BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES, Vol. 64, No. 4, 2016
*DOI: 10.1515/bpasts-2016-0093*

# Inversion of selected structures of block matrices of chosen mechatronic systems

T. TRAWIŃSKI[1]*, A. KOCHAN[1], P. KIELAN[1], and D. KURZYK[2]

[1]Department of Mechatronics, Silesian University of Technology, 10A Akademicka St., 44-100 Gliwice, Poland
[2]Institute of Mathematics, Silesian University of Technology, 23 Kaszubska St., 44-100 Gliwice, Poland

**Abstract.** This paper describes how to calculate the number of algebraic operations necessary to implement block matrix inversion that occurs, among others, in mathematical models of modern positioning systems of mass storage devices. The inversion method of block matrices is presented as well. The presented form of general formulas describing the calculation complexity of inverted form of block matrix were prepared for three different cases of division into internal blocks. The obtained results are compared with a standard Gaussian method and the "inv" method used in Matlab. The proposed method for matrix inversion is much more effective in comparison in standard Matlab matrix inversion "inv" function (almost two times faster) and is much less numerically complex than standard Gauss method.

**Key words:** arrowhead matrices, mechatronic systems, matrix inversion, computational complexity.

## 1. Introduction

Mathematical models of physical objects are formulated using, among others, the Lagrangian formalism. It can be represented by the following form of differential equations written in matrix form [1]:

$$D\ddot{q}+C\dot{q}+Kq+G=\tau \tag{1}$$

where $D$ denotes inertial matrix, $C$ denotes matrix of centrifugal and Coriolis forces, $K$ denotes stiffness matrix, $G$ denotes vector of gravitational forces, $\tau$ denotes vector of generalized forces, $q$ denotes vector of generalized displacements.

The inertial matrix, present in (1), in predominant cases, may be regarded as symmetrical, but for mathematical models of wide set of physical objects its elements are indirect function of time. The elements of inertial matrix may depend on angular or linear displacements. It happens very often in mathematical modelling of advanced mechanical systems such as robot manipulators [1, 23, 24]. In mathematical models of electro-mechanical systems matrix parameters of self- and mutual inductances of stator and rotor windings also depend on angular displacements of the rotor [2]. Similarly, in mathematical models of head positioning systems of modern mass storage devices (hard disk drives), elements of inertial matrices (which are represented by mass moments of inertia or by masses) depend on temporary spatial configurations of the links of its kinematic chain and therefore the angular (or linear) displacement of joints [3, 4]. A thorough analysis of the structures of the inertia matrix of mechanical systems, electromechanical systems, robot manipulators, head positioning systems of hard disk drives, and many others, shows that they have often block structure. The inertia matrices of these systems can be divided internally into elementary matrices (blocks), for example, an electromechanical system (squirrel cage induction motors) inertia matrix [5] may take the form of:

$$\mathbf{D}=\begin{bmatrix} L_s & M_{sr} & \cdots \\ M_{sr}^T & L_r & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix} \tag{2}$$

where $L_s$, $L_r$, $M_{sr}$ denote matrices of self inductances of stator windings and rotor windings, matrix of mutual stator – rotor inductances respectively.

The inertia matrices in electromechanical systems may have structural features allowing them to be divided into a number of submatrices, for the most elementary matrices – the blocks have the size of $2\times2$ [2, 5]. Symmetric inertia matrices encountered in mathematical models of head positioning systems of hard disk drives [3, 4], have very different forms, depending on structures of its kinematic chains. Exemplary forms of these matrices are as follows:

$$\mathbf{D}=\begin{bmatrix} d_{11} & 0 & 0 & 0\ldots & 0 \\ & d_{22} & d_{23} & 0\ldots & 0 \\ & & d_{33} & 0\ldots & 0 \\ & & & \ddots & d_{k-1,k} \\ sym & & & & d_{k,k} \end{bmatrix} \tag{3}$$

$$\mathbf{D}=\begin{bmatrix} d_{11} & d_{12} & d_{13} & \ldots & d_{1,k} \\ & d_{22} & 0 & \ldots & 0 \\ & & d_{33} & \ldots & 0 \\ & & & \ddots & 0 \\ sym & & & & d_{k,k} \end{bmatrix} \tag{4}$$

*e-mail: tomasz.trawinski@polsl.pl

Square matrices with entries equal zero except for their main diagonal, one row and one column have many applications e.g. in wireless communication systems [19], neural-network models [20] as well as issues related with chemistry [21] or phisics [22]. In this paper we propose a method of inversion of symetric matrices containing non-zero blocks in their main diagonal, one column, one row and zeros in remaining entries. This kind of matrices is significant in modeling of mechatronic systems. Considered structures of block matrices can be used to describe an inertia matrix in mathematical model of a head positioning system of a hard disk drive. It seems interesting how fast may a group of symmetric inertia matrices, as encountered in models of hard disk drive head positioning system, be inverted considering their block structure and their reasonable block dimension. It is also worth to investigate influence an internal structure of block matrices has on inversion time as well as on numerical complexity of inversion process. The above mentioned issues motivated the Authors to investigate the problem of numerical complexity of inversion of block matrices with different structures. Similar problem was investigated in [5], where effective method (based on $LDL^T$ decomposition) of finding the inverse of this kind of matrix was proposed. However, approximation of complexity of the algorithm was very general. We propose a detailed analysis of the number of algebraic operations necessary to implement inversion of considered block matrices.

In chapter 2 general information about the internal structure of block matrices, further considered in the article is presented.In this chapter the inverted form of block matrix, derived in former works, is presented. In former works, the authors have not investigated mutual interactions between internal structure of block matrices (consisting more than 16 elements) and its numerical complexity and resultant computation times. In paper [18] the number of algebraic operation has been calculated for an inversion process of strictly defined matrix (only one type), but under different partition of input block matrix, i.e. into 4, 6 and 16 elementary matrices. Other papers showing the relation between block matrix structure and structure of kinematic chains of robots (alternatively kinematic chains of head positioning systems) [3, 4, 6] and [18] or structure of winding of electric machines [2] and [5]. General formulas, describing a relation between block matrix's structure and dimension, and the number of algebraic operations have not been presented in authors's former papers. In chapter 3 of this paper methods of accounting the number of algebraic operations, necessary to make during inversion process are described. Three different cases of block matrices internal portioning are considered. Inversion times for all chosen structures of block matrices are compared with times of inversion using standard inversion method in Matlab ("inv" function). The presented method of block matrix inversion is much more effective than the one used in Matlab. Also, in this article the number of algebraic operations (necessary to invert the matrices) has been calculated and compared with Gaussian method of matrix inversion, and it has shown the advantages of proposed method which exhibits the smallest increase of number of algebraic operation due to block matrix dimension increase.

