BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES, Vol. 66, No. 5, 2018
DOI: 10.24425/124274

Digraph-building method for finding a set of minimal realizations of 2-D dynamic systems

K.A. MARKOWSKI* and K. HRYNIÓW

Warsaw University of Technology, 75 Koszykowa St., 00-662 Warsaw, Poland

Abstract. This paper presents a digraph-building method designed to find the determination of realization of two-dimensional dynamic system. The main differences between the method proposed and other state-of-the-art solutions used include finding a set of realizations (belonging to a defined class) instead of only one realization, and the fact that obtained realizations have minimal size of state matrices. In the article, the proposed method is described, compared to state-of-the-art methods and illustrated with numerical examples. To the best of authors' knowledge, the method shown in the paper is superior to all other state-of-the-art solutions both in terms of number of solutions and their matrix size. Additionally, MATLAB function for determination of realization based on the set of state matrices is included.

Key words: 2-D system, characteristic polynomial, digraph, minimal realization, MATLAB function.

1. Introduction

In the field of two-dimensional (2-D) systems, there are still problems that have not been solved completely, one of which is the realization problem, tackled for example in [1–7]. Each of proposed solutions comes with a set of different restrictions what realization can be obtained and none of them allows for the determination of all the possible solutions for given transfer function. Two-dimensional systems are used extensively for modeling in industrial processes, including for example distillation columns [8], chemical reactors, electronic and electrical circuits; biology and medicine [9]; transportation [10].

The realization problem is presented in detail in Section 2 along with the state-of-the-art. In Section 3, both sufficient conditions for digraph structure allowing to obtain a set of realizations in the class described and the method based on such digraphs, allowing to determine many realizations of 2-D system, are presented. In Section 4, the proposed method is compared to other algorithms, illustrating how the method works for examples used in different papers. Finally, in Section 5, the main points of article and the advantages of method are summarized.

Notation. In this paper the following notation will be used. The set $n\times m$ real matrices will be denoted by $\mathbb{R}^{n\times m}$ and $\mathbb{R}^n = \mathbb{R}^{n\times 1}$. If $\mathbf{G} = [g_{ij}]$ is a matrix, we write $\mathbf{G} \geqslant 0$ (matrix $\mathbf{G}$ is called non-negative), if $g_{ij} \geqslant 0$ for all i, j; $\mathbf{G} > 0$ (matrix $\mathbf{G}$ is called positive), if $\mathbf{G} \geqslant 0$ and any $g_{ij} > 0$; $\mathbf{G} \gg 0$ (matrix $\mathbf{G}$ is called strictly positive), if $g_{ij} > 0$ for all i, j. The set of $n\times m$ real matrices with non-negative entries will be denoted by $\mathbb{R}_+^{n\times m}$ and $\mathbb{R}_+^n = \mathbb{R}_+^{n\times 1}$. The $n\times n$ identity matrix will be denoted by $\mathbf{I}_n$. For more information about the matrix theory, an interested reader is referred, for instance, to [11, 12].

Model. Consider the two-dimensional (2D) general model $\Sigma = (\mathbf{A}_0, \mathbf{A}_1, \mathbf{A}_2, \mathbf{B}_0, \mathbf{B}_1, \mathbf{B}_2, \mathbf{C}, \mathbf{D})$ [13] described by the equation:

$$\begin{aligned} x_{i+1,j+1} &= \mathbf{A}_0 x_{i,j} + \mathbf{A}_1 x_{i,j+1} + \mathbf{A}_2 x_{i+1,j} + \\ &\quad + \mathbf{B}_0 u_{i,j} + \mathbf{B}_1 u_{i,j+1} + \mathbf{B}_2 u_{i+1,j}, \\ y(i,j) &= \mathbf{C} x_{i,j} + \mathbf{D} u_{i,j}, \end{aligned} \tag{1}$$

where $x_{i,j} \in \mathbb{R}^n$, $u_{i,j} \in \mathbb{R}^m$ and $y_{i,j} \in \mathbb{R}^p$ is state, input and output vector, respectively at the point (i, j) and

$$\begin{aligned} &\mathbf{A}_k \in \mathbb{R}^{n\times n},\ \mathbf{B}_k \in \mathbb{R}^{n\times m},\ k = 1, 2 \\ &\mathbf{C} \in \mathbb{R}^{p\times n},\ \mathbf{D} \in \mathbb{R}^{p\times m}. \end{aligned} \tag{2}$$

From (1) for $\mathbf{B}_1 = \mathbf{B}_2 = 0$ we obtain the first Fornasini-Marchesini model [13] and for $\mathbf{A}_0 = 0$ and $\mathbf{B}_0 = 0$ the second Fornasini-Marchesini model [13].

In this paper the special case of the general model (1) the second Fornasini-Marchesini (IIFM) model

$$\begin{aligned} x_{i+1,j+1} &= \mathbf{A}_1 x_{i,j+1} + \mathbf{A}_2 x_{i+1,j} + \\ &\quad + \mathbf{B}_1 u_{i,j+1} + \mathbf{B}_2 u_{i+1,j}, \\ y(i,j) &= \mathbf{C} x_{i,j} + \mathbf{D} u_{i,j}, \end{aligned} \tag{3}$$

will be considered.

The transfer matrix $\mathbf{T}(w_1, w_2) \in \mathbb{R}^{p\times m}$ of the model (3) is given by:

$$\begin{aligned} &\mathbf{T}(w_1, w_2) = \\ &= \mathbf{C}\left[\mathbf{I} - \mathbf{A}_1 w_1 - \mathbf{A}_2 w_2\right]^{-1}\left[\mathbf{B}_1 w_1 + \mathbf{B}_2 w_2\right] + \mathbf{D}. \end{aligned} \tag{4}$$

*e-mail: konrad.markowski@ee.pw.edu.pl

Manuscript submitted 2017-11-29, revised 2018-01-27, initially accepted for publication 2018-01-29, published in October 2018.

In the paper we can assume without loss of generality that the $\mathbf{D} = 0$, as symbolic form of the transfer matrix will be the same, as in the model matrix $\mathbf{D}$ influences only numeric values of the terms.

Polynomial. Let $\mathbb{F}$ be a field e.g., of the real number $\mathbb{R}$. The function $P(w_1, w_2)$ of the variable w_1, w_2, is called polynomial:

$$p(w_1, w_2) = \sum_{i_1=0}^{n_1} \sum_{i_2=0}^{n_2} a_{i_1, i_2} w_1^{i_1} w_2^{i_2} \tag{5}$$

in the variables w_1, w_2, over the field $\mathbb{F}$, where $a_{i_1,i_2} \in \mathbb{F}$ are called the coefficients of the polynomial.

