On Application of Computer Algebra to Control Theory

O. Saito, T.Yuasa and L. Xu
Toyohashi University of Technology Toyohashi 441 Japan

1 Introduction

At present, CADCS (Computer Aided Design of Control System) systems are indispensable in applying control theory to practical problems and various systems have been released. Almost all the systems have been implemented by using numerical computational language such as Fortran and C. However recently the symbolic or the formula manipulation is pointed out to be necessary in CADCS and some systems in which symbolic manipulation can be carried out, have been developed. To begin with, the formula manipulation was studied in the field of artificial intelligence and many general formula manipulation systems, for example, MACSYMA, Mathematica, Maple, and REDUCE have been exploited. Now the formula manipulation , generally speaking, is called as Computer Algebra or Computing Algebra, because its main results are based on the algebra, specially polynomial ring. On the other hand the control theory is constructed on the linear algebra. So there exists deep relationship between the computer algebra and control theory, specially polynomial approach.

The polynomial approach to the design of linear multivariable control systems, has been widely discussed and is attractive in both the theories and applications. As the approach requires the complex manipulation of polynomial and rational function matrices, we encounter difficulties in implementing, in a computational way, the algebraic design procedure. Therefor many numerical algorithms for the computational problems have been developed and may be useful for the implementation of numerical CADCS system. But the authors consider that the formula manipulation, in other words, computing algebra is better way for the CADCS system of polynomial approach.

This paper presents two results in applying the computer algebra to control theory. First it is shown that the general formula manipulation system is very useful for CACDCS system of polynomial approach and a prototype system based on REDUCE is demonstrated. Secondly the new computing algebra system — parallel formula manipulation system — which the authors are developing, is introduced and it is clarified to be usuful for the n-D control system theory.

2 Polynomial approach and Computing Algebra

This section presents a software package based on REDUCE for the polynomial approach. The package allows us to perform successfully the polynomial design procedure.

Before mention of the package, here the polynomial approach is reviewed briefly. Consider the control system configuration in Fig.1, where v, u, u_0 and y are the command input, control input, disturbance input and controlled output vectors, respectively. The plant is modeled by the $(m \times n)$ transfer function matrix $P(s)$, each entry of which is a rational function of variable s. The plant is assumed to be free of hidden modes and strictly proper. Let the 2 degree not to be fully taken into account, however, different control problems require different particular solution choice (e.g. minimum row degrees for LQ optimization in contrast to minimum column degrees for deadbeat) and fair comparison can be made only for a specific task.

EOM is of a genuine polynomial nature and the only one to provide a general solution in one shot. In addition, no initial settings are required. Last but not least, built directly on fundamental theorems, it is of pedagogical value. On the other hand, the method is rather computationally involved and usually the slowest one within the competition. In addition, after each polynomial operation, it demands to determine the degree of the product. This critical step must be accomplished by a specific program module.

PIM is based exclusively on constant matrix operations, which are well-understood and already implemented in MATLAB. It provides one particular solution from the prespecified class of solutions having degree less than the chosen parameter.

SSM is based both on matrix and polynomial operations. It is very flexi regarded particular solutions choice. The solution can be obtained either from the prespecified family or the extra degrees of freedom may be removed by different constraints.

The last two methods, provided they are run also for homogeneous equation, may be used to get a general solution or any other particular solution using (3).

7 References

[1] Antsaklis P. J., Gao Z. (1992), Polynomial and Rational Matrix Interpolation: Theory and Control Application, *Control Systems Technical Report # 71*, University of Notre Dame, Indiana, U.S.A.

[2] Emre E., (1980), The polynomial equation $QQ_c + RP_c = \Phi$ with application to dynamic feedback, SIAM J. Contr. Optimiz., 18, 611-620.

[3] Kraffer F. (1993), State-space method to solve polynomial matrix diophantine equation, *submitted to be published*

[4] Pejchová S., (1993), Software package for polynomial operations I., *Research Report # 1765*, Institute of Information Theory and Automation, Czech Republic.

[5] Kučera V. (1979), Discrete Linear Control, *Academia*, Prague, Czech Republic.

[6] Kwakernaak H. (1990), MATLAB Macros for Polynomial $\mathcal{H}_\infty$ Control System Optimization, *Memorandum no. 881*, University of Twente, the Netherlands.