## 2. Inverting the block matrix using its internal structure

Suppose that the inertia matrix of a physical object can be represented in the following block form:

$$\mathbf{D} = \begin{bmatrix} a_0 & b_1 & b_2 & \dots & b_k \\ b_1^T & a_1 & 0 & \dots & 0 \\ b_2^T & 0 & a_2 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ b_k^T & 0 & 0 & 0 & a_k \end{bmatrix} \tag{5}$$

The division into elementary matrices of the above matrix is essentially arbitrary, but there may occur practical reasons [6] which define them. It happens in branched head positioning systems of hard disk drives [4] and in this case, this division is correlated with the structure of the kinematic chain of head positioning system. Due to the ability to give the physical interpretation of the matrices of the structure of an arrow (arrowhead), they became the subject of research. Arrowhead matrix representation of the inertia matrix describing the equation of physical object under consideration, may also used in the description of operation of the wireless links [9]. One of the problems associated with arrow matrices is effective determination of the eigenvalues of [7–9]. Additionally, considered are parallel matrix inversion methods as described in [10]. In [11] a quick method of solving systems of linear equations of the arrow matrix of coefficients is presented.

As shown in [6] inverted block matrix (5) can be represented as:

$$\mathbf{D_r} = \begin{bmatrix} c_0 & -c_0 b_1 a_1^{-1} & -c_0 b_2 a_2^{-1} & \dots & \xi_1 \\ & c_1 & a_1^{-1} b_1^T c_0 b_2 a_2^{-1} & \dots & \xi_2 \\ & & c_2 & \dots & \xi_3 \\ & & & \ddots & \vdots \\ sym & & & & c_k \end{bmatrix} \tag{6}$$

where
$c_0 = (a_0 - \sum_j^k b_j a_j^{-1} b_j^T)^{-1}$, $c_i = (a_i - b_i^T (c_0^{-1} + b_i a_j^{-1} b_j^T)^{-1} b_i)^{-1}$ for $i \in \{1, \dots, k\}$, $\xi_1 = -c_0 b_k a_k^{-1}$, $\xi_2 = a_1^{-1} b_1^T c_0 b_k a_k^{-1}$, $\xi_3 = a_2^{-1} b_2^T c_0 b_k a_k^{-1}$.

The inverted block matrix of inertia (6) consist of (as can be observed) elementary matrices, which are calculated on the basis of the block matrix elements (5) before the inversion. It is possible to calculate chosen elementary submatrix (6) without the need of calculation of the remaining elements of the blocks matrix.

## 3. The number of algebraic operations during the inversion of the block matrix

For the sake of consequence of further course of the discussion, definition of block dimension is formulated.

**Definition 1.** If the symmetric block matrix D has been divided into elementary matrices using $k$ vertical lines (into $k+1$ columns) and $k$ horizontal lines (into $k+1$ rows) the block size $n$ of the block matrix is defined as an ordered pair of numbers ($k+1$, $k+1$). Block size of the block matrix is written as: $n = (k+1)\times(k+1)$, or briefly by $n = (k+1)$.

In order to demonstrate effectiveness of the computing algorithm of matrix inversion, the number of algebraic operations needed to be performed will be calculated for block matrices with different internal divisions into elementary blocks – submatrices. All of the analyzed cases of the block matrix will have a structure such as matrix (5), it will differ only in size of elementary matrices. As mentioned above, each of the inverse elementary matrices (6) can be calculated individually. A detailed analysis of the structure of the matrix (6) reveals that there are four different types of items – elementary submatrices of inverted block matrix requiring the calculation. These are as follows:

1. first elementary matrix, which later will be called the leading element, has the following form:

$$c_0 = (a_0 - \sum_j^k b_j a_j^{-1} b_j^T)^{-1} \tag{7}$$

2. elementary matrices, that in physical interpretation can be responsible for negative feedback, have following forms:

$$d_i = -c_0 b_i a_i^{-1} \tag{8}$$

for $i \in \{1, \ldots, k\}$, auxiliary matrices having forms:

$$e_i = -a_i^{-1} b_i^T \tag{9}$$

for $i \in \{1, \ldots, k-1\}$.

3. elementary matrices, which in physical interpretation can be responsible for positive couplings, are forms of the products of the matrices (8) and (9)

$$e_i d_j \tag{10}$$

for $i \in \{1, \ldots, k-1\}, j \in \{1, \ldots, k\}$.

4. block matrices, which can be called in physical interpretation as self inertia matrices, have following forms:

$$c_i = (a_i - b_i^T (c_0^{-1} + b_i a_j^{-1} b_j^T)^{-1} b_i)^{-1} \tag{11}$$

for $i \in \{1, \ldots, k\}$.

By introducing the above indications of elementary matrices, inverted block matrix (6) takes the following form:

$$\mathbf{D_r} = \begin{bmatrix} c_0 & d_1 & d_2 & \ldots & d_k \\ & c_1 & e_1 d_2 & \ldots & e_1 d_k \\ & & c_2 & \ldots & e_2 d_k \\ & & & \ddots & \vdots \\ sym & & & & c_k \end{bmatrix} \tag{12}$$

Later in this article the number of algebraic operations required to implement in order to calculate the inverted block matrix (5) for three different cases of the internal structure of input matrices will be calculated.

**3.1. Case 1 – one-piece elementary matrices – a partition in the 1–1–1–1 order.** If the block matrix can be divided into one-piece elementary matrices (it will take the form of a matrix (5)), then the number of algebraic operations related to the calculation of the leading element $c_0$ (7) is determined by: block size $n$ of the block matrix, the inversion process, operations of addition (subtraction) in triples of matrices $b_j a_j^{-1} b_j^T$, multiplication in triples of matrices $b_j a_j^{-1} b_j^T$, and $a_j$ matrix inversion. By convention, such division of the block m atrix for one-piece elementary matrices will be referred as division in 1–1–1–1 order. Calculated number of algebraic operations for various dimensions of the block matrices is shown in Table 1.

Table 1
Number of algebraic operations necessary to calculate the leading element $c_0$

| Block dimension $n$ | 2 | 3 | 4 | 5 |
|---|---|---|---|---|
| Sum of algebraic operations $l_{oI}$ | 4 | 7 | 10 | 13 |

General relationship between the dimension of the block matrix block dimension $n$ and the number of algebraic operations that are needed for the calculation of the leading element $c_0$ (7), is as follows:

$$l_{oI} = 3(n-1) + 1 \tag{13}$$

The calculations effort necessary to perform the designation of negative feedback matrix $d_i$ (8), is associated with the implementation of algebraic operations necessary to: calculate the matrix product $c_0$ and $b_i$, calculate the inverse form of the matrix $a_i$ (in present case – dividing by an element of the matrix). Calculated numbers of algebraic operations for different block dimension $n$ of the block matrix, the negative feedback matrix $d_i$ (8), are shown in Table 2. It should be noted, that in-

Table 2
Number of algebraic operations necessary to calculate the negative feedback matrix $d_i$

| Block dimension $n$ | 2 | 3 | 4 | 5 |
|---|---|---|---|---|
| Sum of algebraic operations $l_{oII}$ | 3 | 6 | 9 | 12 |

creasing size of the block matrix by one, results in occurrence of an additional negative feedback matrix $d_i$ – in the first row and first column of the inverted block matrix.