The set of polynomial (5) over the field $\mathbb{F}$ will be denoted by $\mathbb{F}[w_1, w_2]$.

If $a_{n_1,n_2} \neq 0$, then the non-negative integer $n = n_1 + n_2$ is called the degree of a polynomial and is denoted $deg\, p(w_1, w_2)$, ie., $n = deg\, p(w_1, w_2)$. The polynomial is called monic, if $a_{n_1,n_2} = 1$ and zero polynomial, if $a_{i_1,i_2} = 0$.

Remark 1. For example, for a two-dimensional system the characteristic polynomial consists of two variables: z_1 and z_2 if we have a discrete time system; s_1 and s_2 if we have a continuous time system; z and s if we have a hybrid system.

Interested reader may find Definition and properties of the characteristic polynomial in books on linear algebra, for example in [14, ch. 9].

Directed graph. A directed graph (also called digraph) $\mathcal{D}$ consists of a non-empty finite set $\mathbb{V}(\mathcal{D})$ of elements called vertices and a finite set $\mathbb{A}(\mathcal{D})$ of ordered pairs of distinct vertices called arcs. We call $\mathbb{V}(\mathcal{D})$ the vertex set and $\mathbb{A}(\mathcal{D})$ the arc set of $\mathcal{D}$. We will often write $\mathcal{D} = (\mathbb{V}, \mathbb{A})$ which means that $\mathbb{V}$ and $\mathbb{A}$ are the vertex set and arc set of $\mathcal{D}$, respectively. The order of $\mathcal{D}$ is the number of vertices in $\mathcal{D}$. The size of $\mathcal{D}$ is the number of arc in $\mathcal{D}$. For an arc (v_1, v_2), the first vertex v_1 is its tail and the second vertex v_2 is its head. More information about the digraph theory is given in [15–18].

A two-dimensional digraph $\mathcal{D}^{(2)}$ is a directed graph with two types of arcs. For the first time, this type of digraph was presented in papers [19] and [20]. There exists $\mathcal{A}_1$–arc (or $\mathcal{A}_2$–arc) from vertex v_j to vertex v_i if and only if the (i, j)–th entry of the matrix $\mathbf{A}_1$ (or $\mathbf{A}_2$) is non-zero. There exist $\mathcal{B}_1$–arc (or $\mathcal{B}_2$–arc) from source s_m to vertex v_j if and only if the (i, m)–th entry of the matrix $\mathbf{B}_1$ (or $\mathbf{B}_2$) is non-zero.

For the system described by the matrices

$$\mathbf{A}_1 = \begin{array}{c} v_j \backslash v_i \\ v_1 \\ v_2 \\ v_3 \end{array} \begin{array}{c} \begin{array}{ccc} v_1 & v_2 & v_3 \end{array} \\ \begin{bmatrix} 0 & \mathbf{2} & 0 \\ \mathbf{3} & 0 & 0 \\ 0 & \mathbf{1} & 0 \end{bmatrix} \end{array}, \quad \mathbf{A}_2 = \begin{array}{c} v_j \backslash v_i \\ v_1 \\ v_2 \\ v_3 \end{array} \begin{array}{c} \begin{array}{ccc} v_1 & v_2 & v_3 \end{array} \\ \begin{bmatrix} 0 & 0 & \mathbf{4} \\ 0 & 0 & \mathbf{2} \\ 0 & 0 & 0 \end{bmatrix} \end{array} \tag{6}$$

we can draw two-dimensional digraph $\mathcal{D}^{(2)}$ presented in Fig. 1 consisting from vertices v_1, v_2 and v_3.

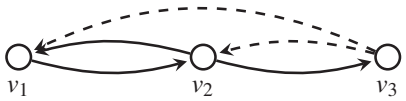

Fig. 1. Digraph $\mathcal{D}^{(2)}$ corresponding to (6)

2. Problem statement

State-of-the-art. The transfer matrix of the multiple-input multiple-output (MIMO) model described by the equation (3) has the form of (4). For single-input single-output (SISO) model, equation (4) takes the following form of transfer function:

$$T(w_1, w_2) = \frac{n(w_1, w_2)}{d(w_1, w_2)} = \frac{\sum\limits_{l=0}^{n} \sum\limits_{\substack{k=0 \\ n \leqslant l+k \leqslant 2n}}^{n} b_{l,k} w_1^l w_2^k}{1 + \sum\limits_{l=0}^{n} \sum\limits_{\substack{k=0 \\ 0 < l+k \leqslant n}}^{n} d_{l,k} w_1^l w_2^k} \tag{7}$$

where

$$\begin{aligned} d(w_1, w_2) &= 1 + \sum_{\substack{l=0 \\ 0<l+k\leqslant n}}^{n} \sum_{k=0}^{n} d_{l,k} w_1^l w_2^k = 1 + d_{n,0} w_1^n + \\ &+ d_{0,n} w_2^n + d_{n-1,1} w_1^{n-1} w_2 + d_{1,n-1} w_1 w_2^{n-1} + \ldots \\ &\ldots + d_{1,0} w_1 + d_{0,1} w_2 \end{aligned} \tag{8}$$

is a characteristic polynomial.

Matrices (2) are called a realization of a given transfer function (7), if they satisfy the (4). A realization is called minimal, if the dimension of a state matrices are minimal among all realizations of $T(w_1, w_2)$.

Remark 2. Proposed algorithm works for 2-D dynamic systems, including positive dynamic systems (if the system (3) satisfies Definition 1 and Theorem 1). Matrices (9) satisfying (4) are called a positive realization of a given transfer function (7).

Definition 1. The model (3) is called (an internally) positive if for all boundary conditions $x(i, 0) \in \mathbb{R}_+^n$ for $i \in \mathbb{Z}_+$ and $x(0, j) \in \mathbb{R}_+^n$ for $j \in \mathbb{Z}_+$ and every sequence of inputs $u(i, j) \in \mathbb{R}_+^m$ we have $x(i, j) \in \mathbb{R}_+^n$ and $y(i, j) \in \mathbb{R}_+^p$ for $i, j \in \mathbb{Z}_+$.

Theorem 1. The two-dimensional IIFM model described by the equation (3) is internally positive if and only if

$$\begin{gathered} \mathbf{A}_k \in \mathbb{R}_+^{n \times n}, \quad \mathbf{B}_k \in \mathbb{R}_+^{n \times m}, \quad k = 1, 2 \\ \mathbf{C} \in \mathbb{R}_+^{p \times n}, \quad \mathbf{D} \in \mathbb{R}_+^{p \times m}. \end{gathered} \tag{9}$$

The proof for Theorem 1 can be found in [2].

