of $A(\sigma)\,y(k) = B(\sigma)\,u(k)$ in terms of the final output conditions $\{y(N), y(N-1), ..., y(N-q+1)\}$ and input sequence and c) if WhatIs='symmetric' then the set of admissibility equations for the symmetric solution in terms of the output conditions $\{y(N), y(N-1), ..., y(N-q+1), y(q-1), ..., y(1), y(0)\}$ and input sequence $\{u(0), u(1), ..., u(N)\}$. Correspondingly solves for a set L of admissible initial output conditions if a solution to these equations exists.

Solutions of nonregular AR-Representations - Procedures for finding the solution of the continuous time AR-Representation [2]

$$A(\rho)\,y(t) = 0$$

where $\rho := \frac{d}{dt}$, $A(\rho) \in R[\rho]^{r\times r}$ with $\det[A(\rho)] = 0$ or $A(\rho) \in R[\rho]^{r\times m}$ with $r \neq m$.

arsolve(A,B,q) - Solution of continuous time AR-Representations.

Parameters : $A(\rho) \in R[\rho]^{r\times r}$ with $\det[A(\rho)] = 0$ or $A(\rho) \in R[\rho]^{r\times m}$ with $r \neq m$ is a known polynomial matrix with greatest degree among its elements q. B is the set of initial conditions $B := \left[\begin{array}{cccc} y(0-)^T & y^{(1)}(0-)^T & \cdots & y^{(q-1)}(0-)^T \end{array}\right]$. The output of this procedure is the solution of the AR-Representation $A(\rho)\,y(t) = 0$ or a message that the system has no solution if the initial conditions given by the matrix B are not compatible.

Several other procedures concerning the forward, backward and symmetric solution for a specific discrete time domain interval and the plot of such solutions are also included in this package.

4 System Representations Package

The Maple name for this package is *linrep*. It contains procedures for determining alternative and equivalent representations of a system. It uses several procedures from the previous package *linstruct* and these are therefore included. The package currently consists of 25 procedures. The procedures contained within the package *linrep* can be subdivided into the following areas and several procedures are discussed:

Right & Left MFDs Procedures - Procedures for representing a matrix $G(s) \in R(s)^{p\times m}$ as a right (resp. left) matrix fraction description (MFD)

$$G(s) = N_1(s)D_1(s)^{-1} (= D_2(s)^{-1}N_2(s))$$

where $N_1(s) \in R_\Omega[s]^{p\times m}$, $D_1(s) \in R_\Omega[s]^{m\times m}$ (resp. $N_2(s) \in R_\Omega[s]^{p\times m}$, $D_2(s) \in R_\Omega[s]^{p\times p}$) i.e. $R_\Omega[s]^{p\times m}$ the ring of rational matrices with no poles in the region Ω.

lmfd(G,WhatIs) -MFD of a rational matrix

Parameters : $G(s)$ is a rational matrix. The output of this procedure is a) if WhatIs='finite' then a left matrix fraction description of a matrix $G(s) \in R(s)^{p\times m}$ where the corresponding matrices $N_2(s) \in R[s]^{p\times m}$, $D_2(s) \in R[s]^{p\times p}$ are left coprime i.e.

$$S^C_{\left(\begin{array}{cc} D_2(s) & N_2(s) \end{array}\right)}(s) = \left(\begin{array}{cc} I_p & O_{p,m} \end{array}\right)$$

where $S^C_{(\cdot)}$ denotes the Smith-McMillan form of the indicated matrix. Similarly the procedure *rmfd(G,'finite')* computes a right coprime MFD, b) if WhatIs='Hurwitz' ('Shur') then a left matrix fraction description of a matrix $G(s) = D_2(s)^{-1}N_2(s) \in R(s)^{p\times m}$ where the corresponding matrices $N_2(s) \in R_\Omega[s]^{p\times m}$, $D_2(s) \in R_\Omega[s]^{p\times p}$ are $\Omega-$coprime where $\Omega \equiv C^+ \cup \{\infty\}$ ($\Omega \equiv \{s \mid |s| \geq 1\}$) i.e.

$$S^\Omega_{\left(\begin{array}{cc} D_2(s) & N_2(s) \end{array}\right)}(s) = \left(\begin{array}{cc} I_p & O_{p,m} \end{array}\right)$$

where $S^\Omega_{(\cdot)}$ denotes the Smith-McMillan form in the region Ω of the indicated matrix. Similarly the procedure *rmfd(G,'Hurwitz')* (*rmfd(G,'Shur')*) computes a right $C^+ \cup \{\infty\}$ $-$MFD ($\Omega-$MFD where $\Omega \equiv \{s \mid |s| \geq 1\}$).

