Int. J. Appl. Math. and Comp. Sci., 2000, Vol.10, No.2, 321–331

FORNASINI-MARCHESINI 2-D STATE SPACE MODELS: TRANSFER FUNCTION COMPUTATION VIA THE DFT[†]

GEORGE E. ANTONIOU*, KELLY EMMONS*

A new algorithm is presented for the computation of the coefficients of the determinantal polynomial and the coefficients of the adjoint polynomial matrix of a given two-dimensional (2-D) state space model of Fornasini-Marchesini type. The algorithm has been implemented in Matlab and uses the discrete Fourier transform (DFT). The simplicity and efficiency of the technique are illustrated by two examples.

Keywords: two dimensions system theory, Fourier transform, DFT algorithm, Fornasini-Marchesini's model

1. Introduction

During the past two decades there has been extensive research on two-dimensional (2-D) systems. These systems describe physical processes which are characterized by two independent variables. Applications of 2-D systems can be found in image processing, computer tomography, geophysics, multipass processes, etc. (Dudgeon and Mersereau, 1984; Gałkowski, 1994; Kaczorek, 1985). State space techniques play a very important role in the analysis and synthesis of 2-D systems (Gałkowski, 1994). An important problem is to determine the coefficients of a transfer function from its state space representation and vice versa. In going from the transfer function to a state space model a number of algorithms have been proposed. In the case where a state space model is available the Leverrier and DFT algorithms with Vandermonde matrices can be used (Antoniou, 1997; Antoniou *et al.*, 1989; Lee, 1976; Luo *et al.*, 1997; Paccagnella and Pierobon, 1976; Yeung and Kumbi, 1988). The DFT has been used for the evaluation of the transfer functions for linear systems (Lee, 1976; Paccagnella and Pierobon, 1976), singular systems (Antoniou *et al.*, 1989), and multidimensional systems (Yeung and Kumbi, 1988) of the Roesser type. Recently the DFT was used in determining the coefficients of the Attasi-Kaczorek singular 2-D model (Antoniou, 1997).

In this paper, a computer implementable algorithm is proposed, using the DFT, for the computation of the 2-D transfer function for the Fornasini-Marchesini 2-D

† This work was supported by the Margaret and Herman Sokol Faculty Award and the MSU Student-Faculty Program Award.

* Image Processing and Systems Laboratory, Department of Computer Science, Montclair State University, Montclair, NJ 07043, U.S.A., e-mail: george.antoniou@montclair.edu

state space models (Fornasini and Marchesini, 1978). The proposed algorithm determines the coefficients of the determinantal polynomial and the coefficients of the adjoint polynomial matrix, using the DFT algorithm. The computational speed of the algorithm can be improved using fast Fourier techniques to implement the DFT (Oppenheim *et al.*, 1999). It is noted that the algorithm has been implemented using Matlab.

2. Background

Two-dimensional (2-D) state space models of the Fornasini-Marchesini (F-M) type have the following structure (Fornasini and Marchesini, 1978):

First F-M model

$$\begin{cases} \boldsymbol{x}(i+1,j+1) = \boldsymbol{A}_1\boldsymbol{x}(i+1,j) + \boldsymbol{A}_2\boldsymbol{x}(i,j+1) + \boldsymbol{b}\boldsymbol{u}(i,j), \\ \boldsymbol{y}(i,j) = \boldsymbol{c}'\boldsymbol{x}(i,j), \end{cases} \tag{1}$$

Second F-M model

$$\begin{cases} \boldsymbol{x}(i+1,j+1) = \boldsymbol{A}_1\boldsymbol{x}(i+1,j) + \boldsymbol{A}_2\boldsymbol{x}(i,j+1) \\ \qquad\qquad + \boldsymbol{b}_1\boldsymbol{u}(i+1,j) + \boldsymbol{b}_2\boldsymbol{u}(i,j+1), \\ \boldsymbol{y}(i,j) = \boldsymbol{c}'\boldsymbol{x}(i,j), \end{cases} \tag{2}$$

where $\boldsymbol{x}(i,j) \in \mathbb{R}^n$, $\boldsymbol{u}(i,j) \in \mathbb{R}^m$, $\boldsymbol{y}(i,j) \in \mathbb{R}^p$; $\boldsymbol{A}_k$, $k = 1,2$ and $\boldsymbol{b}$, $\boldsymbol{c}'$ are real matrices of appropriate dimensions.