Currently, entries of the state matrices are determined using canonical forms [3, 5, 21]. Such methods obtain one of the possible state matrix realizations of the characteristic polynomial. State-of-the-art algorithms proposed in [22–27] are compared to the algorithm presented in this paper in the latter part of the article (Section 4).

Problem formulation. For IIFM SISO model determine a set of possible realizations that should be minimal among all possible. To determine a set of realizations two-dimensional $\mathcal{D}^{(2)}$ digraph theory will be used to find sets of possible state matrices $\mathbf{A}_1$ and $\mathbf{A}_2$.

There are no known methods of finding a set of all possible realizations for a given characteristic polynomial, due to the complexity of the problem that is assumed to be NP-hard [17, 28]. As canonical forms are unable to give more then a few of possible solutions, a parallel graph-based method was first proposed in [29] and [30], but extensive testing showed that it was not feasible for practical implementation as the problem of finding all possible realizations of a given polynomial is of such complexity that it cannot be solved in reasonable time even by the brute-force GPGPU (general-purpose computing on graphics processor units) method [31].

Due to such problems, in this paper improved version of the algorithm is presented. Restrictions on the search space introduced and redefined memory allocation makes the process of finding all the possible sets of minimal solutions belonging to given class possible in practically possible (linearithmic to factorial) time, as shown in Subsection 3.4.

3. Proposed solution

As a solution to the problem of finding a set of possible minimal solutions of given characteristic polynomial two-stage approach is proposed. In the first stage, a digraph-building algorithm is used. The algorithm starts with creating all possible digraph representations for all binomials in the characteristic polynomial (as explained in Subsection 3.1), then joins them by the use of disjoint union to create all possible variants of digraphs representing given polynomial realization. The algorithm uses growth and prune steps to eliminate redundant solutions before the main computational step. The result takes the form of state matrices $\mathbf{A}_1$ and $\mathbf{A}_2$. The second stage of the solution is realized with the use of MATLAB Symbolic Toolbox to determine matrices $\mathbf{B}_1$, $\mathbf{B}_2$ and $\mathbf{C}$.

Three classes of digraph structures corresponding to the characteristic polynomial have been introduced in [32]. The algorithm presented in this article works for structures belonging to class $\mathcal{K}_1$, determined by satisfying the conditions S_{1a} and S_{1b} presented in [32, pp.5] and re-introduced for reader's convenience below. Such structures are the only ones that, at the moment, allow for computation in feasible time.

Class $\mathcal{K}_1$: Digraph structures belonging to class $\mathcal{K}_1$ satisfy all characteristic polynomials of given type (with the same number and power of terms) for any $d_{l,k} \neq 0$ wages in (8). Those are digraph structures that are the most thoroughly examined in papers [28, 31] and that can be computed quickly using digraph-based GPGPU methods, as there is no need for solving a system of polynomial equations [32]. Digraph belongs to class $\mathcal{K}_1$ if the following conditions are satisfied:

$$(S_{1a}): \mathbb{V}_1\left(\mathcal{D}_1^{(2)}\right) \cap \mathbb{V}_2\left(\mathcal{D}_2^{(2)}\right) \cap \cdots \cap \mathbb{V}_j\left(\mathcal{D}_j^{(2)}\right) \neq \{\emptyset\};$$

(S_{1b}): *the number of cycles in digraph* $\mathcal{D}^{(2)}$ *equals* j;

where j is the number of binomials in the characteristic polynomial and $\mathbb{V}_h\left(\mathcal{D}_h^{(2)}\right)$ is a set of vertices of digraph $\mathcal{D}_h^{(2)}$ of h-th binomial, $h = 1, \ldots, j$.

3.1. Stage 1. In the first step, we decompose monic polynomial (8) into a set of j binomials. The factor of 1 is a special case, as it is used in the topology to represent digraph vertices, so polynomial (8) can be represented as

$$\begin{aligned} d(w_1, w_2) = {} & (1 + d_{n,0} w_1^n) \cup (1 + d_{0,n} w_2^n) \cup \\ & \cup (1 + d_{n-1,1} w_1^{n-1} w_2) \cup \\ & \cup \left(1 + d_{1,n-1} w_1 w_2^{n-1}\right) \cup \ldots \cup \\ & \cup (1 + d_{1,0} w_1) \cup (1 + d_{0,1} w_2) = \\ & = B_{n,0} \cup B_{0,n} \cup B_{n-1,1} \cup B_{1,n-1} \cup \ldots \cup \\ & \cup B_{1,0} \cup B_{0,1}, \end{aligned} \tag{10}$$

where $\cup$ is digraphs operation on vertices called composition relative to vertices. In details this kind of relation is presented in paper [33]. For each of binomials $B_{l,k}$, where $l = 1, \ldots, n$, $k = 1, \ldots, n$, $0 < l + k \leqslant n$ we create digraph representation, that consists of $k + l$ vertices and one $k + l$-arc cycle, where $k + l$ is a sum of powers of all variables of the binomial.

Example 1. Let us take polynomial $d_1(w_1, w_2) = 1 + w_1^2 w_2 + w_1, w_2$. To create digraph representation for it we need first to create digraph representation for two binomials: $B_{2,1} = 1 + w_1^2 w_2$ presented in Fig. 2a and $B_{1,1} = 1 + w_1, w_2$ in Fig. 2b.

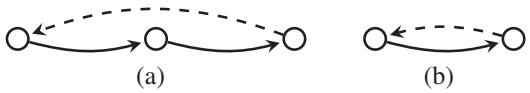

Fig. 2. (a) Digraph $\mathcal{D}_1^{(2)}$ corresponding to binomial $1 + w_1^2 w_2$; (b) Digraph $\mathcal{D}_2^{(2)}$ corresponding to binomial $1 + w_1, w_2$

After creation of all digraph representations of binomials in the polynomial, we can determine all possible characteristic polynomial realizations using all combinations of the digraph binomial representations. Finally, we combine by disjoint union received binomial digraphs into a set of digraphs, representing each possible combination of joining binomial digraph representations, each of which is corresponding to the characteristic polynomial (8).

Example 2. To achieve digraph representation for our polynomial $d_1(w_1, w_2)$ from Example 1 we need to add digraph representations of binomials. There are three possible realizations of polynomial d_1 by adding binomial $B_{1,1}$ to vertices: v_1 and v_2 (Fig. 3a); v_2 and v_3 (Fig. 3b); v_1 and v_3 (Fig. 3c).