Realization Procedures - Procedures for realizing polynomial and stricly proper matrices $G_{pol}(s) \in R[s]^{p\times m}$ and $G_{spr}(s) \in R_{pr}(s)^{p\times m}$.

realization(G,WhatIs) - minimal realization of a rational matrix

Parameters : $G(s) \in R(s)^{p\times m}$ is a rational matrix. a) If WhatIs=3 then the output of this procedure is the triple pair
$\left(\left(\begin{array}{cc} C & C_\infty \end{array}\right) \in R^{p\times n}, \left(\begin{array}{cc} J & 0 \\ 0 & J_\infty \end{array}\right) \in R^{n\times n}, \left(\begin{array}{c} B \\ B_\infty \end{array}\right) \in R^{n\times m}\right.$
where J, J_∞ are in block Jordan form such that

$$G(s) = \left(\begin{array}{cc} C & C_\infty \end{array}\right)\left(\begin{array}{cc} sI-J & 0 \\ 0 & I-sJ_\infty \end{array}\right)^{-1}\left(\begin{array}{c} B \\ B_\infty \end{array}\right)$$

This is based on an algorithm as seen in [3] and $\left(\left(\begin{array}{cc} C & C_\infty \end{array}\right), \left(\begin{array}{cc} J & 0 \\ 0 & J_\infty \end{array}\right), \left(\begin{array}{c} B \\ B_\infty \end{array}\right)\right)$ is a minimal realization of $G(s)$ i.e the dimension n is minimal. b) If WhatIs=4 then the output of this procedure is the quadruple pair $\left(\left(\begin{array}{cc} C & C_\infty \end{array}\right) \in R^{p\times \nu}, \left(\begin{array}{cc} J & 0 \\ 0 & \hat{J}_\infty \end{array}\right) \in R^{\nu\times \nu}, \left(\begin{array}{c} B \\ B_\infty \end{array}\right) \in R^{\nu\times m}, D_\infty \in R^{p\times m}\right)$ where $J, \hat{J}_\infty$ are in block Jordan form such that

$$G(s) = \left(\begin{array}{cc} C & C_\infty \end{array}\right)\left(\begin{array}{cc} sI-J & 0 \\ 0 & I-s\hat{J}_\infty \end{array}\right)^{-1}\left(\begin{array}{c} B \\ B_\infty \end{array}\right) + D_\infty$$

where $\nu \leq n$.

Reduction Procedures - Procedures for reducing a polynomial matrix description (PMD) to an equivalent generalized state space (GSS) form i.e. Given given a PMD

$$T(\rho)\beta(t) = U(\rho)u(t)$$
$$y(t) = V(\rho)\beta(t) + W(\rho)u(t)$$

where $\rho := \frac{d}{dt}$, $T(\rho) \in R[\rho]^{r\times r}$ $(\det[T(\rho)] \neq 0), U(\rho) \in R[\rho]^{r\times m}, V(\rho) \in R[\rho]^{p\times r}, W(\rho) \in R[\rho]^{p\times m}$ compute a positive integer σ and matrices $E, A \in R^{\sigma\times\sigma}$, $(\det[sE - A] \neq 0), B \in R^{\sigma\times m}, C \in R^{p\times\sigma}, D \in R^{p\times m}$ such that the (GSS) model described by

$$E\frac{dx(t)}{dt} = Ax(t) + Bu(t)$$
$$y(t) = Cx(t) + Du(t)$$

possesses the same finite and infinite behavior as the corresponding PMD model.

reduction(P,r,p,m,WhatIs) - Equivalent (GSS) model
Parameters : $P(s) \in R(s)^{(r+p)\times(r+m)}$ is the Rosenbrock system matrix of the PMD model. The output of this procedure is a) if WhatIs='Bosgra' then the Rosenbrock system matrix of the Bosgra & Van Der Weiden equivalent (GSS) model [3] while b) if WhatIs='Vardulakis' then the Rosenbrock system matrix of the Vardulakis equivalent (GSS) model [3]. The second model, in general, is superior too that of Bosgra & Van Der Weiden equivalent model in that it has lower dimension σ.