Applying the 2-D z-w transforms to the systems (1) and (2) with zero initial conditions, their transfer functions respectively become

$$\boldsymbol{T}_1(z,w) = \boldsymbol{c}' \left[\boldsymbol{I}zw - \boldsymbol{A}_1 z - \boldsymbol{A}_2 w\right]^{-1}\boldsymbol{b} \tag{3}$$

and

$$\boldsymbol{T}_2(z,w) = \boldsymbol{c}' \left[\boldsymbol{I}zw - \boldsymbol{A}_1 z - \boldsymbol{A}_2 w\right]^{-1}(\boldsymbol{b}_1 z + \boldsymbol{b}_2 w). \tag{4}$$

In the following section, an interpolative approach is developed for determining the transfer function $\boldsymbol{T}(s)$, given the matrices $\boldsymbol{A}_k$, $k = 1,2$ and $\boldsymbol{b}$, $\boldsymbol{c}'$, using the DFT. For the sake of completeness, a brief description of the DFT follows.

2.1. 2-D DFT

Given finite sequences $X(k_1,k_2)$ and $\tilde{X}(r_1,r_2)$, $k_1,r_1 = 0,\dots,M$ and $k_2,r_2 = 0,\dots,N$, the following relationships are necessary in order for the sequences to constitute a 2-D DFT pair (Oppenheim *et al.*, 1999):

$$\tilde{X}(r_1, r_2) = \sum_{k_1=0}^{M} \sum_{k_2=0}^{N} X(k_1, k_2) W_1^{-k_1 r_1} W_2^{-k_2 r_2}, \tag{5}$$

$$X(k_1, k_2) = \frac{1}{(M+1)(N+1)} \sum_{r_1=0}^{M} \sum_{r_2=0}^{N} \tilde{X}(r_1, r_2) W_1^{k_1 r_1} W_2^{k_2 r_2}, \tag{6}$$

where $r_1 = 0, \ldots, M,\ r_2 = 0, \ldots, N,\ k_1 = 0, \ldots, M,\ k_2 = 0, \ldots, N,$

$$X = [x_{ij}], \quad \tilde{X} = [\tilde{x}_{ij}], \quad i = 1, \ldots, p, \quad j = 1, \ldots, m$$

and

$$W_1 = e^{2\pi j/(M+1)}, \quad W_2 = e^{2\pi j/(N+1)}$$

3. First F-M Model: Algorithm

Let the transfer function, $\boldsymbol{T}(z, w)$, of the first F-M 2-D state space model be defined as

$$\boldsymbol{T}(z, w) = \frac{\boldsymbol{N}(z, w)}{d(z, w)}, \tag{7}$$

where

$$\boldsymbol{N}(z, w) = \boldsymbol{c}' \operatorname{adj}[\boldsymbol{I}zw - \boldsymbol{A}_1 z - \boldsymbol{A}_2 w]\boldsymbol{b}, \tag{8}$$

$$d(z, w) = \det[\boldsymbol{I}zw - \boldsymbol{A}_1 z - \boldsymbol{A}_2 w]. \tag{9}$$

Note that $\deg_z[\boldsymbol{N}(z, w)] = \deg_w[\boldsymbol{N}(z, w)] = n$ and $\deg_z[d(z, w)] = \deg_w[d(z, w)] = n$, where $\deg_z[\cdot]$ and $\deg_w[\cdot]$ denote the degrees with respect to z and w, respectively. Consequently, (8) and (9) can be written in polynomial form as follows:

$$\boldsymbol{N}(z, w) = \sum_{k=0}^{n} \sum_{r=0}^{n} \boldsymbol{P}_{kr} z^k w^r, \tag{10}$$

$$d(z, w) = \sum_{k=0}^{n} \sum_{r=0}^{n} q_{kr} z^k w^r, \tag{11}$$

where $\boldsymbol{P}_{kr}$ are matrices with dimensions $p \times m$, while q_{kr} are scalars.

