An algorithm to modify consistent initialization of differential-algebraic equations obtained by pantelides algorithm using minimally singular subsets

Keisuke Shimakoa and Masanobu Kogab

aDepartment of Creative Informatics, Kyushu Institute of Technology, Fukuoka, Japan; bDepartment of Intelligent and Control Systems, Kyushu Institute of Technology, Fukuoka, Japan

ABSTRACT

In this paper, an algorithm for index reduction of differential algebraic equations (DAE) is proposed. Pantelides algorithm has already been proposed as an algorithm for this purpose. This conventional algorithm has succeeded in reducing the calculation time required for index reduction. However, there exist some DAE systems whose index cannot be reduced correctly with the conventional algorithm because it uses sufficient condition to reduce the index. We propose an algorithm to modify the solution obtained by Pantelides algorithm to deal with a wider class of DAE systems. The proposed algorithm deals with DAE systems by using the necessary and sufficient condition to reduce the index systematically that cannot be reduced by the conventional algorithm. We implemented the proposed algorithm as the functions on Maxima and evaluated the algorithm by using some examples.

1. Introduction

A Differential Algebraic Equation (DAE) is a system of ordinary differential equations (ODE) and algebraic equations. ODEs often appear to describe phenomena in the fields such as science, engineering, and economics. Algebraic equations represent constraints that must be satisfied for phenomena. Since theories regarding DAEs have attracted more recent attention in comparison with that of ODE, they are in a state of flux [1]. The DAE system is usually solved by approximating DAE to ODE. However, it has been required to solve the DAE without approximation with software tools.

If the problem is well-posed, the DAE system can be analytically differentiated to obtain an ODE for each unknown variable. The minimum number of differentiations required for this is called the “index” of the variable. In particular, a DAE system with an index of 2 or more is called a high-index DAE system, and it is said that it is difficult to solve because of hidden constraints on the initial conditions.

One way to solve a high-index DAE system is the reduction of the index. Once the index of the DAE system is reduced to at most 1, a numerically stable solver with an index of 1 will be able to solve stably. For the above reasons, the index reduction is applied to the high-index DAE system.

MATLAB [2] and Mathematica [3] can solve the DAE systems. These tools deal with high-index DAE systems by reducing the index to at most 1. The tools listed here use Pantelides algorithm [4] to reduce the index. Pantelides algorithm uses the condition to differentiate the equations of the DAE systems and reduces the index until they reach 1 at most. However, the index of some DAE system cannot be reduced correctly under the condition provided by the algorithm. Figure 1 shows the result in MATLAB when Pantelides algorithm is actually applied in MATLAB to a DAE system whose index cannot be reduced correctly with Pantelides algorithm.

In this example, reduceDAEIndex function of MATLAB is applied to the DAE system in [4]

\[
\begin{align*}
\dot{x} &= 2y_1 + 3y_2 \cdots f_1 \\
0 &= x + y_1 + y_2 + 1 \cdots f_2 \\
0 &= 2x + y_1 + y_2 \cdots f_3
\end{align*}
\]

where \(x(t), y_1(t), \text{ and } y_2(t)\) are state variables of the system. reduceDAEIndex receives the original equations and variables as the input arguments and generates new variables and equations. \([\text{newEqs, newVars}] = \text{reduceDAEIndex}(\text{eqs, vars})\) converts a high-index DAE system eqs to an equivalent system newEqs with index 1. Pantelides algorithm is used for this process. After the conversion, reduceDAEIndex calls isLowIndexDAE function to check whether the index of DAE system is at most 1. If the index of newEqs is 2 or more, reduceDAEIndex issues a warning. It can be seen that the index of DAE system cannot be correctly reduced.

\(\Sigma\) method [5] was also proposed as a method for index reduction, but the index of some DAE system cannot be reduced correctly with the algorithm. Reference [6] proposed an approach with symbolic

CONTACT Keisuke Shimako shimako@mk.ces.kyutech.ac.jp

© 2021 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group. This is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.
simplifications, which deals with first-order linear constant coefficient DAEs and some first-order DAEs with simple nonlinearity of variables, but not all. Reference [7] proposed a structural-algebraic method to fix the failure of Pantelides algorithm but is limited to coupled systems. Reference [8] proposed two methods to deal with a wider class of DAE by modifying the structural analysis used in $\Sigma$ method and Pantelides algorithm. However, some DAE that cannot be dealt with by the method introduced in this reference exists because methods in reference [8] use another sufficient condition to differentiate equations.