5 Matrix Diophantine Equation Package

The Maple name for this package is *lindioph*. It contains procedures for determining a) solutions of matrix diophantine equations and b) solutions of synthesis problems in Linear Systems. It uses several procedures from the package *linstruct* and these are therefore included. The package currently consists of 33 procedures. The procedures contained within the package *lindioph* can be subdivided into the following areas and several procedures are discussed:

Solutions of Matrix Diophantine Equations - Procedures for finding solutions of matrix diophantine equations i.e.

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

where $A(s) \in \mathbf{R}(s)^{p\times m}$, $B(s) \in \mathbf{R}(s)^{p\times n}$ and $C(s) \in \mathbf{R}(s)^{p\times l}$ [7, 9].

mdesolve(A,B,C,X,Y,WhatIs) - solution of matrix diophantine equations
Parameters : A, B, C are known rational matrices which are involved in the above diophantine equation. a) If WhatIs='general' then the output of this procedure is true if the above diophantine equation has a rational solution and false if not. The family of all the rational solutions X and Y is also computed. b) If WhatIs='constant' then the output of this procedure is true if the above diophantine equation has a constant solution and false if not. The family of all the constant solutions X and Y is also computed. c) If WhatIs='polynomial' then the output of this procedure is true if the above diophantine equation has a polynomial solution and false if not. The family of all the polynomial solutions X and Y is also computed. d) If WhatIs='proper' then the output of this procedure is true if the above diophantine equation has a proper solution and false if not. The family of all the proper solutions X and Y is also computed. e) If WhatIs='Hurwitz' then the output of this procedure is true if the above diophantine equation has a proper and Hurwitz stable solution and false if not. The family of all the proper and Hurwitz stable solutions X and Y is also computed. f) If WhatIs='Shur' then the output of this procedure is true if the above diophantine equation has a proper and Shur stable solution and false if not. The family of all the proper and Shur stable solutions X and Y is also computed. Similarly the procedure *bmdesolve(A,B,C,X,Y,WhatIs)* solves the bilateral diophantine equation $A(s)X(s) + Y(s)B(s) = C(s)$.

Synthesis Problems of Linear Systems - Procedures for finding the solution of specific synthesis problems such as model matching problem, disturbance rejection problem e.t.c. [8]. Consider the following closed loop system

Figure 1. Closed loop system.

stabcomp(P,WhatIs,s) - family of all stabilizing compensators of P under unity output feedback.
Parameters : P is a known proper rational (transfer function) matrix of s. If WhatIs=0 (WhatIs=1) then the output of this procedure are all the $C^+ \cup \{\infty\}$-stabilizing compensators C of P (Ω-stabilizing compensators C of P where $\Omega := \{s : |s| \geq 1\}$) [10].

matching(P,H,WhatIs,s) - model matching by dynamic compensation and unity output feedback of P.
Parameters : P and H are known proper rational (transfer function) matrix of s. If WhatIs=0 (WhatIs=1) then the output of this procedure are all the $C^+ \cup \{\infty\}$-stabilizing compensators ($\Omega-$stabilizing compensators where $\Omega := \{s : |s| \geq 1\}$) which gives rise to the closed loop transfer function matrix H under unity output feedback (see Figure 1) [10].

decoupling(P,H,WhatIs,s) - decoupling by dynamic compensation and unity output feedback of P.

Parameters : P is a known proper rational (transfer function) matrix of s. If WhatIs=0 (WhatIs=1) then H is nonsingular, diagonal and has arbitrary desired poles outside $C^+ \cup \{\infty\}$ ($\Omega := \{s : |s| \geq 1\}$) and the output of this procedure is a stabilizing compensator C for P such that the closed loop system under unity output feedback has the transfer function matrix H. (see Figure 1) [10].

simstab($\mathbf{P_0},\mathbf{P_1}$,*WhatIs,s) - simultaneous stabilization of two plants with a common compensator.*