The numerator polynomial matrix $\boldsymbol{N}(z, w)$ and the denominator polynomial $d(z, w)$ can be numerically computed at $(n+1)^2$ points, evenly spaced on the unit 2-D disc. The $(n+1)^2$ points can be chosen as $(z, w) = [v(i), v(j)],\ i, j = 0, \ldots, n,$ or according to our definition as

$$v_1(r) = v_2(r) = W^{-r}, \quad r = 0, \ldots, n, \tag{12}$$

where

$$W_1 = W_2 = W = e^{2\pi j/(n+1)}. \tag{13}$$

The values of the transfer function (7) at the $(n+1)^2$ points form its corresponding 2-D DFT coefficients.

3.1. Denominator Polynomial

To evaluate the denominator coefficients q_{kr}, define

$$a_{ij} = \det\big[\boldsymbol{I}v_1(i)v_2(j) - \boldsymbol{A}_1 v_1(i) - \boldsymbol{A}_2 v_2(j)\big]. \tag{14}$$

Using (9) and (14), a_{ij} can also be defined as

$$a_{ij} = d\big[v_1(i), v_2(j)\big] \tag{15}$$

provided that at least one of $a_{ij} \neq 0$.

Equations (11), (12) and (15) yield

$$a_{ij} = \sum_{k=0}^{n}\sum_{r=0}^{n} q_{kr} W^{-(ik+jr)}. \tag{16}$$

In (16), $[a_{ij}]$, $[q_{kr}]$ form a DFT pair. Therefore the coefficients q_{kr} can be computed using the inverse 2-D DFT as follows:

$$q_{kr} = \frac{1}{(n+1)^2}\sum_{i=0}^{n}\sum_{j=0}^{n} a_{ij} W^{ik+jr}, \tag{17}$$

where $k, r = 0, \ldots, n$.

3.2. Numerator Polynomial

To evaluate the numerator matrix polynomial $\boldsymbol{P}_{kr}$, define

$$\boldsymbol{F}_{ij} = \boldsymbol{c}'\mathrm{adj}\big[\boldsymbol{I}v_1(i)v_2(j) - \boldsymbol{A}_1 v_1(i) - \boldsymbol{A}_2 v_2(j)\big]\boldsymbol{b} \tag{18}$$

provided that at least one of $\boldsymbol{F}_{ij} \neq 0$. Using (8) and (18), $\boldsymbol{F}_{ij}$ can also be defined as

$$\boldsymbol{F}_{ij} = \boldsymbol{N}\big[v_1(i), v_2(j)\big]. \tag{19}$$

Equations (10), (12) and (19) yield

$$\boldsymbol{F}_{ij} = \sum_{k=0}^{n}\sum_{r=0}^{n} \boldsymbol{P}_{kr} W^{-(ik+jr)}. \tag{20}$$

In (20), $[\boldsymbol{F}_{ij}]$, $[\boldsymbol{P}_{kr}]$ form a DFT pair. Therefore the coefficients P_{kr} can be computed using the inverse 2-D DFT as follows:

$$\boldsymbol{P}_{kr} = \frac{1}{(n+1)^2}\sum_{i=0}^{n}\sum_{j=0}^{n} \boldsymbol{F}_{ij} W^{ik+jr}, \tag{21}$$

where $k, r = 0, \ldots, n$.

Finally, the transfer function sought is

$$\boldsymbol{T}(z,w) = \frac{\boldsymbol{N}(z,w)}{d(z,w)}, \tag{22}$$

where

$$d(z,w) = \sum_{k=0}^{n}\sum_{r=0}^{n} q_{kr} z^k w^r, \tag{23}$$

$$\boldsymbol{N}(z,w) = \sum_{k=0}^{n}\sum_{r=0}^{n} \boldsymbol{P}_{kr} z^k w^r. \tag{24}$$

A synopsis of the presented algorithm is given in Table 1.