In [9], we proposed an algorithm to modify the solution obtained by Pantelides algorithm, which is used in many software tools, to deal with a wider class of DAE systems. In addition, since the proposed algorithm uses the necessary and sufficient condition to differentiate equations of DAE, our method can deal with a wider class of DAE systems than method proposed in reference [8]. This paper proposes a revised algorithm along with more descriptions about the algorithm and the examples. The main difference from [9] is that the termination condition of the algorithm is clearly defined and the updating operations in the iteration of the algorithm is divided into two stages for improving the efficiency. Although the same examples in [9] were used in this paper, the explanation and the discussion after being applied to the revised algorithm are given.

This paper is organized as follows. In Section 2, the problem to be solved is formulated. In Section 3, the conventional algorithm for reducing the index of high-index DAE and the problem with this algorithm are introduced. In Section 4, a new algorithm is proposed. In Section 5, the behaviour and execution time of the proposed algorithm is evaluated using some examples. Finally in Section 6, we conclude the paper.

### 2. Problem formulation

The problem to be solved in this paper is formulated as follows.

**Problem:** Consider the DAE system

$$ f(x, \dot{x}, y, t) = 0 \quad (2) $$

where $x, \dot{x} \in \mathbb{R}^n, y \in \mathbb{R}^m, f : G \subseteq \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^m \times R \rightarrow \mathbb{R}^{n+m}$. Variables in the equations are distinguished by $x$ and $y$. While $x$ is a variable whose time derivative $\dot{x}$ exists in the DAE system to be considered, $y$ is a variable whose time derivative $\dot{y}$ does not exist in the DAE system.

Derive an algorithm that provides the additional condition that the initial values of DAE should satisfy.

In Pantelides algorithm, system (2) is represented as

$$ f(z, x, t) = 0, \quad (3) $$

where $z = (\dot{x}, y)$ is the variable which introduces the new variable when it is differentiated.

If there is an $f$ in the DAE system to be considered in which $f_z$ obtained by partially differentiating $f$ with respect to $z$ has equal to or more columns than rows and is not full row rank, Pantelides algorithm cannot correctly perform index reduction.

This paper proposes a method to deal with DAE system represented by system (2) where $f_z$ has equal to or more columns than rows and is not full row rank.

### 3. Conventional algorithm

In this section, the conventional method is introduced. Although Pantelides algorithm was originally proposed to find the consistent initial conditions for DAE systems, it is commonly used as an index reduction algorithm. The initial conditions of the system whose index of the DAE system is at most 1 is easily obtained. Since this study modifies the solution obtained by Pantelides algorithm, we will discuss the method to determine the initial conditions of the DAE system.

#### 3.1. Subset to be differentiated

When it comes to the initial conditions of a DAE system, the initial conditions must not only satisfy the original equations but also satisfy the hidden constraints obtained by differentiation. However, blindly differentiating equations is not efficient because some equations show hidden constraints by differentiation and others do not. Therefore, Pantelides algorithm establishes its own condition to select the equations to differentiate.

Whether differentiating equations reveals hidden constraints on the initial conditions depends not only on an equation but also on a subset

$$ \tilde{f}(\tilde{x}, \tilde{x}, t) = 0. \quad (4) $$

$\tilde{f}$ is composed of $k$ equations with $\tilde{x} \in \mathbb{R}^\ell$ and $\tilde{z} \in \mathbb{R}^r$. $r$ is the row rank of $\tilde{f}_z$. Since $\tilde{f}_z$ is a matrix of $k$ rows and $\ell$ columns, it holds the relation

$$ r \leq \min(\ell, k). \quad (5) $$

Based on the above conditions, we consider which subset of equations should be differentiated and the hidden constraints for determining the initial conditions of the DAE system will appear. Assuming that $(\tilde{f}_z : \tilde{f}_z)$ has full
row rank, i.e. \( k \), and that subset (4) is differentiable, we differentiate subset (4) with respect to time to obtain:

\[
\dot{\tilde{z}} + \tilde{f}_{\tilde{x}} \tilde{z} + \tilde{f}_{\tilde{t}} = 0.
\] (6)

which can be transformed into

\[
(\tilde{f}_{\tilde{z}} : \tilde{f}_{\tilde{x}}) \begin{pmatrix} \dot{\tilde{z}} \\ \dot{\tilde{x}} \end{pmatrix} = -\tilde{f}_{\tilde{t}}.
\] (7)

With the relationship (5), Equation (7) can be transformed into

\[
\begin{pmatrix} A & B \\ O & C \end{pmatrix} \begin{pmatrix} \dot{\tilde{z}} \\ \dot{\tilde{x}} \end{pmatrix} = \begin{pmatrix} a \\ b \end{pmatrix}
\] (8)

by using the elementary row operations. Equation (8) gives

\[
A \dot{\tilde{z}} + B \dot{\tilde{x}} = a
\] (9)

and

\[
C \dot{\tilde{x}} = b
\] (10)

where \( A \in R^r \times R^l, B \in R^r \times R^l, C \in R^{k-r} \times R^l \) are matrices and \( a \in R^r, b \in R^{k-r} \) are vectors, whose elements are functions of \( (\tilde{z}, \tilde{x}, t) \) in general.