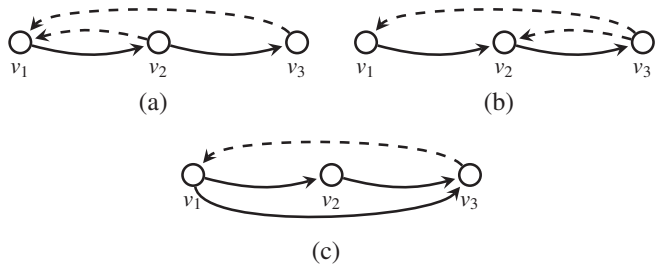

Fig. 3. Sample polynomial digraphs corresponding to union of digraphs $\mathcal{D}_1^{(2)} \cup \mathcal{D}_2^{(2)}$: (a) $\mathcal{D}_3^{(2)}$; (b) $\mathcal{D}_4^{(2)}$; (c) $\mathcal{D}_5^{(2)}$

As can be seen in example above we do not need to put digraph $\mathcal{D}_1^{(2)}$ on other vertices as its different placement will be synonymous to rotation.

To determine a set of possible solutions in finite time proper allocation of memory for parallel computation in advance is necessary. Determination of exact number of solutions that will be created after the prune step of the algorithm, as it allows us to prepare a number of kernels for each binomial in advance and it is essential for full parallelism of the algorithm. We can determine the number of possible distinct digraph solutions for given binomial using the formula:

$$p = \binom{m-1}{x_1, x_2, \cdots, x_c} = \frac{(m-1)!}{x_1! x_2! \cdots x_c!}, \tag{11}$$

where x_1 represents number of occurrences of i-th variable in the binomial and $x_1 + x_2 + \cdots + x_c = m$.

Remark 3. As stated in [32] not all digraph structures representing characteristic polynomial (8) created from binomial sub-graphs can be easily obtained. For some of them, it is impossible to obtain state matrices that are positive at all, while others generate solutions for which it is needed to get the coefficients of state matrices by solving a system of polynomial equations that, in some cases, can be under-determined. Those solutions are always outside of the class $\mathcal{K}_1$, which constitutes the reason for checking conditions S_{1a} and S_{1b}.

Remark 4. The solutions obtained in this stage tend to be minimal in size of state matrices, which is n (the order of the characteristic polynomial), as in the proof of Cayley-Hamilton theorem.

3.2. Stage 2. In the second step, we must find matrices $\mathbf{B}_1$, $\mathbf{B}_2$ and $\mathbf{C}$. Let us assume thet the matrices have the following form:

$$\mathbf{B}_1 = \begin{bmatrix} b_1^1 \\ b_2^1 \\ \vdots \\ b_1^1 \end{bmatrix}, \quad \mathbf{B}_2 = \begin{bmatrix} b_1^2 \\ b_2^2 \\ \vdots \\ b_1^2 \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} c_1 \, c_2 \cdots c_n \end{bmatrix}. \tag{12}$$

After inserting matrices (12) and state matrices $\mathbf{A}_1$ and $\mathbf{A}_2$ determined in the first stage to the (4) we obtain polynomial $\tilde{n}(w_1, w_2)$. After comparing variables with the same powers of the

$$\begin{aligned} n(w_1, w_2) = \sum_{\substack{l=0 \\ n \leqslant l+k \leqslant 2n}}^{n} \sum_{k=0}^{n} n_{l,k} w_1^l w_2^k = d_{n,n} w_1^n w_2^n + \\ + d_{n-1,n} w_1^{n-1} w_2^n + d_{n,n-1} w_1^n w_2^{n-1} + \ldots \\ \ldots + d_{n,0} w_1^n + d_{0,n} w_2^n \end{aligned} \tag{13}$$

with the polynomial $\tilde{n}(w_1, w_2)$ we obtain a non-linear set of the equations. After solving them, we obtain entries of the matrices (12).

That part of the algorithm is realised by the MATLAB function *realisationBC* created by the authors and freely available at author's website[1].

3.3. Algorithm overview. For clarity, algorithm presented in the paper is shown below in form of pseudo-code. First stage of the algorithm is presented in details in [34], while the second stage is documented in help file explaining the workings of the function (available with the function). Interested reader, wanting to see how the implemented computer version of the algorithm works, should see [34], where all functions that algorithm consists of are presented, alongside with explanation of how they work, what values are assigned to variables and execution method. Also, details about kernel allocation are presented and more explanation of how the algorithm complexity, presented in 3.4, was calculated.

3.4. Algorithm complexity. The complexity of the algorithm presented tends to be high and can be assumed to be NP-hard, as digraph realization for the characteristic polynomial includes operations that are considered NP-complete or NP-hard problems [28, 35]. The computational complexity of the presented computer algorithm can be estimated as

$$\big(V-1\big)\big(c^2 + s\log s + V\big), \tag{14}$$

for the growth/prune part, is factorial and can be presented as $\mathbf{T}(\mathbf{V}) = \mathbf{O}(V!)$ in big O notation. For the digraph creation part, the computational complexity of the algorithm run parallel can be presented as

$$\big(2n+1\big)V^2 + nV\log V + V\log V^n, \tag{15}$$

which makes the computational complexity of the part $\mathbf{T}(\mathbf{V}) = \mathbf{O}(V\log V^n)$ and makes the part of algorithm solvable

[1] https://cloud.ee.pw.edu.pl/nextcloud/index.php/s/DsQd6mEMoQK7Aza

in linearithmic time, if there are enough kernels available. The complexity of Stage 1 of the algorithm is presented in more detail in [28, 34, 35]

Algorithm 1. *DetermineState Matrix*()

1: Determine number of *binomials* in characteristic polynomial;
2: **for** *binomial* = 1 **to** *binomials* **do**
3: Determine all possible digraph $\mathcal{D}^{(2)}$ realization for each binomial B (growth step);
4: Remove redundant solutions by quick comparison of values of hash function (prune step);
5: **end for**
6: Determine number of possible digraph *solutions* (to allocate memory for kernels)
7: Determine all possible polynomial digraph realizations as a combinations of the digraph binomial representations;
8: **for** *kernel* = 1 **to** *solutions* **do**
9: Create digraph realization by means of disjoint union of assigned sub-digraph combinations;
10: Fill $\mathbf{A}_1$, $\mathbf{A}_2$ matrices based on arcs of created digraph;
11: Check the positivity of $\mathbf{A}_1$, $\mathbf{A}_2$ matrices;
12: Determine intersection set of all sub-digraphs;
13: Count number of cycles in obtained digraph and compare to number of *binomials*;
14: Check $(\mathbf{I} - \mathbf{A}_1 w_1 - \mathbf{A}_2 w_2)$;
15: Perform synchronization between device (GPU) and host (CPU);
16: **end for**
17: **for** *kernel* = 1 **to** *solutions* **do**
18: Obtain $\mathbf{B}_1$, $\mathbf{B}_2$, $\mathbf{C}$ matrices using the MATLAB function;
19: **end for**