Table 1. Algorithm for the first F-M model.

Let $n = \dim \boldsymbol{I} = \dim \boldsymbol{A}_1 = \dim \boldsymbol{A}_1$ $W = e^{2\pi j/(n+1)}$
for $r = 0$ to n do $v_1(r) = v_2(r) = W^{-r}, \quad r = 0, \dots, n$

for $i = 0$ to n do
for $j = 0$ to n do
$a_{ij} = \det[\boldsymbol{I}v_1(i)v_2(j) - \boldsymbol{A}_1 v_1(i) - \boldsymbol{A}_2 v_2(j)]$
$\boldsymbol{F}_{ij} = \boldsymbol{c}' \mathrm{adj}\,[\boldsymbol{I}zw - \boldsymbol{A}_1 z - \boldsymbol{A}_2 w]^{-1} \boldsymbol{b}$
end for
end for

for $i = 0$ to n do
for $j = 0$ to n do
$q_{kr} = \frac{1}{(n+1)^2} \sum_{i=0}^{n}\sum_{j=0}^{n} a_{ij} W^{ik+jr}$
$\boldsymbol{P}_{k,r} = \frac{1}{(n+1)^2} \sum_{i=0}^{n}\sum_{j=0}^{n} \boldsymbol{F}_{ij} W^{ik+jr}$
end for
end for

$\boldsymbol{T}(z,w) = \frac{\boldsymbol{N}(z,w)}{d(z,w)} = \frac{\sum_{k=0}^{n}\sum_{r=0}^{n} \boldsymbol{P}_{kr} z^k w^r}{\sum_{k=0}^{n}\sum_{r=0}^{n} q_{kr} z^k w^r}$

3.3. Example

Consider the system described by the following 2-D state space model:

$$\begin{aligned} \boldsymbol{x}(i+1,j+1) &= \boldsymbol{A}_1\boldsymbol{x}(i+1,j) + \boldsymbol{A}_2\boldsymbol{x}(i,j+1) + \boldsymbol{b}u(i,j), \\ y(i,j) &= \boldsymbol{c}'\boldsymbol{x}(i,j), \end{aligned}$$

where

$$\boldsymbol{A}_1 = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad \boldsymbol{A}_2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix},$$

$$\boldsymbol{b} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \qquad \boldsymbol{c}' = \begin{bmatrix} 1 & 0 \end{bmatrix}.$$

We would like to determine the transfer function for this system using the technique outlined above.

The direct application of the proposed algorithm yields

$$\begin{bmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \\ a_{20} & a_{21} & a_{22} \end{bmatrix} = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 0.5-2.5981j & 2.0 \\ -1 & 2 & 0.5+2.5981j \end{bmatrix}$$

and

$$\begin{bmatrix} F_{00} & F_{01} & F_{02} \\ F_{10} & F_{11} & F_{12} \\ F_{20} & F_{21} & F_{22} \end{bmatrix} = \begin{bmatrix} 0 & -1.5-0.8660j & -1.5+0.8660j \\ 0 & 1.7321j & 1.5+0.8660j \\ 0 & 1.5-0.8660j & -1.7321j \end{bmatrix}.$$

Using (17), the denominator coeffients are

$$\begin{bmatrix} q_{00} & q_{01} & q_{02} \\ q_{10} & q_{11} & q_{12} \\ q_{20} & q_{21} & q_{22} \end{bmatrix} = \begin{bmatrix} 0 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}.$$

From (21), the numerator matrix polynomials are

$$\begin{bmatrix} P_{00} & P_{01} & P_{02} \\ P_{10} & P_{11} & P_{12} \\ P_{20} & P_{21} & P_{22} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