[7] Šebek M. (1990), An Algorithm for Spectral Factorizaton of Polynomial Matrices with any Signature, *Memorandum no. 912*, University of Twente, the Netherlands.

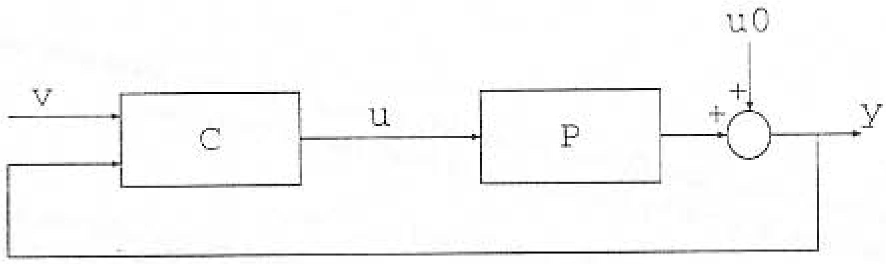

Fig.1. Configuration of feedback control system

freedom controller be described as

$$u = C_1(s)v - C_2(s)y \tag{1}$$

The controller must be such that its transfer function is proper and the obtained control system is internally stable. Define the polynomial matrix fraction description of the plant $P(s)$.

$$P(s) = D_{p1}^{-1}(s)N_{p1}(s) = N_{p2}(s)D_{p2}^{-1}(s) \tag{2}$$

where $(D_{p1}(s), N_{p1}(s))$ is any left and $(N_{p2}(s), D_{p2}(s))$ is any right coprime pair of polynomial matrices. Then there exist polynomial matrices X_0, Y_0, X_1 and Y_1 such that

$$D_{p1}X_1 + N_{p1}Y_1 = I, \qquad X_0D_{p2} + Y_0N_{p2} = I \tag{3}$$

Let the controller

$$C = [C_1 \quad -C_2] = D_c^{-1}[N_\pi \quad -N_f] \tag{4}$$

be a left coprime matrix-fraction description. The system will be internally stable if and only if D_c and N_f satisfy the unilateral matrix equation

$$D_cD_{p2} + N_fN_{p2} = 1 \tag{5}$$

Thus, the set of all stabilizing controllers C is represented by (6)

$$C = (X_0 - TN_{p1})^{-1}[N_\pi - (Y_0 + TD_{p1})] \tag{6}$$

where N_π and T are arbitrary polynomial matrices. By a straightforward analysis of Fig.1, it follows that the transfer function H_{vy} from v to y is

$$H_{vy} = N_{p2}N_\pi \tag{7}$$

and the transfer function from u_0 to y is

$$H_{u_0y} = I - N_{p2}N_f \tag{8}$$

These equations show that H_{vy} and H_{u_0y} can be independently designed through the choice of two free matrix parameters N_π and T. These matrix parameters will be then redefined by considering additional constraints on H_{vy} and H_{u_0y}. For example, consider the tracking constraint. Let command input v be described as

$$v = D_v^{-1}N_vv_0 \tag{9}$$

where v_0 is any constant vector, and D_v and N_v are left coprime. The free parameter matrix N_π for the tracking problem is given by the solution of bilateral matrix equation

$$N_{p2} N_\pi + M D_v = I \tag{10}$$

It has also been shown that the regulator problem can be related to the solution of such a bilateral equation. Therefore, the unilateral and bilateral equations play the key role in the algebraic approach.

Table 1.

Function name	Arguments	Results
!*BILATERAL	A,B,C,var,arb	Solutions of the bilateral equation AX+YB=C.
!*DIAG!-MAT	A,var	Diagonalised matrix of A.
!*DIOPHANTINE	a,b,c,var,arb	Solutions of the Diophantine equation ax+by=c .
!*DIVISOR!-L(R)	A,B,var	U and V such that A=BU+V(A=UB+V), where deg V < deg B.
!*EXTRACTION!-L(R)	C,G,var,arb	Left (Right) extraction, C' such that C=GC'(C=C'G).
!*GCL(R)D	A,B,var	The Left(Right) greatest common divisor of A and B.
!*MFD!-L(R)	F,var	Left(Right) coprime fraction of F.
!*SMITH	A,var	Smith form of A.
!*UNILATERAL!-L(R)	A,B,C,var,arb	Solutions of the unilateral equation AX+BY=C(AX+YB=C).