General relationship between block dimension of the block matrix and the number of algebraic operations that are needed for the calculation of the negative feedback matrix $d_i$ (8) is as follows:

$$l_{oII} = 3(n-1) \tag{14}$$

for $n = 2, 3, \ldots$

The number of algebraic operations needed in order to determine the matrix of positive feedback $e_i d_j$ (10), is related with the calculation of the auxiliary matrix $e_i$, and its product with negative feedback matrix $d_j$. It should be emphasized that this type of matrices occurs for block dimension $n \geq 3$. These calculations require: inversion of the matrix $a_i$, multiplication of matrices $a_i^{-1} b_i^T$, multiplication by $(-1)$ and multiplication by matrix $d_j$. Calculated numbers of algebraic operations for various dimensions of the block matrix, for the positive feedback matrix $e_i d_j$ are shown in Table 3.

Table 3
Number of algebraic operations necessary to calculate the positive feedback matrix $e_i d_j$

| Block dimension $n$ | 3 | 4 | 5 | 6 |
|---|---|---|---|---|
| Sum of algebraic operations $l_{oIII}$ | 3 | 9 | 18 | 30 |

General relationship between block dimension of the block matrix and the number of algebraic operations that are needed for the calculation of the positive feedback matrix $e_i d_j$ (10) is as follows (assuming that matrix $d_i$ was calculated in previous step):

$$l_{oIII} = 3\frac{(n-2)(n-1)}{2} \tag{15}$$

for $n = 3, 4, \ldots$

The number of algebraic operations needed in order to determine self inertia matrices (11) is associated with following calculations: the product of matrices $b_i a_i^{-1} b_i^T$; matrix $a_i$ inversion (in this case by division by the element of this matrix); the summation in the inner brackets; inversion of the internal expression (in parentheses) and multiplying it by the matrices $b_i$ and $a_i^T$, subtraction of expressions contained in external parentheses and its inversions. Calculated numbers of algebraic operations for different block dimensions $n$ of block matrix, necessary for calculation of self inertia matrices, are shown in Table 4.

Table 4
Number of algebraic operations necessary to calculate the self inertia matrices $c_i$

| Block dimension $n$ | 2 | 3 | 4 | 5 |
|---|---|---|---|---|
| Sum of algebraic operations $l_{oIII}$ | 8 | 16 | 24 | 32 |

General relationship between block dimension of the block matrix and the number of algebraic operations that are needed for the calculation of the self inertia matrices $c_i$ (11) is as follows:

$$l_{oIV} = 8(n-1) \tag{16}$$

for $n = 1, 2, \ldots$

The total amount of algebraic operations needed to calculate the inverted form of block matrix of the block dimension $n$ and assuming that all matrices are one-piece, is given by the following formula:

$$l_o = 3\frac{n^2}{2} + 19\frac{n}{2} - 10 \tag{17}$$

for $n = 1, 2, \ldots$

Relationship showing the number of algebraic operations needed to be done to calculate the elementary matrices (with the block dimension growth of the block matrix), can be represented graphically in Fig. 1.

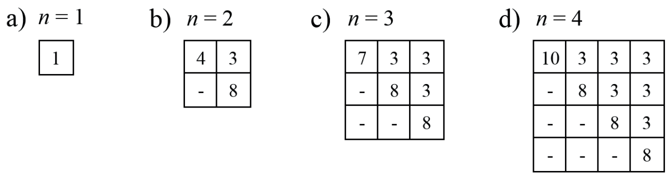


Fig. 1. The number of algebraic operations required to be implemented in order to calculate the matrix inverse: a) – for a single-element matrix (special case) the number of operations – 1, b) – for block matrices with block dimension 2 the number of operations – 15, c) – for matrices with block dimension 3 the number of operations – 32, d) – for matrices with block dimension 4 the number of operations – 52

**3.2. Case 2 – elementary matrices in partition in the 1–2–2–2 order.** Assume that the block matrix can be divided into elementary matrices, so could be present at the block main diagonal as to: one-piece matrix, $2\times 2$ dimensional square and symmetric matrices. The effect of such division of input matrix into elementary matrices, is that in the first line rectangular matrices ($1\times 2$ dimensional) and in the first column rectangular matrices (with dimension $2\times 1$) are located. Such a division of the block matrix into one-piece elementary and 2 by 2 dimensional symmetric matrices will be called the division in the 1–2–2–2 order. The block matrix 3) fulfils the above mentioned conditions, and its partition into elementary matrices in 1–2–2–2 order is shown in Fig. 2.

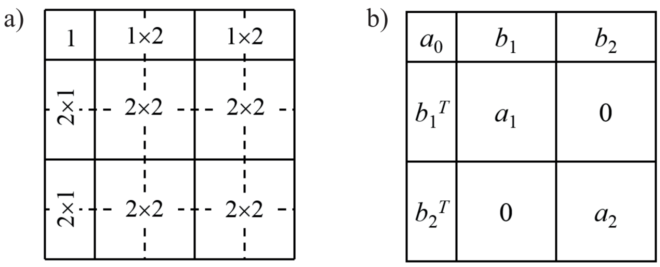


Fig. 2. Matrix partitioning into an 1–2–2–2 order: a) – block dimensions of submatrices, b) – signs assignment to the submatrices

The number of algebraic operation necessary to calculate the leading element $c_0$ (7) is determined by block dimension $n$ of input matrices, but also by dimensions of matrices in triple $b_j a_j^{-1} b_j^T$. The number of algebraic operations necessary to calculate matrices $b_j a_j^{-1} b_j^T$ results from two rectangular matrices multiplications (with dimensions $1\times2$ and $2\times1$) with square matrix (with dimension $2\times2$). Furthermore it is necessary to calculate the inverse form of 4 elementary $a_j^{-1}$ matrix. The calculation effort which should be carried out for leading elements $c_0$ calculations is summarized in Table 5.

Table 5
Number of algebraic operations necessary to calculate the leading element $c_0$

| Block dimension $n$ | 2 | 3 | 4 | 5 |
|---|---|---|---|---|
| Inversion of 4-elementary submatrices $a_j^{-1}$ | 15 | 30 | 45 | 60 |
| Multiplication of triple matrices $b_j a_j^{-1} b_j^T$ | 9 | 18 | 27 | 36 |
| Additions / Subtractions | 1 | 2 | 3 | 4 |
| Inversions / Divisions | 1 | 1 | 1 | 1 |
| Sum of algebraic operations $l_{oI}$ | 26 | 51 | 76 | 101 |

Generally, relationship between block dimension $n$ of block matrix with its partition into elementary matrices with dimensions according to 1–2–2–2 order, and numbers of algebraic operations necessary to calculate the leading elements $c_0$ (7), is as follows:

$$l_{oI} = 25(n-1)+1 \tag{18}$$

for $n = 1, 2, \ldots$

Calculated numbers of algebraic operations needed for a determination of the negative feedback matrix $d_i$ (8), in relation to different block dimensions of input matrices is presented in Table 6. It is worth to underline, that increase of block dimension by one results in appearing of additional negative feedback matrices $d_i$ ($1\times2$ and $2\times1$ dimensional) in the first row and first column of the inverted block matrix.