Once the denominator and the adjoint matrix have been computed, (22) can be utilized to obtain the transfer function $T(z,w)$. Therefore we obtain

$$T(z,w) = \frac{P_{11}zw + P_{10}z}{z^2w^2 - z^2 - w^2} \tag{25}$$

or

$$T(z,w) = \frac{zw - z}{z^2w^2 - z^2 - w^2}. \tag{26}$$

4. Second F-M Model: Algorithm

Let the transfer function, $\boldsymbol{T}(z,w)$, of the second F-M 2-D state space model be defined as

$$\boldsymbol{T}(z,w) = \frac{\boldsymbol{N}(z,w)}{d(z,w)}, \tag{27}$$

where

$$\boldsymbol{N}(z,w) = \boldsymbol{c}' \operatorname{adj}[\boldsymbol{I}zw - \boldsymbol{A}_1 z - \boldsymbol{A}_2 w][\boldsymbol{b}_1 z + \boldsymbol{b}_2 w], \tag{28}$$

$$d(z,w) = \det[\boldsymbol{I}zw - \boldsymbol{A}_1 z - \boldsymbol{A}_2 w]. \tag{29}$$

Note that we have $\deg_z[\boldsymbol{N}(z,w)] = \deg_w[\boldsymbol{N}(z,w)] = n$ and $\deg_z[d(z,w)] = \deg_w[d(z,w)] = n$, where $\deg_z[\cdot]$ and $\deg_w[\cdot]$ denote the degrees with respect to z and w, respectively. Equations (28) and (29) can be written in polynomial form as follows:

$$\boldsymbol{N}(z,w) = \sum_{k=0}^{n}\sum_{r=0}^{n} \boldsymbol{P}_{kr} z^k w^r, \tag{30}$$

$$d(z,w) = \sum_{k=0}^{n}\sum_{r=0}^{n} q_{kr} z^k w^r, \tag{31}$$

where $\boldsymbol{P}_{kr}$ are matrices of dimensions $p \times m$, while q_{kr} are scalars.

The numerator and denominator polynomials (30), (31) can be numerically computed at $(n+1)^2$ points, evenly spaced on the unit 2-D disc. The $(n+1)^2$ points can be choosen as $(z,w) = [v(i), v(j)]$, $i,j = 0,\dots,n$, or according to our definition as

$$v_1(r) = v_2(r) = W^{-r}, \quad r = 0,\dots,n, \tag{32}$$

where

$$W_1 = W_2 = W = e^{2\pi j/(n+1)}. \tag{33}$$

The values of the transfer function (27) at the $(n+1)^2$ points form its corresponding 2-D DFT coefficients.

4.1. Denominator Polynomial

To evaluate the denominator coefficients (q_{kr}), define

$$a_{ij} = \det\big[\boldsymbol{I}v_1(i)v_2(j) - \boldsymbol{A}_1 v_1(i) - \boldsymbol{A}_2 v_2(j)\big]. \tag{34}$$

Using (9) and (34), a_{ij} can also be defined as

$$a_{ij} = d\big[v_1(i), v_2(j)\big] \tag{35}$$

provided that at least one of $a_{ij} \neq 0$.

Equations (31), (32) and (35) yield

$$a_{ij} = \sum_{k=0}^{n}\sum_{r=0}^{n} q_{kr} W^{-(ik+jr)}. \tag{36}$$

Here $[a_{ij}]$, $[q_{kr}]$ form a DFT pair. Therefore the coefficients q_{kr} can be computed using the inverse 2-D DFT as follows:

$$q_{kr} = \frac{1}{(n+1)^2}\sum_{i=0}^{n}\sum_{j=0}^{n} a_{ij} W^{ik+jr}, \tag{37}$$

where $k, r = 0, \ldots, n$.