Parameters :P_0 and P_1 are known proper rational (transfer function) matrices of s. If WhatIs=0 (WhatIs=1) then the output of this procedure are all the common $C^+ \cup \{\infty\}$-stabilizing compensators C of P_0 and P_1 (common Ω-stabilizing compensators of P_0 and P_1 where $\Omega := \{s : |s| \geq 1\}$) [11].

multicomp(P,$\mathbf{C}_0, \mathbf{C}_1$*, WhatIs,s) - simultaneous stabilization of a plant with two compensators.*

Parameters :P is a known proper rational (transfer function) matrix of s. If WhatIs=0 (WhatIs=1) then the output of this procedure are two $C^+ \cup \{\infty\}$-stabilizing compensators (Ω-stabilizing compensators where $\Omega := \{s : |s| \geq 1\}$) C_0 and C_1 that stabilizes P [10].

multicomps(P,$\mathbf{C}$*, WhatIs,s) - simultaneous stabilization of a plant with two same compensators.*

Parameters :P is a known proper rational (transfer function) matrix of s. If WhatIs=0 (WhatIs=1) then the output of this procedure are two $C^+ \cup \{\infty\}$-stabilizing compensators (Ω-stabilizing compensators where $\Omega := \{s : |s| \geq 1\}$) $C = C_0 = C_1$ that stabilizes P [10].

tracking(P,$\mathbf{y}_{1\Omega}$*, WhatIs,s) - asymptotic tracking by unity output feedback.*

Parameters :P is a known proper rational (transfer function) matrix of s and the Laplace transform of the reference input $u_1(t)$ is $\hat{u}_1(s) = \beta_1(s)/y_{1\Omega}(s)$ where $y_{1\Omega}(s) \in R[s]$ is monic and fixed having zeros only in $C^+ \cup \{\infty\}$ ($\Omega := \{s : |s| \geq 1\}$). If WhatIs=0 (WhatIs=1) then the output of this procedure are all the $C^+ \cup \{\infty\}$-stabilizing compensators C (Ω-stabilizing compensators C where $\Omega := \{s : |s| \geq 1\}$) that asymptotically tracks $u_1(t)$ if $u_2(t) \equiv 0$, $u_3(t) \equiv 0$ (see Figure 1) [10].

rejection(P,$y_{3\Omega}$*, WhatIs,s)-disturbance rejection by unity output feedback.*

Parameters :P is a known proper rational (transfer function) matrix of s and the Laplace transform of the disturbance input $u_3(t)$ is $\hat{u}_3(s) = \beta_3(s)/y_{3\Omega}(s)$ where $y_{3\Omega}(s) \in R[s]$ is monic and fixed having zeros only in $\Omega := C^+ \cup \{\infty\}$ if WhatIs=0 ($\Omega := \{s : |s| \geq 1\}$ if WhatIs=1). If WhatIs=0 (WhatIs=1) then the output of this procedure are all the $C^+ \cup \{\infty\}$-stabilizing compensators (Ω-stabilizing compensators where $\Omega := \{s : |s| \geq 1\}$) that asymptotically rejects $u_3(t)$ if $u_1(t) \equiv 0$, $u_2(t) \equiv 0$ (see Figure 1) [10].

6 Implementation via MAPLE

The above procedures are easily implemented through the following three steps :
(i) Read in the linear algebra package contained within MAPLE via >with(linalg):
(ii) Load required package into MAPLE via >with(package):
(iii) Implement the procedure via >Procedure-Name(arguments).

7 Illustrated Example

Consider the rational matrix

$$A(s) = \begin{pmatrix} s^3 & 0 & \frac{1}{s^2-2} & 0 \\ 0 & 2s & 0 & \frac{1}{s(s-1)} \\ 0 & 0 & \frac{s-1}{s^3} & 0 \\ 0 & s-1 & 0 & \frac{1}{s^3} \end{pmatrix}$$

Find a) the Smith McMillan form of $A(s)$ at $s = \infty$ i.e. $S^{\infty}_{A(s)}(s)$ and b) the biproper matrices $U(s), V(s)$ s.t. $U(s)S^{\infty}_{A(s)}(s)V(s) = A(s)$. We can find the above result in MAPLE by calling the procedure *smith* from the package *linstruct.*