Table 6
Number of algebraic operations necessary to calculate the negative feedback matrix $d_i$

| Block dimension $n$ | 2 | 3 | 4 | 5 |
|---|---|---|---|---|
| Inversion of 4-elementary submatrices $a_j^{-1}$ | 15 | 30 | 45 | 60 |
| Multiplication of triple matrices $-c_0 b_i a_i^{-1}$ | 9 | 18 | 27 | 36 |
| Sum of algebraic operations $l_{oII}$ | 23 | 46 | 69 | 92 |

The overall relationship between the block dimension $n$ of a block matrix (partitioned into submatrices in the 1–2–2–2 order) and the number of algebraic operations needed to calculate the negative feedback matrix $d_i$ (8), is as follows:

$$l_{oII} = 23(n-1) \tag{19}$$

for $n = 2, 3, \ldots$

The calculated number of algebraic operations needed to obtain the positive couplings matrices $e_i d_j$ (10), upon different block dimensions of input block matrix, is shown in Table 7.

Table 7
Number of algebraic operations necessary to calculate the positive feedback matrix $e_i d_j$

| Block dimension $n$ | 3 | 4 | 5 | 6 |
|---|---|---|---|---|
| Inversion of 4-elementary submatrices $a_j^{-1}$ | 15 | 45 | 90 | 150 |
| Matrix multiplication $-a_i^{-1} b_i^T$ | 8 | 24 | 48 | 80 |
| Matrix multiplication $e_i d_j$ | 4 | 12 | 24 | 40 |
| Sum of algebraic operations $l_{oIII}$ | 27 | 81 | 162 | 270 |

Overall relationship between dimension of a block matrix (partitioned into following 1–2–2–2 order) and the number of algebraic operations needed to calculate the submatrices $e_i d_j$ (10), is as follows:

$$l_{oIII} = 27\frac{(n-2)(n-1)}{2} \tag{20}$$

for $n = 3, 4, \ldots$

The number of algebraic operations for different block dimensions $n$ of block matrix (resulting from the partitioning in 1–2–2–2 order) of the self inertia matrices $c_i$ (11), are shown in Table 8.

Table 8
Number of algebraic operations necessary to calculate the self inertia matrices $c_i$

| Block dimension $n$ | 2 | 3 | 4 | 5 |
|---|---|---|---|---|
| Multiplication of triple matrices $b_i a_i^{-1} b_i^T$ | 24 | 48 | 72 | 96 |
| Addition with $c_0^{-1}$ | 1 | 2 | 3 | 4 |
| Inverting of matrix $(c_0^{-1} + b_i a_i^{-1} b_i^T)$ | 1 | 2 | 3 | 4 |
| Calculation of matrix $b_i^T (c_0^{-1} + b_i a_i^{-1} b_i^T)^{-1} b_i$ | 8 | 16 | 24 | 32 |
| Subtractions | 4 | 8 | 12 | 16 |
| Inversion of 4-elementary submatrices | 15 | 30 | 45 | 60 |
| Sum of algebraic operations $l_{oIV}$ | 53 | 106 | 159 | 212 |

The overall relationship between the block dimension $n$ of the block matrix (partitioned in the following 1–2–2–2 order) and the number of algebraic operations needed to calculate $c_i$ matrices (11), is as follows:

$$l_{oIV} = 53(n-1) \tag{21}$$

for $n = 1, 2, \ldots$

The sum of algebraic operations needed to calculate an inverse block matrix with block dimension $n$ (partitioned in 1–2–2–2 order) can be obtained by the following expression:

$$l_o = 27\frac{n^2}{2} + 121\frac{n}{2} - 73 \tag{22}$$

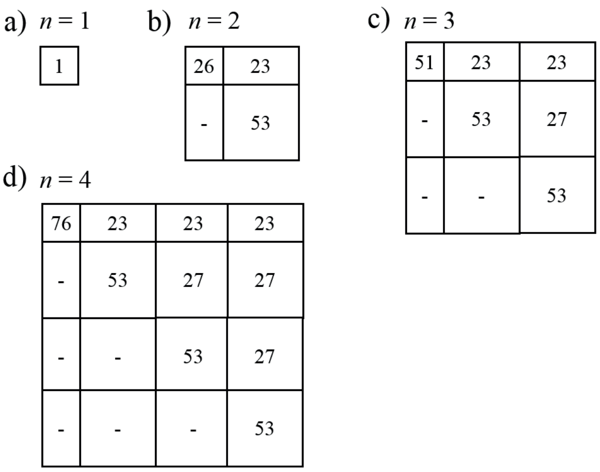


Fig. 3. The number of algebraic operations required to calculate the inverse matrix: a) – for a single-element matrix (special case) the number of operations – 1, b) – for block matrices with block dimension 2 the number of operations – 102, c) – for matrices with block dimension 3 the number of operations – 230, d) – for matrices with block dimension 4 the number of operations – 385

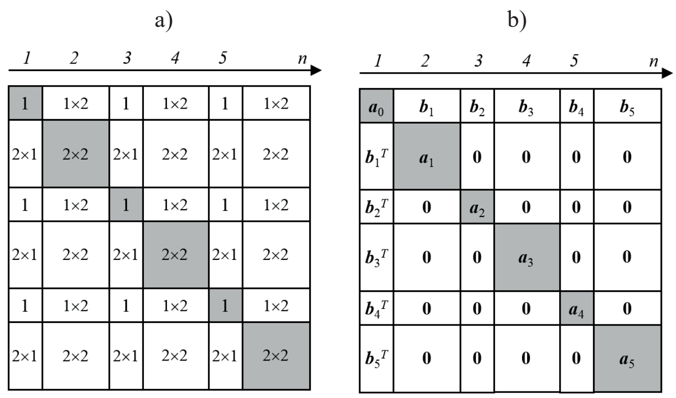

Fig. 4. Matrix partitioning into 1–2–1–2 order: a) – block dimensions of submatrices, b) – marks assignment to the submatrices

Relations showing the number of algebraic operations needed to be done to calculate the elementary matrices of inverted block matrix (with an increase of block dimension $n$ of the input block matrix) can be represented graphically, as presented in Fig. 3. It should be considered that the input block matrix has been partitioned into elementary matrices with dimensions resulting from following 1–2–2–2 order.

**3.3. Case 3 – elementary matrices in partition in the 1–2–1–2 order.** Assume that the block matrix can be divided into elementary matrices in such a way that 1-element matrices $(1\times1)$ and square matrices $(2\times2)$ occur alternately on the main diagonal. All matrices appearing on the main diagonal are symmetrical. The result of this partition is that the elementary matrices in the first row are, alternately, rectangular matrices (of dimension $1\times2$) and 1-element matrices $(1\times1)$, and the first column is the block transposition of the first row. Such partitioning of input block matrix will be referred to the partition in the 1–2–1–2 order. Such conditions correspond to an exemplary block matrix shown in Fig. 4 and the matrix given by (3). Number of algebraic operations necessary to be implemented in order to calculate the form of the leading element $c_0$ (7), using the partitioning of input matrices into elementary matrices in the 1–2–1–2 order, depend on the block dimension $n$ of input matrix. If a block dimension is $n = 2$, the input block matrix is composed of four matrices (Fig. 4a) and the number of algebraic operations necessary to calculate the leading element $c_0$ (7) is the same as in the case analyzed in Chapter 3.2 – formula (18).