4.2. Numerator Polynomial

To evaluate the numerator matrix polynomial $\boldsymbol{P}_{kr}$, define

$$\boldsymbol{F}_{ij} = \boldsymbol{c}'\mathrm{adj}\big[\boldsymbol{I}v_1(i)v_2(j) - \boldsymbol{A}_1 v_1(i) - \boldsymbol{A}_2 v_2(j)\big]\big[\boldsymbol{b}_1 v_1(i) + \boldsymbol{b}_2 v_2(j)\big] \tag{38}$$

provided that at least one of $\boldsymbol{F}_{ij} \neq 0$. Using (28) and (38), $\boldsymbol{F}_{ij}$ can also be written down as

$$\boldsymbol{F}_{ij} = \boldsymbol{N}\big[v_1(i), v_2(j)\big]. \tag{39}$$

Equations (30), (32) and (39) yield

$$\boldsymbol{F}_{ij} = \sum_{k=0}^{n}\sum_{r=0}^{n} \boldsymbol{P}_{kr} W^{-(ik+jr)}. \tag{40}$$

In (40), $[\boldsymbol{F}_{ij}]$, $[\boldsymbol{P}_{kr}]$ form a DFT pair. Therefore the coefficients $\boldsymbol{P}_{kr}$ can be computed using the inverse 2-D DFT as follows:

$$\boldsymbol{P}_{kr} = \frac{1}{(n+1)^2}\sum_{i=0}^{n}\sum_{j=0}^{n} \boldsymbol{F}_{ij} W^{ik+jr}, \tag{41}$$

where $k, r = 0, \ldots, n$.

Finally, the transfer function sought is

$$\boldsymbol{T}(z, w) = \frac{\boldsymbol{N}(z, w)}{d(z, w)}, \tag{42}$$

where

$$d(z, w) = \sum_{k=0}^{n}\sum_{r=0}^{n} q_{kr} z^k w^r, \tag{43}$$

$$\boldsymbol{N}(z, w) = \sum_{k=0}^{n}\sum_{r=0}^{n} \boldsymbol{P}_{kr} z^k w^r. \tag{44}$$

4.3. Example

Consider the system described by the following 2-D state space model:

$$\begin{aligned} \boldsymbol{x}(i+1,j+1) &= \boldsymbol{A}_1\boldsymbol{x}(i+1,j) + \boldsymbol{A}_2\boldsymbol{x}(i,j+1) + \boldsymbol{b}_1 u(i+1,j) + \boldsymbol{b}_2 u(i,j+1), \\ y(i,j) &= \boldsymbol{c}'\boldsymbol{x}(i,j), \end{aligned}$$

where

$$\boldsymbol{A}_1 = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \boldsymbol{A}_2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix},$$

$$\boldsymbol{b}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \boldsymbol{b}_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad \boldsymbol{c}' = \begin{bmatrix} 0 & 1 \end{bmatrix}$$

We would like to determine the transfer function for this system using the technique outlined above.

The direct application of the proposed algorithm yields

$$\begin{bmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \\ a_{20} & a_{21} & a_{22} \end{bmatrix} = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 0.5-2.598j & 2 \\ -1 & 2 & 0.5+2.598j \end{bmatrix}$$

and

$$\begin{bmatrix} F_{00} & F_{01} & F_{02} \\ F_{10} & F_{11} & F_{12} \\ F_{20} & F_{21} & F_{22} \end{bmatrix} = \begin{bmatrix} 3 & 1.5+0.8660j & 1.5-0.8660j \\ 1.5+2.5981j & -0.6429+0.1237j & 0.75-0.4330j \\ 1.5-2.5981j & 0.75+0.4330j & -0.4286+0.4949j \end{bmatrix}.$$

Using (37), the denonimator coefficients are

$$\begin{bmatrix} q_{00} & q_{01} & q_{02} \\ q_{10} & q_{11} & q_{12} \\ q_{20} & q_{21} & q_{22} \end{bmatrix} = \begin{bmatrix} 0 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}.$$

Based on (41), the matrix numerator matrix coefficients are

$$\begin{bmatrix} P_{00} & P_{01} & P_{02} \\ P_{10} & P_{11} & P_{12} \\ P_{20} & P_{21} & P_{22} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 1 \\ 0 & 0 & 0 \end{bmatrix}.$$