$> with(lina\lg):$
Warning: new definition for norm
Warning: new definition for trace
$> with(linstruct):$
$> A := matrix(4,4,[s\hat{}3,0,1/(s\hat{}2-2),0,0,2*s,0,1/(s*(s-1)),0,0,(s-1)/s\hat{}3,0,0,s-1,0,1/s\hat{}3]);$

$$A := \begin{pmatrix} s^3 & 0 & \frac{1}{s^2-2} & 0 \\ 0 & 2s & 0 & \frac{1}{s(s-1)} \\ 0 & 0 & \frac{s-1}{s^3} & 0 \\ 0 & s-1 & 0 & \frac{1}{s^3} \end{pmatrix}$$

$> smith(A,U,V,'\inf inite');$

$$\begin{pmatrix} s^3 & 0 & 0 & 0 \\ 0 & s & 0 & 0 \\ 0 & 0 & \frac{1}{s^2} & 0 \\ 0 & 0 & 0 & \frac{1}{s^2} \end{pmatrix}$$

$> op(U);$

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & \frac{s-1}{s} & 0 \\ 0 & \frac{s-1}{s} & 0 & -\frac{1}{2}\frac{s-2}{s} \end{pmatrix}$$

$> op(V);$

$$\begin{pmatrix} 1 & 0 & \frac{1}{(s^2-2)s^3} & 0 \\ 0 & 1 & 0 & \frac{1}{2}\frac{1}{s^2(s-1)} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

8 Conclusions.

In this paper we have developed and presented a package for use in Linear Systems Analysis and Synthesis which is divided into four (4) sub-packages. Each package contains numerous procedures and can be read in independently of one another or as a whole. The package is easily implemented as it is self contained and is easily expendable by adding new procedures or packages. The MAPLE codes can be supplied free by J. Jones in the email address J.Jones1@lut.ac.uk or by P.Tzekis in the email address Tzekis@olymp.ccf.auth.gr under request. One example arising from control theory has also been considered to show the implementation of one of these procedures in MAPLE. Structural properties such as controllability, observability e.t.c. as well as structural invariants of Polynomial Matrix Descriptions (PMDs) will be discussed in further work, using the proposed procedures.

9 Bibliography

[1] Char B.W., Geddes K.G. Gonnet G.H. and Watt S.M., 1991, *Maple V Language Reference Manual*, Springer Verlag.
[2] Jones J., Karampetakis N. P. and Pugh A.C., 1996, Computation of the generalised inverse of a rational matrix via MAPLE and applications., Proceedings of the IEEE Symposium on Computer-Aided Control System Design, Deadborn, Michigan, USA, September 15-18, 1996.
[3] Karampetakis N. P. and Vardulakis A.I., 1995, On the reduction of a polynomial matrix model of a linear multivariable system to generalised state space form., *Proceedings of the EURACO Workshop on "Recent Results in Robust and Adaptive Control"*, 11-14th September, 1995, Florence, Italy, pp.65-92.
[4] Karampetakis N. P., Jones J. and Antoniou S., 1996, Forward, backward and symmetric solutions of discrete ARMA-representations, submitted for possible publication in the journal of *Circuit Systems & Signal Processing.*
[5] Karampetakis N., 1997, Computation of the generalized inverse of a polynomial matrix and applications, *Linear Algebra and its Applications*, **252**, pp.35-60.
[6] Kucera V., 1979, *Discrete Linear Control, The Polynomial Equation Approach.*, John Wiley & Sons, Chichester.
[7] Kucera V., 1991, *Analysis and Design of Discrete Linear Control Systems*, Prentice Hall.
[8] Kucera V., 1993, Diophantine equations in Control - A Survey., *Automatica*, Vol. **29**, No.6, pp.1361-1375.
[9] Tzekis P., Karampetakis N.P. and Vardulakis A.I., 1996, Solutions of Diophantine equations over rings via MAPLE, *Proceedings of the 4rth IEEE Mediterranean Symposium on New Directions in Control and Automation*, June 10-14, 1996, Chania, Crete.
[10] Vardulakis A.I.G., 1991, *Linear Multivariable Control, Algebraic Analysis and Synthesis Methods.*, John Wiley & Sons, Chichester.
[11] Vidyasagar M., 1985, *Control System Synthesis, A Factorization Approach*, MIT Press, Cambridge.