The complexity of the second stage of the algorithm, performed with the use of MATLAB function, is hard to estimate as for fast and efficient running MATLAB uses build-in MEX functions. Tracing how they work in detail, even with the use of debuggers, is almost impossible for complicated problems. The idea of algorithm's work is presented in help file attached to the function and the function's code can be downloaded and checked, but the actual running of the MATLAB function is hard to estimate even for function's authors. Due to that next step in optimization of algorithm is moving the last stage of the algorithm from MATLAB MuPAD environment into a CUDA C code run parallel on GPU device, both due to improved speed of such solution and the ability to analyze the real computational complexity of the function.

4. Comparison and Discussion

We compared the effectiveness of the proposed method with the existing ones in Subsection 4.2 on all examples used in papers showing those methods and presented the workings in-detail on two examples presented in Subsection 4.1.

4.1. Examples

Example 3. Consider the following two-dimensional transfer function used in papers [23] and [24]:

$$G(z_1, z_2) = \frac{\alpha_1 z_1 + \alpha_2 z_2}{1 - \beta_1 z_1 - \beta_2 z_2}. \tag{16}$$

Using algorithm proposed in this paper we can write the following state matrices which satisfy the characteristic polynomial $d(z_1, z_2) = 1 - \beta_1 z_1 - \beta_2 z_2$. In the Fig. 4, we have presented its two-dimensional digraph.

$$\begin{aligned} \mathbf{A}_1 &= \left[w(v_1, v_1)_{\mathfrak{A}_1}\right] = \left[\beta_1\right], \\ \mathbf{A}_2 &= \left[w(v_1, v_1)_{\mathfrak{A}_2}\right] = \left[\beta_2\right]. \end{aligned} \tag{17}$$

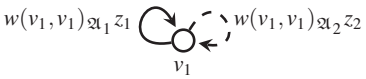

Fig. 4. Two-dimensional digraph $\mathcal{D}^{(2)}$ corresponding to (17)

Substituting state matrices (17) and matrices

$$\mathbf{B}_1 = \left[b_1^1\right], \ \mathbf{B}_2 = \left[b_1^2\right], \ \mathbf{C} = \left[c_1\right]$$

to the equation $n(z_1, z_2) = \mathbf{C}\left[\mathbf{I} - \mathbf{A}_1 z_1 - \mathbf{A}_2 z_2\right]^{-1}(\mathbf{B}_1 z_1 - \mathbf{B}_2 z_2)$, we obtain: $c_1 b_1^1 z_1 + c_1 b_1^2 z_2 = n(z_1, z_2) = \alpha_1 z_1 + \alpha_2 z_2$. After comparing the same power of the variables z_1 and z_2, we get the set of the equations: $c_1 b_1^1 = \alpha_1$, $c_1 b_1^2 = \alpha_2$. After solving them we receive the matrices

$$\mathbf{B}_1 = \left[b_1^1\right] = \left[\alpha_1 / c_1\right], \mathbf{B}_2 = \left[b_1^2\right] = \left[\alpha_2 / c_1\right], \mathbf{C}_1 = \left[c_1\right]. \tag{18}$$

which satisfy the transfer function (16).

In paper [23, pp.207] and in paper [24, pp.III-292], we can find a remark with information that the procedure proposed does not give in general minimal realization. The order of the obtained realization for our algorithm is 1, and it is minimal among all possible, while the order obtained in [23] and [24] is 2. Additionally, we have one minimal structure of the two-dimensional digraph, but entries of the matrices $\mathbf{B}_1$, $\mathbf{B}_2$ depend on the entry of the matrix $\mathbf{C}$. Thus we have an infinite number of realizations which satisfy the transfer function (16). Comparison between the algorithm proposed in this paper and algorithms proposed in papers [23] and [24] is presented in Table 2.

Example 4. Consider the following two-dimensional transfer function used in paper [26]:

$$G(z_1, z_2) = \frac{b_{10} z_1 + b_{01} z_2 + b_{11} z_1 z_2}{1 - a_{10} z_1 - a_{01} z_2 - a_{11} z_1 z_2 - a_{02} z_2^2}. \tag{19}$$

Using the algorithm proposed in this paper, we can write the following state matrices which satisfy the characteristic polynomial $d(z_1, z_2)$. In Fig. 5, we have presented one of possible two-dimensional digraph realizations.

$$\mathbf{A}_1 = \begin{bmatrix} w(v_1, v_1)_{\mathfrak{A}_1} & 0 \\ w(v_1, v_2)_{\mathfrak{A}_1} & 0 \end{bmatrix} = \begin{bmatrix} a_{10} & 0 \\ a_{11} & 0 \end{bmatrix},$$
$$\mathbf{A}_2 = \begin{bmatrix} w(v_1, v_1)_{\mathfrak{A}_2} & w(v_2, v_1)_{\mathfrak{A}_2} \\ w(v_1, v_2)_{\mathfrak{A}_1} & 0 \end{bmatrix} = \begin{bmatrix} a_{01} & 1 \\ a_{02} & 0 \end{bmatrix}. \tag{20}$$

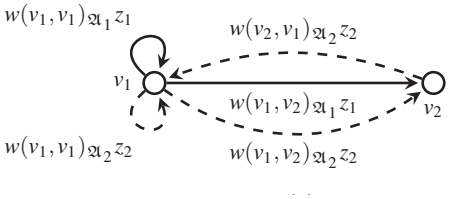

Fig. 5. Two-dimensional digraph $\mathcal{D}^{(2)}$ corresponding to (20)