Total algebraic operations will be different for larger block dimensions $n$ of the input block matrix, and will depend on the number of 1- and 4-element submatrices, lying on the main diagonal of the input block matrix. Algebraic computation effort associated with the leading element $c_0$ (7) is shown in Table 9.

Table 9
Number of algebraic operations necessary to calculate the leading element $c_0$

| Block dimension $n$ | 2 | 3 | 4 | 5 |
|---|---|---|---|---|
| Inversion of 4-elementary submatrices $a_i^{-1}$ | 15 | 16 | 31 | 32 |
| Multiplication of triple matrices $b_i a_i^{-1} b_i^T$ | 9 | 10 | 19 | 20 |
| Additions / Subtractions | 1 | 2 | 3 | 4 |
| Inversions / Divisions | 1 | 1 | 1 | 1 |
| Sum of algebraic operations $l_{oI}$ | 26 | 29 | 54 | 57 |

In general, the relationship between the block dimension $n$ of the block matrix and the number of algebraic operations, needed to calculate the leading element $c_0$, using the partitioning into elementary matrices following the 1–2–1–2 order, is as follows:

$$l_{oI} = n + 24d + 2(n - d - 1) \tag{23}$$

for $d < n \wedge n = 2, 3, \ldots$; where $d$ – numbers of 4-element submatrices on input block matrix diagonal.

The calculation effort necessary to be performed in order to determine the negative feedback matrices $d_i$ (8), also in this case will depend on the block dimension of the input matrix and the number of 1- and 4-element matrices located on its main diagonal. The calculated numbers of algebraic operations needed to obtain the negative feedback matrices $d_i$ (8), are shown in Table 10.

Table 10
Number of algebraic operations necessary to calculate the negative feedback matrix $d_i$

| Block dimension $n$ | 2 | 3 | 4 | 5 |
|---|---|---|---|---|
| Inversion of 4-elementary submatrices $a_j^{-1}$ | 15 | 16 | 31 | 32 |
| Multiplication of triple matrices $-c_0 b_i a_i^{-1}$ | 8 | 10 | 18 | 20 |
| Sum of algebraic operations $l_{oII}$ | 23 | 26 | 49 | 52 |

The relationship between the dimension of the block matrix (partitioned into blocks in the following 1–2–1–2 order) and the number of algebraic operations needed to calculate the negative feedback matrices $d_i$ (8), is as follows:

$$l_{oII} = 23d + 3(n - d - 1) \tag{24}$$

for $d < n \wedge n = 2, 3, \ldots;$ where $d$ – numbers of 4-element submatrices on the input block matrix diagonal.

The number of algebraic operations required to determine the positive feedback matrices $e_i d_j$ (10) depend on the block dimension $n$ of the input block matrix and the numbers of 1- and 4-element submatrices lying on the main diagonal. As a result, we obtain three types of calculated matrices, namely: 1-element, 2-element (rectangular), 4-element (square and symmetrical) matrices. Analyzing the relationship (10) and the block matrix form shown in Fig. 4, it can be concluded that in the calculation of the 1-element matrices only 1-element matrices take part, while in the calculation of the 2-element matrices two kind of elementary matrices: 1-element and 4-element submatrices are involved. In the calculation of the 4-element matrices two 4-element submatrices, one 1-element and two 2-element submatrices are involved. The relationships between the dimensions of the matrices creating the positive feedback matrices $e_i d_j$ (10), and a number of its internal elements generated are shown in Table 11.

Table 11
Relationship between dimensions – numbers of elements of positive feedback matrices $e_i d_j$ and dimensions of input matrices

| Block dimension $n$ | 2 | 3 | 4 | 5 |
|---|---|---|---|---|
| Inversion of 4-elementary submatrices $a_j^{-1}$ | 15 | 16 | 31 | 32 |
| Multiplication of triple matrices $-c_0 b_i a_i^{-1}$ | 8 | 10 | 18 | 20 |
| Sum of algebraic operations $l_{oII}$ | 23 | 26 | 49 | 52 |

The number of algebraic operations, with a different block dimension $n$ of block matrix (resulting from the partitioning in following 1–2–1–2 order), are shown in Fig. 5.

The sum of all 2-element matrices (in rows and columns, occurring above the main diagonal of inverted block matrix) under increasing block dimension $n$ of the input block matrix (partitioning into elementary matrices in the 1–2–1–2 order), increases accordingly with sequence described by the so-called third diagonal of Lozanic triangle [15]. Successive numbers which represents block dimension $n$ (of input block matrix) correspond a number, representing the sum of all 2-element positive feedback matrices $e_i d_j$ (according to the function $f_2(n)$) are shown in Table 12.

| Case | Dimensions | | | | | Number of elements of positive feedback matrices |
|---|---|---|---|---|---|---|
| | $e_i$ | | $d_j$ | | | |
| | $a_i^{-1}$ | $b_i^T$ | $c_0$ | $b_j$ | $a_j^{-1}$ | |
| 1 | 1×1 | 1×1 | 1×1 | 1×1 | 1×1 | 1 |
| 2 | 1×1 | 1×1 | 1×1 | 1×2 | 2×2 | 2 |
| 3 | 2×2 | 1×2 | 1×1 | 1×1 | 1×1 | 2 |
| 4 | 2×2 | 1×2 | 1×1 | 1×2 | 2×2 | 4 |

Fig. 5. Relationship between dimensions and number of elements of positive feedback matrices $e_i d_j$ and dimensions of input matrices

Table 12
Number of algebraic operations necessary to calculate the positive feedback matrix $e_i d_j$

| Block dimension $n$ | 3 | 4 | 5 | 6 |
|---|---|---|---|---|
| Fig. 5, case 1 (1-element) $a_i^{-1} b_i^T c_0 b_j a_j^{-1}$ | 0 | 0 | 6 | 6 |
| Fig. 5, case 2 or 3 (2-element) $a_i^{-1} b_i^T c_0 b_j a_j^{-1}$ | 27 | 54 | 108 | 162 |
| Fig. 5, case 4 (4-element) $a_i^{-1} b_i^T c_0 b_j a_j^{-1}$ | 0 | 51 | 51 | 153 |
| Sum of algebraic operations $l_{oIII}$ | 27 | 105 | 165 | 321 |

The sum of all 2-element matrices (in rows and columns, occurring above the main diagonal of inverted block matrix) under increasing block dimension $n$ of the input block matrix (partitioning into elementary matrices in the 1–2–1–2 order), increases accordingly with sequence described by the so-called third diagonal of Lozanic triangle [15]. Successive numbers which represents block dimension $n$ (of input block matrix) correspond to a number representing the sum of all 2-element positive feedback matrices $e_i d_j$ (according to the function $f_2(n)$) are shown in Table 13.