*F=rational matrix; A,B,C and G=polynomial matrices; a,b, c=polynomials; var=main variable; arb=free parameter for the solutions.

The software package for solving above equations (1)-(10) is developed based on REDUCE. The main functions of our package are listed in Table 1. All functions are programmed by RLISP. Of course before our study, some packages based on numerical computation were developed and numerical algorithms were also studied, for example,the matrix fraction description or bilateral equation. These algorithms are very complex and have no logical connection mutually because they avoid the symbolic or direct polynomial manipulations. If the general formula manipulation can be used, systematic algorithms which are obtained in linear algebraic theory can be adopted. In fact, the algorithms in Kucera [5] are used in our study, arranged for symbolic manipulation and have the hierarchical structure given by Fig.2. Then the upper functions were efficiently written by calling lower functions.

Fig. 3 shows the results of function BILATERAL with matrices A, B, C given by

$$A = \begin{bmatrix} 1 & 0 & -s & s^2 \\ 0 & s(s^2-1) & s^2(-s^2+1) & s(-s^2+1) \\ 1 & s^2-1 & -s^2 & 1 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & -s+1 \end{bmatrix} \qquad C = \begin{bmatrix} 1 & -s+1 \\ 0 & s^4+s^2-s-1 \\ 1 & s^3+1 \end{bmatrix}$$

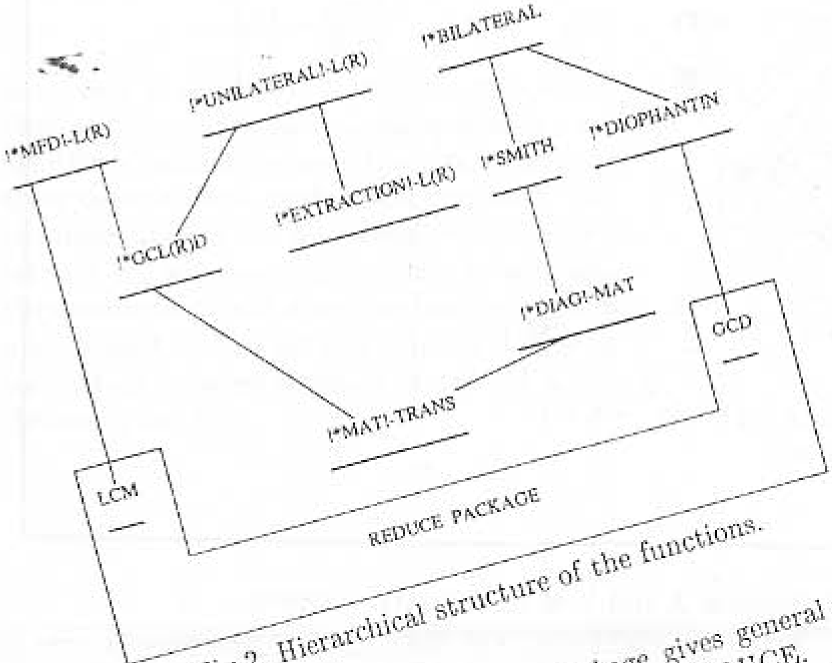

Fig.2. Hierarchical structure of the functions.

In Fig.3. R means arbitrary polynomial. Our package gives general solutions and can be performed by menu system with less knowledge of LISP or REDUCE.

The package dose not only include the results by Tanttu and Altonen [2] but can be extended to the stable rational function ring and the general polynomial ring given by Pernebo [4].

3 2-D system theory and parallel computing algebra

Over the last two decades, the theory of 2 or n-dimensional (2-D or n-D) systems that are characterizable by two or n independent variables, has been investigated. Analogously to the one dimensional case represented in Section 2, by polynomial approach, the feedback stabilization problem of 2D systems can also be reduced to solving the unilateral matrix equation

$$A(v,w)X(v,w) + B(v,w)Y(v,w) = \Phi(v,w) \tag{13}$$

and the tracking problem to solving the bilateral matrix equation

$$A(v,w)X(v,w) + Y(v,w)B(v,w) = \Theta(v,w) \tag{14}$$

where $A(v,w)$, $B(v,w)$, $X(v,w)$, $Y(v,w)$, $\Phi(v,w)$ and $\Theta(v,w)$ are appropriate 2-D polynomial matrices and $\det\Phi(v,w)$ and $\det\Theta(v,w)$ are stable, i.e., devoid of zeroes in the unit bidisc $\bar{U}^2$. However, since n-D ($n > 1$) polynomial ring is not Euclidean and has many properties substantially different from the 1-D case, such as coprimeness, the existing 1-D algorithm cannot be applied to solve Equations (13) and (14). The solvability conditions and some solutio procedures have been proposed recently (see, e.g., [8]-[10]), and these algorithms have also be implemented in our package.