Substituting state matrices (20) and matrices

$$\mathbf{B}_1 = \left[b_1^1 b_2^1\right]^T, \ \mathbf{B}_2 = \left[b_1^2 b_2^2\right]^T, \ \mathbf{C} = \left[c_1 c_2\right]$$

to the equation $n(z_1, z_2) = \mathbf{C}\left[\mathbf{I} - \mathbf{A}_1 z_1 - \mathbf{A}_2 z_2\right]^{-1}(\mathbf{B}_1 z_1 + \mathbf{B}_2 z_2)$, we obtain:

$$\begin{aligned} &\left(b_1^1 c_1 + b_1^2 c_2\right) z_1 + \left(b_2^1 c_1 + b_2^2 c_2\right) z_2 + \left(a_{10} b_1^2 c_2 + \right. \\ &+ a_{11} b_1^1 c_2) z_1^2 + \left(b_2^2 c_1 - a_{10} b_2^2 c_2 + a_{02} b_2^1 c_2\right) z_2^2 + \\ &+ \left(b_1^2 c_1 - a_{01} b_1^2 c_2 + a_{02} b_1^1 c_2 - a_{10} b_2^2 c_2 + a_{11} b_2^1 c_2\right) z_1 z_2 = \\ &= n(z_1, z_2) = b_{10} z_1 + b_{01} z_2 + b_{11} z_1 z_2. \end{aligned}$$

After comparing the same power of the variables z_1 and z_2, we get the set of equations:

$$\begin{cases} b_1^1 c_1 + b_1^2 c_2 & = b_{10} \\ b_2^1 c_1 + b_2^2 c_2 & = b_{01} \\ a_{10} b_1^2 c_2 + a_{11} b_1^1 c_2 & = 0 \\ b_2^2 c_1 - a_{10} b_2^2 c_2 + a_{02} b_2^1 c_2 & = 0 \\ b_1^2 c_1 - a_{01} b_1^2 c_2 + a_{02} b_1^1 c_2 - a_{10} b_2^2 c_2 + a_{11} b_2^1 c_2 & = b_{11}. \end{cases}$$

After solving them, we receive the matrices

$$\mathbf{B}_1 = \begin{bmatrix} b_1^1 \\ b_2^1 \end{bmatrix} = \begin{bmatrix} b_{10}/c_1 \\ b_{11}/c_1 \end{bmatrix}, \ \mathbf{B}_2 = \begin{bmatrix} b_1^2 \\ b_2^2 \end{bmatrix} = \begin{bmatrix} b_{01}/c_1 \\ 0 \end{bmatrix}, \tag{21}$$
$$\mathbf{C} = \begin{bmatrix} c_1 & c_2 \end{bmatrix} = \begin{bmatrix} c_1 & 0 \end{bmatrix}.$$

In paper [26, pp. 634–635], we can find consideration how to determine one realization using the procedure proposed by the author. The presented procedure does not give a minimal realization. The order of realization obtained using the method presented in this paper is 2, and is minimal among all possible realizations, while the order obtained in [26] is 4. Additionally, we have not one but a few minimal structures of the two-dimensional digraph (see Table 1) corresponding to the charac-

Table 1
Potential and real number of the digraph structures

Characteristic Polynomial	Grow	Prune	Solutions	
			Potential	Real
Second Fornasini-Marchesini model				
$1 + a_{20} z_2^2 + a_{11} z_1 z_2 + a_{10} z_1 + a_{01} z_2$ [a]	1	1	4	2
$1 - a_5 z_1^2 z_2 - a_4 z_2^2 - a_3 z_1^2 - a_2 z_2 - a_1 z_1$ [b]	1	1	81	2
$1 + d_{21} z_1^2 z_2 + d_{30} z_1^3 + d_{11} z_1 z_2 + d_{10} z_1 + d_{01} z_2$ [c]	1	1	54	4
$1 - \beta_1 z_1 - \beta_2 z_2$ [d][h]	1	1	1	1
$1 + a_{12} z_1 z_2^2 + a_{11} z_1 z_2 + a_{01} z_2$ [e]	1	1	18	6
$1 + z_1^2 z_2^2 + z_1 z_2$ [f]	3	2	24	8
$1 - d_{22} z_1^2 z_2^2 - d_{12} z_1 z_2^2 - d_{21} z_1^2 z_2 - d_{11} z_1 z_2 - d_{10} z_1 - d_{01} z_2$ [g]	3	2	129 024	84
General Model				
$1 + z_1^2 z_2^2 + z_1 z_2$ [f]	1	1	3	2
$1 - d_{22} z_1^2 z_2^2 - d_{12} z_1 z_2^2 - d_{21} z_1^2 z_2 - d_{11} z_1 z_2 - d_{10} z_1 - d_{01} z_2$ [g]	1	1	8	1

Examples from papers: [a][26, pp. 634–635]. [b][25, pp. 1463]. [c][25, pp. 1461]. [d][24, pp. III–292]. [e][26, pp. 637]. [f][22, pp. 636]. [g][27, pp. 6–7]. [h][23, pp. 207].

teristic polynomial $n(z_1, z_2)$, and all structure entries of the matrices $\mathbf{B}_1$ and $\mathbf{B}_2$ depend on the entry of the matrix $\mathbf{C}$. Thus we have an infinite number of realizations which satisfy the transfer function (20). A comparison between the algorithm proposed in this paper and the algorithm proposed in paper [26] is presented in Table 2.

Remark 5. Using algorithm presented in this paper, we can obtain another digraph structure presented in Fig. 6. Using the digraph structure presented in Fig. 6 we can write the matrices

$$\mathbf{A}_1 = \begin{bmatrix} 0 & 0 \\ w(v_1, v_2)_{\mathfrak{A}_1} & w(v_2, v_2)_{\mathfrak{A}_1} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ a_{11} & a_{10} \end{bmatrix},$$

$$\mathbf{A}_2 = \begin{bmatrix} 0 & w(v_2, v_1)_{\mathfrak{A}_2} \\ w(v_1, v_2)_{\mathfrak{A}_2} & w(v_2, v_2)_{\mathfrak{A}_2} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ a_{20} & a_{01} \end{bmatrix}, \tag{22}$$

$$\mathbf{B}_1 = \begin{bmatrix} b_1^1 \\ b_2^1 \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{a_{20}b_{10}}{a_{01}c_1} \end{bmatrix}, \quad \mathbf{B}_2 = \begin{bmatrix} b_1^2 \\ b_2^2 \end{bmatrix} = \begin{bmatrix} \frac{b_{01}}{c_1} \\ 0 \end{bmatrix},$$

$$\mathbf{C} = \begin{bmatrix} c_1 & c_2 \end{bmatrix} = \begin{bmatrix} c_1 \frac{a_{01}}{a_{20}} c_1 \end{bmatrix}.$$

The coefficient must additionally satisfy the following condition,

$$\frac{a_{20}b_{10}}{a_{10}} + \frac{a_{01}a_{11}b_{01}}{a_{20}} - a_{10}b_{01} = b_{11}$$

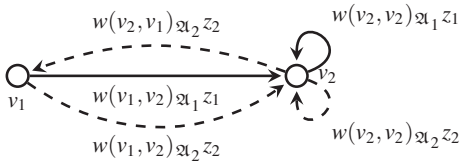

Fig. 6. Two-dimensional digraph $\mathcal{D}^{(2)}$ corresponding to (20)

to satisfy the transfer function (19). The obtained set of realizations satisfies the minimality condition, and we can see that it is minimal among all possible realizations.