Table 13
Number of 2-element positive feedback matrices $e_i d_j$

| $n$ | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | … |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $f_2(n)$ | 1 | 2 | 4 | 6 | 9 | 12 | 16 | 20 | 25 | 30 | 36 | … |

The sum of all 4-element submatrices (lying above the main diagonal) due to a increase of the block dimension $n$ of the input block matrix (partitioning in 1-2-1-2 order), increases following a numerical sequence described by repeated triangular numbers [12-14]. Successive numbers which represents block dimension $n$ corresponds a number, representing the sum of all 4-element positive feedback matrices $e_i d_j$, according to function $f_4(n)$ shown in Table 14.

Table 14
Number of 4-element positive feedback matrices $e_i d_j$

| $n$ | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | … |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $f_4(n)$ | 0 | 1 | 1 | 3 | 3 | 6 | 6 | 10 | 10 | 15 | 15 | … |

The sum of all 1-element matrices due to increase of the block dimension $n$ of the input block matrix (partitioning in

1–2–1–2 order), increases accordingly with a numerical sequence described by repeated triangular numbers [12–14]. Successive numbers representing the block dimension $n$ corresponds a number, representing the sum of all 1-element positive feedback matrices $e_i d_j$, according to function $f_1(n)$ are shown in Table 15.

Table 15
Number of 1-element positive feedback matrices $e_i d_j$

| $n$ | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | … |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $f_4(n)$ | 0 | 0 | 1 | 1 | 3 | 3 | 6 | 6 | 10 | 10 | 15 | … |

The overall relationship between the block dimension $n$ of the input block matrix (partitioning in 1–2–1–2 order) and the number of algebraic operations needed to calculate the positive feedback matrices $e_i d_j$ (10), is as follows:

$$l_{oIII} = 51 f_4(n) + 27 f_2(n) + 6 f_1(n) \tag{25}$$

for $n = 3, 4, \ldots$

Calculated number of algebraic operations for different block dimensions $n$ of input block matrix (partitioning in 1–2–1–2 order), required to obtain the self inertia matrices $c_i$ (11), are shown in Table 16.

Table 16
Number of algebraic operations necessary to calculate the self inertia matrices $c_i$

| Block dimension $n$ | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|
| 4-element matrices | 53 | 53 | 106 | 106 | 159 | 159 |
| 1-element matrices | – | 8 | 8 | 16 | 16 | 24 |
| Sum of algebraic operations $l_{oIV}$ | 53 | 61 | 114 | 122 | 175 | 183 |

The overall relationship between the block dimension $n$ of the input block matrix (partitioned in 1–2–1–2 order) and the number of algebraic operations needed to calculate self inertia matrices $c_i$ (11), is as follows:

$$l_{oIV} = \begin{cases} 53\frac{n}{2} + 8(\frac{n}{2} - 1) & \text{for } n = 2,4,6,8,\ldots \\ 53\frac{n-1}{2} + 8\frac{n-1}{2} & \text{for } n = 1,3,5,7,\ldots \end{cases} \tag{26}$$

The sum of algebraic operations needed to calculate the inverse input block matrix (partitioned in 1–2–1–2 order) may be given by the following expression:

$$l_o = \begin{cases} 102 & \text{for } n = 2 \\ 53\frac{n}{2} + 8(\frac{n}{2} - 1) + l_{oIII} + l_{oII} + l_{oI} & \text{for } n = 2,6\ldots \\ 53\frac{n-1}{2} + 8\frac{n-1}{2} + l_{oIII} + l_{oII} + l_{oI} & \text{for } n = 1,3\ldots \end{cases} \tag{27}$$

Relationships showing the number of algebraic operations needed to be done to calculate the elementary matrices of inverted input block matrix (according to the growth of block dimension) can be represented graphically in Fig. 6. It should be considered that the input block matrix is partitioned into elementary matrices with dimensions consistent with the result of 1–2–1–2 partition order.

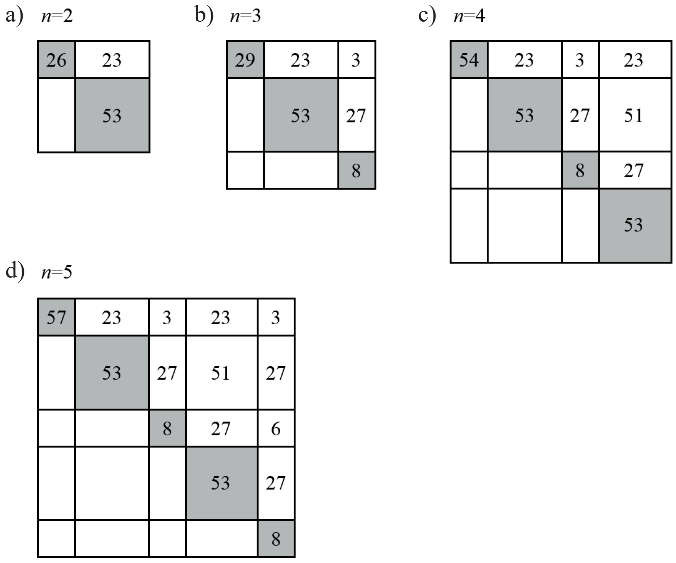


Fig. 6. The number of algebraic operations required to calculate the inverted matrix: a) – for an input block matrix with block dimension $n = 2$ the number of algebraic operations – 102, b) – for block matrices with $n = 3$ the number of operations – 143, c) – for matrices with $n = 4$ the number of operations – 322, d) – for matrices $n = 5$ the number of operations – 396

## 4. The number of algebraic operations and computation times

Numerical experiment was carried out in two stages. The Gauss method is well known and described in the literature, it is also known for its computational complexity, but there is no information on the method "inv" (Matlab function) in terms of the computational complexity. So it was decided to present the effectiveness of shown algorithms in the light of the computational complexity compared to Gasuss method, and in the light of time consumed during the direct calculation using "inv" method and block inversion method.

The number of algebraic operations that must be performed in order to calculate the inverse of an input block matrix using its partitioning into blocks, has been compared with the number of algebraic operations performed using standard Gauss methods. The number of algebraic operations performed by Gauss method, can be represented as a formula [16]:

$$l_o = \frac{2}{3}n^3 + \frac{3}{2}n^2 - \frac{7}{6}n \tag{28}$$

where $n$ – block dimension of the input matrix.

Results of algebraic calculation, showing the comparison between the methods employing the Gauss methods and of the inversion using block matrices, are shown in Figs. 7–9.