In the implementation by using the geneal formula manipulation system as in Section was found that some problems may happen. For example, the computing speed of the exi formula system is not so enough, or middle results are often exploded because non Euc ing must be treated.

```
xterm
X :=
     |-                                                                   -|
     |           2                                        2                |
     | R4*S - R6*S  + 1   R1*( - S + 1) - R3*S + R5*S  + 2                  |
     |                                                                     |
     | R4*S - R6          - R2 - R3*S - R5                                 |
     |                                                                     |
     | R4                 - R3                                             |
     |                                                                     |
     | R6                 - R5                                             |
     |                                                                     |
     |-                                                                   -|

Y :=
     |-                                              -|
     |  - R1 - 1                                      |
     |                                                |
     |                            3    2              |
     |  - (R2*S)*(S + 1) - S  - S  - 2*S - 1          |
     |                                                |
     |                            2                   |
     |  - R1 - R2*(S + 1) - (S  + S + 1)              |
     |                                                |
     |-                                              -|
```

Fig.3. Solution X and Y of 1-D bilateral equation.

Hereupon the parallel formula manipulation system for 2-D system theory is being developed for avoiding above problems. The parallel system should be studied from two view points. One is the parallel computer architecture and the other is the parallel algorithm. Here the Single Instruction stream Multi Data stream (SIMD) architecture is adopted. The practical parallel computer system, named as SM-1, is implemented by Yuasa's group. SM-1 has the Front End (FE) computer (sparc EWS) and 1024 Processing Elements (PE) which have their own 32bit processor and 1Mbyte memory, illustrated in Fig.4. The parallel formula manipulation system is implemented with parallel Kyoto Common LISP on SM-1. As SIMD architecture is adopted, the parallel algorithm has to be constructed with same instruction, broadcasted to each PE.

For example, a function is calculating, at a stretch, all the maximal minor determinants of the composite matrix $[A(v,w)B(v,w)]$, which is necessary for solving Equation (13) by the procedure given in [8][9]. The function has a facility of broadcasting a instruction of calculating the determinant with each data from FE to PE. Each PE calculates minor determinant simultaneously with Laplace algorithm and feedbacks the result to FE. Fig.5 shows the result of calculating 6 deteminants $a_1, .., a_6$ with our function for the (2×4) composite matrix, given as

$$A = \begin{bmatrix} 3(2v-1)(2v-5)/16 & 0 \\ -3w^2(2w-3)(2v-5)/2 & 2(8w+6v-15) \end{bmatrix} \tag{15}$$

$$B = \begin{bmatrix} -3(2v-1)(w-3v)/16 & (2v-5)^2/16 \\ (6w^4-18w^3v-9w^3+27w^2v+8v-4)/2 & -w^2((2w-3)(2v-5)-8w)/2 \end{bmatrix} \tag{16}$$

Another function is calculating the inverse matrix on 2 variable polynomial ring. Conventionally the inverse matrix is obtained with Gauss Jordan method. But it is not appropriate because 2 variable polynomial is not Euclidean. Here

$$A^{-1} = \frac{adj\, A}{det A} \tag{17}$$

is applied. It consists of calculating $(n \times n) + 1$ determinants. The conventional system can not use this algorithm because the computation of these determinants needs much cpu time. But the determinants can be gotten simultaneously on each PE with same instruction. In addition, the denominator, det A and the nominator polynomial adj$A$ of each entry is reduced at once by calculating greatest common divisor. Fig.6 is the result for A