Remark 6. It should be noted that the algorithm proposed in this paper gives 4 potential realizations, but of those only 2 realizations, presented in Fig. 5 and Fig. 6, can be achieved (see the first row in Table 1). Furthermore, it should be remembered that we can obtain at least two additional solutions by the re-enumeration of the vertices in the digraphs, which is equivalent to the transposition of the state matrices.

4.2. Comparison with state-of-the-art algorithms. In Table 1, we presented a potential and real number of digraph structures for examples considered in the state-of-the-art papers: [22–27]. Potential solutions hold the solutions that can be created by the algorithm, while real solutions hold only solutions that belong to $\mathcal{K}_1$ class. It should be noted that the potential and real solutions do not contain realizations that we can receive by re-encumbering the vertices in the digraphs. This operation is very simple, but it is not included in the algorithm, as it leads to unnecessary operations and can be obtained by transposition of **A** matrices after the algorithm is finished.

Table 2
Comparison the size of the realization and the number of the solutions

Characteristic Polynomial	Size ($n \times n$)	Solution (number)	Proposed algorithm	
			Size	Solution
Second Fornasini-Marchesini model				
$1 + a_{02}z_2^2 + a_{11}z_1z_2 + a_{10}z_1 + a_{01}z_2$ [a]	4	1	2	4
$1 - a_5z_1^2z_2 - a_4z_2^2 - a_3z_1^2 - a_2z_2 - a_1z_1$ [b]	4	1	3	2
$1 + d_{21}z_1^2z_2 + d_{30}z_1^3 + d_{11}z_1z_2 + d_{10}z_1 + d_{01}z_2$ [c]	3	1	3	12
$1 - \beta_1z_1 - \beta_2z_2$ [d][h]	2	1	1	1
$1 + a_{12}z_1z_2^2 + a_{11}z_1z_2 + a_{01}z_2$ [e]	3	1	3	6
$1 + z_1^2z_2^2 + z_1z_2$ [f]	–	1	4	8
$1 - d_{22}z_1^2z_2^2 - d_{12}z_1z_2^2 - d_{21}z_1^2z_2 - d_{11}z_1z_2 - d_{10}z_1 - d_{01}z_2$ [g]	–	1	4	84
General Model				
$1 + z_1^2z_2^2 + z_1z_2$ [f]	2	1	2	2
$1 - d_{22}z_1^2z_2^2 - d_{12}z_1z_2^2 - d_{21}z_1^2z_2 - d_{11}z_1z_2 - d_{10}z_1 - d_{01}z_2$ [g]	2	1	2	2

Examples from papers: [a][26, pp. 634–635]. [b][25, pp. 1463]. [c][25, pp. 1461]. [d][24, pp. III–292]. [e][26, pp. 637]. [f][22, pp. 636]. [g][27, pp. 6–7]. [h][23, pp. 207].

In Table 2, we presented the comparison of the size of the realization and the number of the solutions for the examples presented in the papers considered in Table 1. After looking at the table, we can say that the algorithm proposed in this paper gives more realizations and additionally all of them are minimal among all possible and thus can be proven as superior on grounds of achieved solutions to other state-of-the-art methods.

In case of computational complexity, compared algorithms tend to be proposed in form unsuitable for direct computer implementation, not optimized in any way and presented in a way that makes estimation hard (text or incomplete set of steps instead of pseudo-code). Moreover, as they were not intended for computer implementation, their complexity is never mentioned. Authors tried to the best of their abilities to evaluate the complexity and fastest of the compared algorithms have quadratic complexity of $\mathbf{T}(\mathbf{V}) = \mathbf{O}(n^2)$ [24, 25], at least over-quadratic complexity $\mathbf{T}(\mathbf{V}) = \mathbf{O}(n^{2.373})$ [22, 23, 27] or cubic complexity $\mathbf{T}(\mathbf{V}) = \mathbf{O}(n^3)$ [26]. In addition all of the algorithms construct sparse matrices of large size and perform large number of additional operations, which for $n \leq 5$ exceed the number of operations resulting from complexity estimation. Despite being able to obtain only single, often non-minimal solution, the complexity of the compared algorithms is so high that their direct computer implementation would be working slower then second and subsequent GPGPU executions of proposed algorithm, generating a set of minimal solutions.

Remark 7. In the paper only some of the experimental results obtained were presented in Tables 1 and 2. Interested reader can get more experimental results from dataset presented on the author's webpage[2].

5. Concluding remarks

This paper presents a method for finding a set of realizations of 2-D system, which is based on the multi-dimensional digraph theory. Presented method is shown to be superior to all other state-of-the-art solutions the authors know of, as demonstrated in Section 4. The method gives as a result a set of realizations of 2-D dynamic system, both in the form of digraph and matrices representations.

Presented method differs from other methods, which are capable of finding only a few of existing realizations in that it is able to find a large set of possible realizations restricted by class $\mathcal{K}_1$. Moreover, the obtained solutions that are always minimal in terms of state matrices size and as presented in Table 2 of the same or smaller size then obtained by other methods.

Many new realizations in form of [**A**, **B**, **C**, **D**] matrices obtained using both the proposed method and attached MATLAB functions allow for more detailed examination of 2-D system dynamics. Moreover it allows possibilities of experimental evaluation of many complex and still not theoretically solved problems, such as the calculation of the reachability index.

Acknowledgements. The research has been financed with the funds of the Statutory Research for 2018.