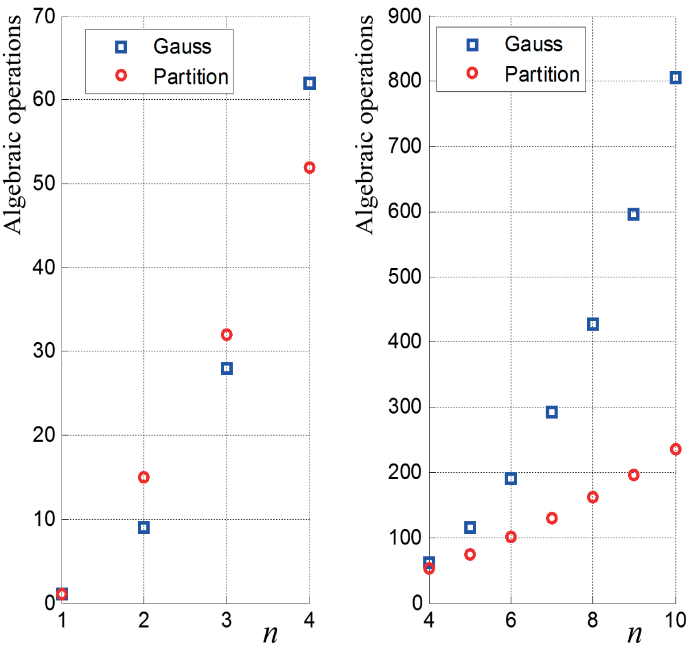


Fig. 7. Number of algebraic operations performed during matrix inversion versus block dimensions: squares – numbers of algebraic operations for Gauss method, circles – numbers of algebraic operations for inversion using block matrices (partitioned in 1–1–1–1 order)

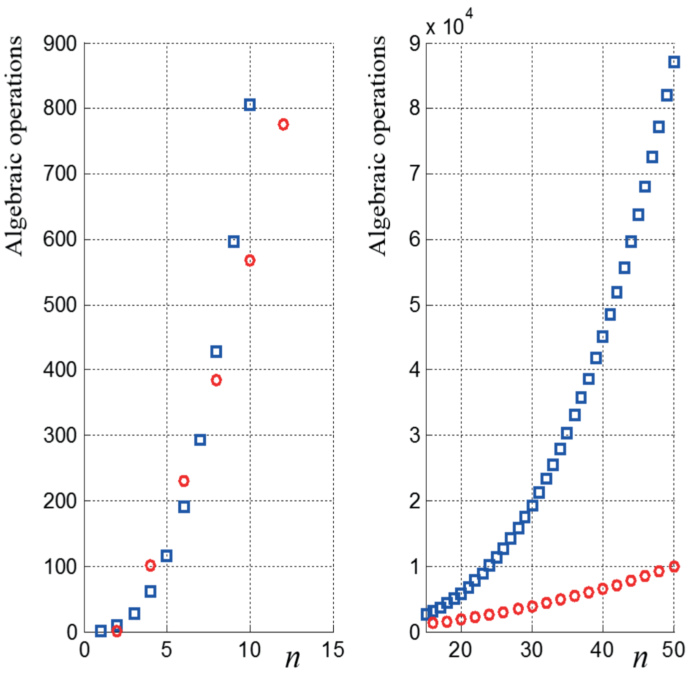


Fig. 8. Number of algebraic operations performed during matrix inversion versus block dimensions: squares – numbers of algebraic operations for Gauss method, circles – numbers of algebraic operations for inversion using block matrices (partitioned in 1–2–2–2 order)

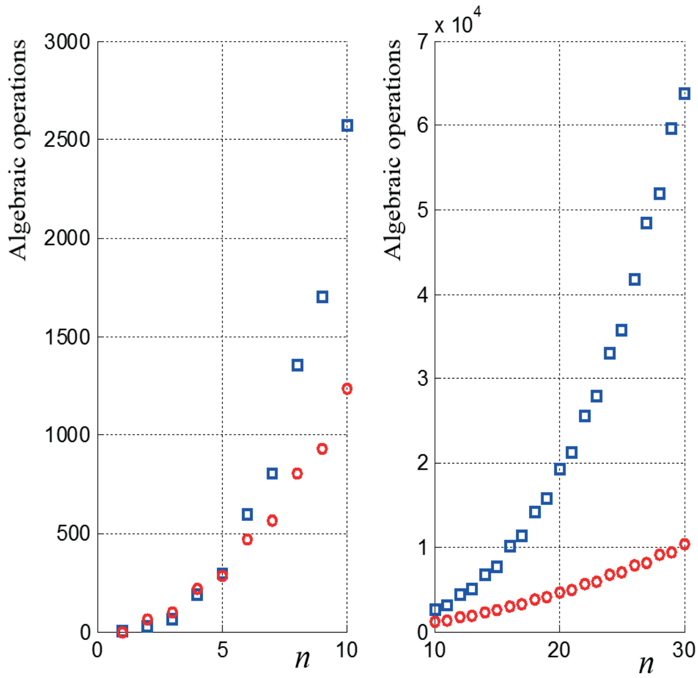


Fig. 9. Number of algebraic operations performed during matrix inversion versus block dimensions: squares – numbers of algebraic operations for Gauss method, circles – numbers of algebraic operations for inversion using block matrices (partitioned in 1–2–1–2 order)

In order to compare the calculation times of matrix inversion performed by block inversion methods and the methods used in commercial software, the implementation of above mentioned algorithm using MATLAB software has been done. In the programming environment the comparison with the "inv" a standard function of the matrix inversion in MATLAB (under the following conditions: forced CPU affinity for MATLAB with single core processor) has been made. The priority of performed tasks had been set to high.Calculations were made on computer equipped with CPU AMD Phenom (tm) II X4 850 3.30 GHz frequency on operating system Windows 7 Professional 64-bit. Calculations has been performed in two loops: an internal loop performing 10,000 times the calculation of matrix inversion and calculating the average time value of single pass through the loop; and outer loop repeated 1000 times and calculating the total computation time. This way of carrying out numerical experiment, using a double loop, alows to determine if indeed calculations are carried out in a single stream of processor unit. Linear relationship between the number of passes the internal loop and the outer loop and the times of performed calculation indicates properly conducted numerical experiment. In other words, in the course of the experiment no events inside the operating system does not interfere with the calculations. For the block inversion we have used matrix partitioned in the 1–1–1–1 order.

The calculation results of the matrix inversion times of the procedures are summarized in Tables 17–20, linear increase of calculation time indicates properly conducted experiments.

Table 17
Inversion of the matrices of block dimension $n = 3$

| | Block inversion | method – "inv" |
|---|---|---|
| time of internal loop [s] | 0.0416(37) | 0.0791(86) |
| time of outer loop [s] | 41.637 | 79.186 |

Table 18
Inversion of the matrices of block dimension $n = 5$

| | Block inversion | method – "inv" |
|---|---|---|
| time of internal loop [s] | 0.0663(32) | 0.131(38) |
| time of outer loop [s] | 66.332 | 131.38 |

Table 19
Inversion of the matrices of block dimension $n = 7$

| | Block inversion | method – "inv" |
|---|---|---|
| time of internal loop [s] | 0.0932(57) | 0.1648(0) |
| time of outer loop [s] | 93.257 | 164.8 |

Table 20
Inversion of the matrices of block dimension $n = 10$

| | Block inversion | method – "inv" |
|---|---|---|
| time of internal loop [s] | 0.1621(6) | 0.2317(4) |
| time of outer loop [s] | 162.16 | 231.74 |

The results presented times of calculations testify in favor of the method of the block matrix inversion compared to the method of "inv" used in Matlab. Computation times are shorter than 1.5 to 2 times.