Once the denominator and the adjoint matrix have been computed, (42) can be utilized to obtain the transfer function $T(z,w)$. Therefore we obtain

$$T(z,w) = \frac{P_{11}zw + P_{12}zw^2}{z^2w^2 - z^2 - w^2}, \tag{45}$$

and finally

$$T(z,w) = \frac{2zw + zw^2}{z^2w^2 - z^2 - w^2}. \tag{46}$$

5. Complexity of the Algorithm

The proposed algorithm has two parts. In the first part the matrices $\boldsymbol{F}_{ij}$ and the scalars a_{ij} are evaluated with a cost of $O(n^3)$ operations. In the second part the coefficients of $\boldsymbol{P}_{kr}$ and q_{kr} are evaluated using the DFT with a cost of $O(pm(n+1)^4)$ operations. For more efficient computation, especially for high-order systems, the fast Fourier transform can be used to implement the DFT. In this case the coefficients of q_{kr} and $\boldsymbol{P}_{kr}$ are evaluated with computational costs $[(n+1)\log_2(n+1)][(n+1)\log_2(n+1)]$ and $pm[(n+1)\log_2(n+1)][(n+1)\log_2(n+1)]$, respectively (Oppenheim *et al.*, 1999).

6. Conclusion

In this paper, two algorithms are proposed for determining the coefficients of a 2-D transfer function from its state space representation, having the Fornasini-Marchesini stucture. The algorithms are simple and are based on the DFT. For the improvement of the computational speed, especially for high order systems, the DFT algorithm can be implemented with fast Fourier methods.

Acknowledgment

The authors would like to thank the reviewers for their valuable comments and suggestions.

References

Antoniou G.E. (1997): *DFT: Transfer function computation for the Attasi-Kaczorek 2-D state space model.* — Foundations of Computing and Decision Sciences, Vol.22, No.3, pp.211–218.

Antoniou G.E., Glentis G.O.A., Varoufakis S.J. and Karras D.A.,(1989): *Transfer function determination of singular systems using the DFT.* — IEEE Trans. Circuit Syst., Vol.CAS–36, No.8, pp.1140–1142.

Dudgeon D. and Mersereau R. (1984): *Multidimensional Digital Signal Processing.* — Englewood Cliffs, New Yersey: Prentice Hall.

Fornasini E. and Marchesini E. (1978): *Doubly indexed dynamical systems: State space models and structural properties.* — Math. Syst. Theory, Vol.12, No.1, pp.59–72.

Gałkowski K. (1994): *State Space Realizations on n-D Systems.* — Wrocław Technical University, Series Monographs No.76, Technical University Press, Wrocław, Poland.

Kaczorek T. (1985): *Two Dimensional Linear Systems.* — Lecture Notes in Control and Information Sciences, Berlin: Springer.

Lee T. (1976): *A simple method to determine the characteristic function $f(s) = |s\boldsymbol{I} - \boldsymbol{A}|$ by discrete Fourier series and fast Fourier transform.* — IEEE Trans. Circuit Syst., Vol.CAS–23, No.2, p.242.

Luo H., Lu W.-S. and Antoniou A. (1997): *New algorithms for the derivation of the transfer-function matrices of 2-D state-space discrete systems.* — IEEE Trans. Circuits Syst., I: Fundamental Theory and Applications, Vol.44, No.2, pp.112–119.

Oppenheim A.V., Scheafer R.W. and Buck J.R. (1999): *Digital Time Discrete Signal Processing.* — Englewood Cliffs, New Yersey: Prentice-Hall.

Paccagnella L.E. and Pierobon G.L. (1976): *FFT calculation of a determinental polynomial.* — IEEE Trans. Automat. Contr., Vol.AC–21, No.3, pp.401–402.

Yeung K.S. and Kumbi F. (1988): *Symbolic matrix inversion with application to electronic circuits.* — IEEE Trans. Circuit Syst., Vol.CAS–35, No.2, pp.235–239.

Received: 16 September 1999
Revised: 20 December 1999
Re-revised: 22 February 2000