References

[1] D.G. Luenberger, *Introduction to Dynamic Systems: Theory, Models, and Applications*. New York: Wiley, 1979, ch. Positive linear systems.
[2] T. Kaczorek, *Positive 1D and 2D systems*. London: Springer Verlag, 2001.
[3] L. Benvenuti and L. Farina, "A tutorial on the positive realization problem," *IEEE Transactions on Automatic Control*, vol. 49, no. 5, pp. 651–664, 2004.
[4] L. Ntogramatzidis, M. Cantoni, and R. Yang, "On the partial realization of noncausal 2-D linear systems," *IEEE Transactions of Circuits and Systems*, vol. 54, pp. 1800–1808, 2007.
[5] T. Kaczorek, "Positive realization for 2D systems with delays," in *Proceedings of 2007 International Workshop on Multidimensional (nD) Systems*. IEEE, 2007, pp. 137–141.
[6] L. Xu, H. Fan, Z. Lin, and N. Bose, "A direct-construction approach to multidimensional realization and LFR uncertainty modeling," *Multidimensional Systems and Signal Processing*, vol. 19, no. 3–4, pp. 323–359, 2008.
[7] T. Kaczorek and L. Sajewski, *The Realization Problem for Positive and Fractional Systems*. Berlin: Springer International Publishing, 2014.
[8] C.M. Ionescu, D. Copot, A. Maxim, E. Dulf, R. Both, and R.D. Keyser, "Robust autotuning mpc for a class of process control applications," in *2016 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR)*, 2016, pp. 1–6.
[9] J. DiStefano III, *Dynamic Systems Biology Modeling and Simulation*. Academic Press, 2013.
[10] M. Sebek and Z. Hurak, "2-D polynomial approach to control of leader following vehicular platoons," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 6017–6022, 2011.
[11] A. Berman, M. Neumann, and R.J. Stern, *Nonnegative Matrices in Dynamic Systems*. New York: Wiley, 1989.
[12] R.A. Horn and C.R. Johnson, *Topics in Matrix Analysis*. Cambridge Univ. Press, 1991.
[13] E. Fornasini and G. Marchesini, "State-space realization theory of two-dimensional filters," *IEEE Trans, Autom. Contr.*, vol. 21, pp. 481–491, 1976.
[14] T. Blyth and E. Robertson, *Basic Linear Algebra (2nd Edition)*. London: Springer, 2002.
[15] L.R. Foulds, *Graph Theory Applications*. Springer Verlag, 1991.
[16] W.D. Wallis, *A Beginner's Guide to Graph Theory*. Biiokhäuser, 2007.
[17] J. Bang-Jensen and G. Gutin, *Digraphs: Theory, Algorithms and Applications (2nd Edition)*. London: Springer-Verlag, 2009.
[18] C. Godsil and G. Royle, *Algebraic Graph Theory*. Springer Verlag, 2001.
[19] E. Fornasini and M.E. Valcher, "Directed graphs, 2D state models, and characteristic polynomials of irreducible matrix pairs," *Linear Algebra and Its Applications*, vol. 263, pp. 275–310, 1997.
[20] "On the positive reachability of 2D positive systems," *LCNIS*, pp. 297–304, 2003.

[2] https://cloud.ee.pw.edu.pl/nextcloud/index.php/s/ssq7ddd7ppTJDnq

[21] K.A. Markowski, "Determination of positive realization of two dimensional systems using digraph theory and GPU computing method," in *International Symposiumon Theoretical Electrical Engineering, 24th–26th June 2013: Pilsen, Czech Republic*, 2013, pp. II7–II8.

[22] T. Kaczorek, "Realization problem for general model of twodimensional linear systems," *Bull. Pol. Ac.: Tech.*, vol. 35, no. 11–12, pp. 633–637, 1987.

[23] M. Bisiacco, E. Fornasini, and G. Marchesini, "Dynamic regulation of 2D systems: A state-space approach," *Linear Algebra and Its Applications*, vol. 122–124, pp. 195–218, 1989.

[24] L. Xu, L. Wu, Q. Wu, Z. Lin, and Y. Xiao, "Reduced-order realization of Fornasini-Marchesini model for 2D systems," in Circuits and Systems, 2004. ISCAS '04. *Proceedings of the 2004 International Symposium on*, vol. 3, 2004, pp. III–289–292.

[25] L. Xu, Q. Wu, Z. Lin, Y. Xiao, and Y. Anazawa, "Futher results on realisation of 2D filters by Fornasini-Marchesini model," in *8th International Conference on Control, Automation, Robotics and Vision, Kunming, China, 6–9th December*, 2004, pp. 1460– 1464.

[26] L. Xu, L.Wu, Q.Wu, Z. Lin, and Y. Xiao, "On realization of 2D discrete systems by Fornasini-Marchesini model," International *Journal of Control, Automation, and Systems*, vol. 4, no. 3, pp. 631–639, 2005.

[27] T. Kaczorek, "Positive realization of 2D general model," *Logistyka*, vol. nr 3, pp. 1–13, 2007.

[28] K. Hryniów and K. A. Markowski, "Optimisation of digraphs-based realisations for polynomials of one and two variables," in *Progress in Automation, Robotics and Measuring Techniques*, ser. Advances in Intelligent Systems and Computing, R. Szewczyk, C. Zieliński, and M. Kaliczyńska, Eds. Springer International Publishing, 2015, vol. 350, pp. 73–83.

[29] "Parallel digraphs-building algorithm for polynomial realisations," in *Proceedings of 2014 15th International Carpathian Control Conference (ICCC)*, 2014, pp. 174–179.

[30] "Reachability index calculation by parallel digraphsbuilding," in *19th International Conference on Methods and Models in Automation and Robotics (MMAR)*, Miedzyzdroje, Poland, September 2–5, 2014, 2014, pp. 808–813.

[31] "Conditions for digraphs representation of the characteristic polynomial," in *Young Scientists Towards the Challenges of Modern Technology*, 2014, pp. 77–80.

[32] "Classes of digraph structures corresponding to characteristic polynomials," in *Challenges in Automation, Robotics and Measurement Techniques*, ser. Advances in Intelligent Systems andComputing, R. Szewczyk, C. Zieliński, and M. Kaliczyńska, Eds. Springer International Publishing, 2015, vol. 440, pp. 329–339.

[33] K.A. Markowski, "Minimal positive realizations of linear continuous-time fractional descriptor systems: Two cases of an input-output digraph structure", *International Journal of Applied Mathematics and Computer Science* 28(1), 9–24 (2018).[Online]. Available: https://content.sciendo.com/view/journals/amcs/28/1/article-p9.xml.

[34] K. Hryniów and K.A. Markowski, "Parallel digraphs-building computer algorithm for finding a set of characteristic polynomial realisations of dynamic system," *Journal of Automation, Mobile Robotics & Intelligent Systems (JAMRIS)*, vol. 10, no. 3, pp. 38–51, 2016.

[35] "Optimisation of digraphs creation for parallel algorithm for finding a complete set of solutions of characteristic polynomial," in *20th International Conference on Methods and Models in Automation and Robotics*, 2015, pp. 1139–1144.