## 5. Summary

Number of elementary operations required to determine the inverse matrix using the described algorithm depend on the structure of the matrix and block size of block matrix. The described algorithm is most effective for matrices with large block size and divided into blocks of small size. This is particularly noticeable if you compare arrays block divided in the order 1–1–1–1 and 1–2–2–2 in order, the computational complexity of the array divided into blocks of one-piece require a minimum number of algebraic operations during the matrix inversion. Computational complexity in the above two cases, which is obvious, is of the order $O(n^2)$. The computational complexity of matrix inversion divided in order 1–2–1–2 may at first glance appear to be $O(n)$, but if we make a thorough analysis of the increase in the complexity of the component $e_i d_j$, see Table 12, it turns out that it grows as well $O(n^2)$. However, in the case of matrix inversion divided in order 1–2–1–2 computational complexity is between 1–1–1–1 and 1–2–2–2. In all of the analyzed cases, the elements causing the most demanding calculations are the positive feedback matrices $e_i \mathrm{d}_j$, here the complexity is of the order of $O(n^2)$. As shown by formulas (15), (20) and in Table 12, the computational complexity $O(n)$ are linked with the other elements of the matrix like $c_i$, $c_0$ and $d_i$. Interesting results were obtained by juxtaposing formulas describing the computational complexity of the block matrix inversion and the computational complexity resulting from the Gauss method. Inverting block matrix of small size block near $n = 5$ (and in this case smaller then 4, 5 or 7) have a slightly more complex computation than the Gauss method. However, the block matrix size greater than three (in the case of partitioning in the order 1–1–1–1) for presented method is more effective, as well in the case of the matrix partitioned in order 1–2–2–2, this method is more effective for matrices with greater block dimension then $n = 7$. The calculation of the matrix inverse, divided in the order 1–2–1–2 become more efficient than the Gauss method on the block dimension greater then $n = 5$. The exact values of computational complexity of the algorithm for block matrices of the presented structures are given by formulae (17, 22) and (27). Presented method is also characterized by shorter calculation times compared to the standard method of inverting used in MATLAB/Simulink, computation times are shorter than 1.5 to 2 times.

## References


[1] J. de Jesus Rubio, C. Torres, C. Aguilar, "Optimal control based in a mathematical model applied to robotic arms", *International Journal of Innovative Computing, Information and Control* 7(8), 5045–5062, (2011).

[2] T. Trawiński, K. Kluszczyński, W. Kolton, "Lumped parameter model of double armature VCM motor for head positioning system of mass storage devices", *Przegląd Elektrotechniczny* 87(12b), 184–187, [in Polish] (2011).

[3] D. Słota, T. Trawiński, R. Wituła, "Inversion of dynamic matrices of HDD head positioning system", *Appl. Math. Model.* 35(3), 1497–1505, (2011).

[4] T. Trawiński, "Kinematic chains of branched head positioning system of hard disk drives", *Przegląd Elektrotechniczny* 87(3), 204–207,(2011).

[5] T. Trawiński, "Block form of inverse inductance matrix for poliharmonic model of an induction machine", *XXVIII International Conference on Fundamentals of Electrotechnics and Circuit Theory, IC-SPETO'2005*, II, p.371, (2005).

[6] T. Trawiński, *Modeling of Driving Lay-out of Mass Storage Head Positioning System*, Wydawnictwa Politechniki Śląskiej, Gliwice, [in Polish] (2010).

[7] N. Jakovcevic Stor, I. Slapnicar, J. L. Barlow, "Accurate eigenvalue decomposition of arrowhead matrices and applications", *Linear Algebra and its Applications*, 464, 62–89, (2015).

[8] F. Diele, N. Mastronardi, M. Van Barel, E. Van Camp, "On computing the spectral decomposition of symmetric arrowhead matrices", *Computational Science and Its Applications – ICCSA 2004 Lecture Notes in Computer Science*, 3044, 932–941, (2004).

[9] L. Shen, B. W. Suter, "Bounds for eigenvalues of arrowhead matrices and their applications to hub matrices and wireless communications", *EURASIP Journal on Advances in Signal Processing*, 2009, (2009).

[10] G. A. Gravvanis, K. M. Giannoutakis, "Parallel exact and approximate arrow-type inverses on symmetric multiprocessor systems", *Computational Science – ICCS 2006 Lecture Notes in Computer Science*, 3991, 506–513, (2006).

[11] G. A. Gravvanis, "Solving symmetric arrowhead and special tridiagonal linear systems by fast approximate inverse preconditioning", *Journal of Mathematical Modelling and Algorithms* 1(4), 269–282, (2002).

[12] J. Tanton, "A dozen questions about triangular numbers", *Math Horizons* 13(2), 6–7, 21–23 (2005).

[13] P. A. Piza, "On the squares of some triangular numbers", *Mathematics magazine* 23(1), 15–16, (1949).

[14] J. A. Ewell, "A trio of triangular number theorems", *American Mathematical Monthly* 105(9), 848–849, (1998).

[15] "Lozanic triangle", *The On-Line Encyclopedia of Integer Sequences*, (2014).

[16] K. A. Atkinson, *An Introduction to Numerical Analysis*, John Wiley and Sons (2nd ed.), New York, 1989.

[17] W. Hołubowski, D. Kurzyk, T. Trawiński, "A fast method for computing the inverse of symmetric block arrowhead matrices", *Appl. Math. Inf. Sci*. 9(2L), 1–5, (2015).

[18] T. Trawiński, "Inversion method of matrices with chosen structure with help of block matrices", *Przegląd Elektrotechniczny* 85(6), 98–101, [in Polish] (2009).

[19] H. T. Kung, B. Wilsey Suter, "A hub matrix theory and applications to wireless communications". *EURASIP Journal on Advances in Signal Processing* 2007(1), 1–8, (2007).

[20] E. Mizutani, J. W. Demmel, "On structure-exploiting trustregion regularized nonlinear least squares algorithms for neuralnetwork learning", *Neural Networks* 16(5), 745–753, (2003).

[21] M. Bixon, J. Jortner, "Intramolecular radiationless transitions", *The Journal of Chemical Physics* 48(2), 715–726, (1968).

[22] J. W. Gadzuk, "Localized vibrational modes in Fermi liquids. General theory", *Physical Review B* 24(4), 1651, (1981).

[23] A. Ratajczak, "Trajectory reproduction and trajectory tracking problem for the nonholonomic systems", *Bull. Pol. Ac.: Tech*. 64(1), 63–70 (2016).

[24] A. Mazur, M. Cholewiński, "Implementation of factitious force method for control of 5R manipulator with skid-steering platform REX", *Bull. Pol. Ac.: Tech*. 64(1), 71–80 (2016).