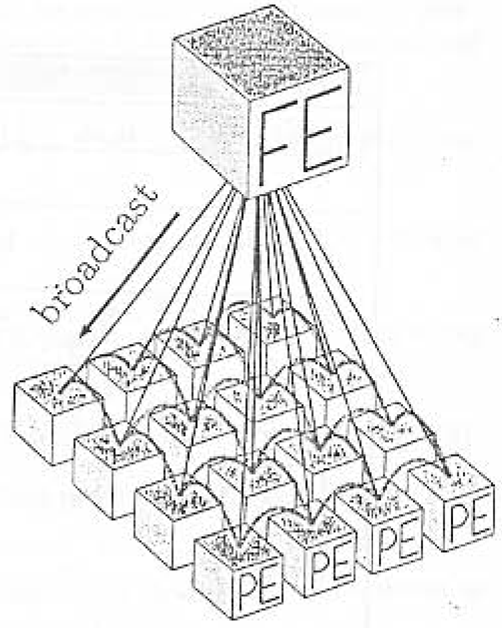

Fig.4. Architecture of SM-1

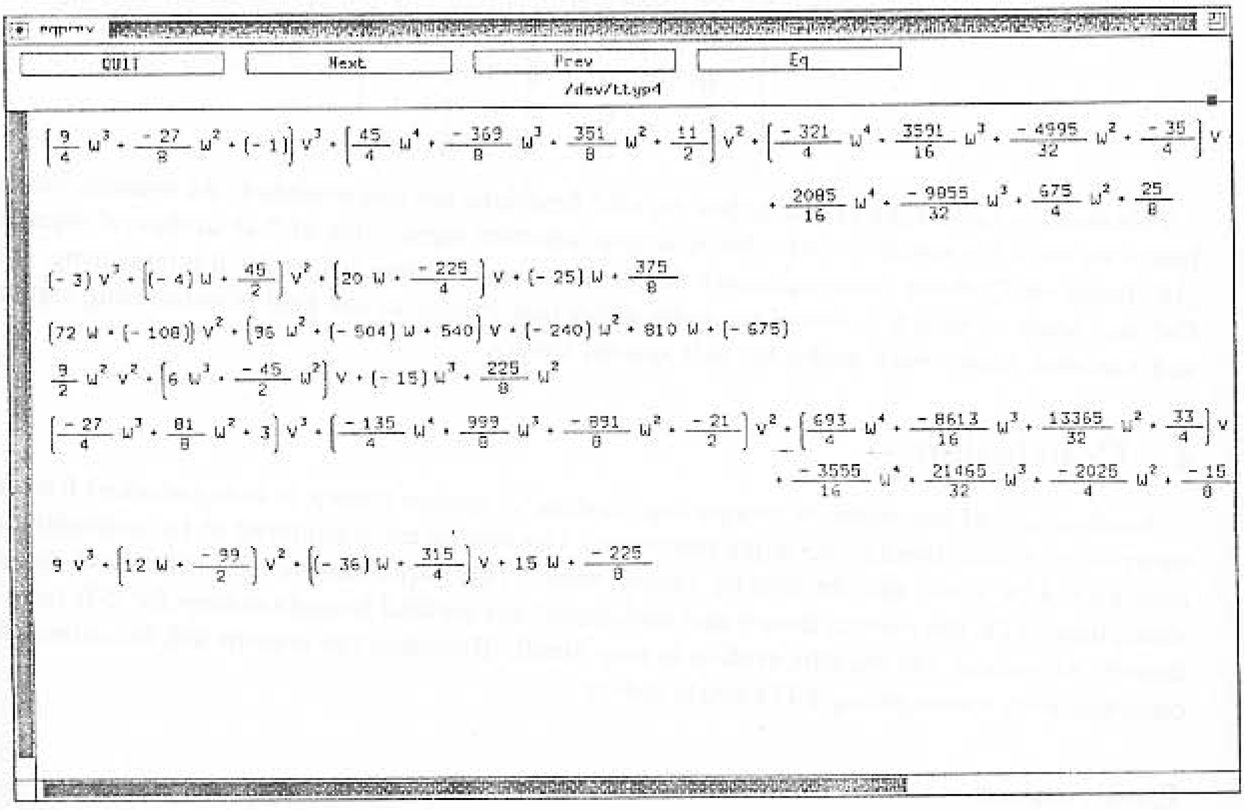

Fig.5. Results of minor determinants.

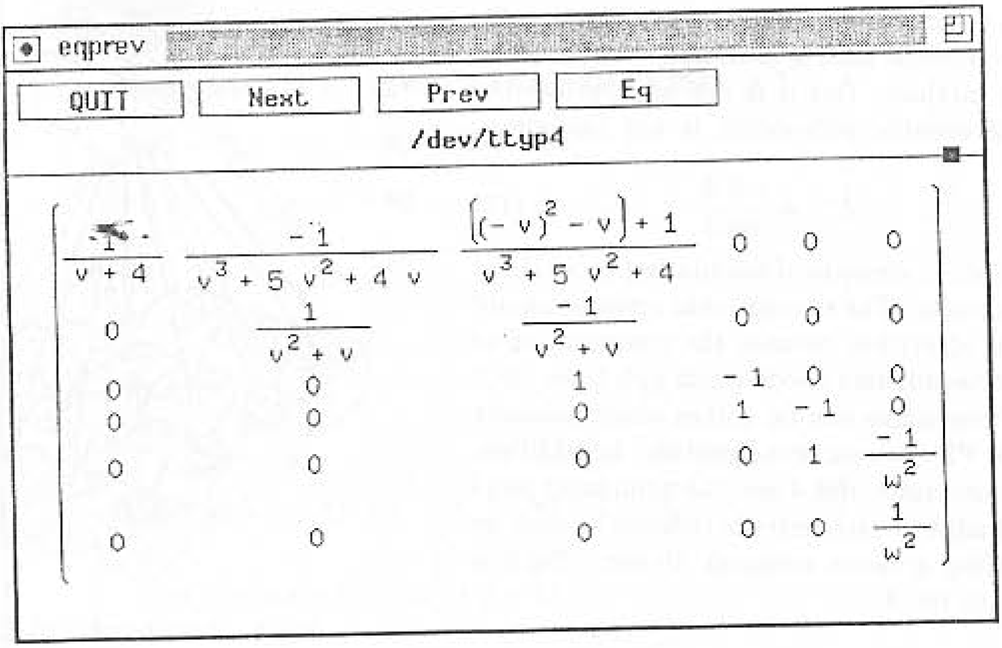

Fig.6. Result of inverse matrix.

$$A = \begin{bmatrix} v+4 & 1 & 1 & 1 & 1 & 1 \\ 0 & v^2+v & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & w^2 \end{bmatrix} \tag{18}$$

This study is just began and only few parallel functions are implemented. At present, various functions for 2-D system theory , for example, solution algorithms of 2-D unilateral equation (13) based on Gröbner bases approach for polynomial modules are under implementing. The Gröbner bases approach is one of the most important results in the field of computing algebra, and has been shown very usuful for n-D system theory.

4 Conclusion

Applications of computer or computing algebra to system theory is being studied from the viewpoints of both theory and implementation. The results are considered to be interesting not only for CADCS but also for control theory itself. This paper showed the utilities of formula manipulation for the control theory and introduced our parallel formula system for 2-D (or n-D) theory. At present the parallel system is very small. Hereafter the system will be extended in company with investigating 2-D control theory.

References

[1] W. Wolovich: Linear Multivariable System. New York Spriger-Verlag (1974)

[2] J. Tanttu and M. Altonen: Max—a program system for symbolic manipulation of polynomial matrices, Proc.3rd IFAC/IFIP Symp. on Computer Aided Design in Control and Engineering System (1985)

[3] O. Saito, M. Kanno and K. Abe: Symbolic manipulation CAD of control engineering, Int. J. Cont. Vol.48, No.2 (1988)

[4] L. Pernebo: An algebraic theory for the design of controllers for linear multivariable system IEEE Trans. AC-26 (1981)

[5] V. Kucera: Discrete Linear Control:The Polynomial Equation Approach, New York Wiley (1979)

[6] S. Zak: On the polynomial matrix equation $AX + YB = C$, IEEE Trans. AC-30 (1985)

[7] B. Buchburger: Grobner bases, An Algorithme Method in Polynomial Ideal Theory, Multidimensional System Theory, Dordrecht (1985)

[8] J. Guiver and N. Bose: Cousal and Weakly Cousal 2-D Filters with Applications in Stabilization, Multidimensional System Theory, Dordrect (1985)

[9] M. Bisiacco, R. Fornasini and G.Marchesini: Controoer Design for 2-D System, Frequency Domain and State Space Methods for Linear System, Elservier (1986)

[10] L. Xu, O.saito and K.Abe: Bilateral Polynomial Matrix Equation in Two Indeterminates, Multidimensional System and Signal Processing, Vol.1 (1990)