Differentiation introduces the new variables \( \tilde{z} \) which is not included in the original system (2); it creates the new Equations (9) and (10). We note that, because of relationship (5), \( \ell \) is equal to or greater than \( r \) and one can therefore partition vector \( \tilde{z} \) into \( \tilde{z}_1 \in R^r \) and \( \tilde{z}_2 \in R^{\ell-r} \), and matrix \( A \) into \( A_1 \in R^r \times R^r \) and \( A_2 \in R^r \times R^{\ell-r} \) such that \( A_1 \) is nonsingular. Solving the resulting equations, one obtains from Equation (9) an explicit expression for \( \tilde{z}_1 \):

\[
\tilde{z}_1 = A_1^{-1} a - A_1^{-1}(B \dot{\tilde{x}} + A_2 \tilde{z}_2)
\] (11)

Now for any triplet \((\tilde{x}, \tilde{z}, \tilde{y})\), one can always find values for the new variables \( \tilde{z} \) such that Equation (9) is satisfied. No new constraint concerning the original variable set appears in Equation (9).

However, since by assumption \( \text{rank}(\tilde{f}_{\tilde{z}} : \tilde{f}_{\tilde{x}}) = k \), matrix \( C \) in Equation (10) has full row rank, \( (k-r) \). Thus, equation (10) constitutes \( (k-r) \) new equations which the original variable set must satisfy together with the original system (2). Therefore, the equations that must be satisfied by a set of consistent initial conditions can be generated by locating all subset (4) such that

\[
r < k
\] (12)

3.2. Conventional method

Pantelides algorithm uses

\[
\ell < k
\] (13)

as the condition to select the equations to differentiate. A subset of equations that satisfy condition (13) is called \textit{structurally singular} with respect to the variable subset \( \tilde{z} \). A structurally singular subset is called \textit{minimally structurally singular} (MSS) if none of its proper subsets is structurally singular. Pantelides algorithm differentiates MSS subsets.

The purpose of using such unique condition to differentiate the equations is to speed up the processing. Assuming a DAE system consists of \((n + m)\) equations, there are \(2^{(n+m)} - 1\) non-empty subsets. If we try to find a subset to be differentiated by condition (12), in the worst case we will have to check the rank of the Jacobian matrix \(2^{(n+m)} - 1\) times, and if \((n + m)\) increases, the algorithm will take quite long time. On the other hand, in the case of condition (13), the subset to be differentiated can be searched only by considering the relationship between the numbers, so that even if \((n + m)\) increases, the calculation time does not increase so much.

3.3. The problem with the conventional algorithm

Here, we point out the fundamental problem with the conventional algorithm. The conventional algorithm is using sufficient condition to differentiate equations of DAE. Because of relationship (5), the condition (13) is a sufficient condition for condition (12). Even though condition (12) is satisfied, the algorithm terminates without differentiating the equation to be differentiated if condition (13). Such problems cannot be dealt with Pantelides algorithm. Such cases occur when \( \tilde{f}_{\tilde{z}} \) has equal to or more columns than rows and is not row full rank. We solve this problem by using the necessary and sufficient condition for the equations that cannot be differentiated by sufficient condition systematically.

4. Proposed algorithm

In this section, we introduce the proposed algorithm.

4.1. Minimally singular subset

In the algorithm proposed in this study, condition (12) is used to select the equations to differentiate. A subset of equations that satisfies condition (12) is defined as a \textit{singular subset}. In addition, a singular subset is called \textit{minimally singular} (MS) if none of its proper subsets is singular. The proposed algorithm differentiates MS subsets.

4.2. The flowchart of the proposed algorithm

The flowchart of the proposed algorithm is shown in Figure 2.

The dotted block in Figure 2 represents the conventional algorithm, and the solid part represent the blocks of the proposed algorithm.
The proposed algorithm are described in detail below. First, the conventional algorithm is applied to the given system. If the conventional algorithm differentiates some equations, we update the DAE. Next, we use Jacobian matrix to determine whether the initial conditions of the DAE system could been obtained correctly. If the Jacobian matrix does not include any linearly dependent vector, it is determined as “Yes” and the algorithm ends. Otherwise, the subset of equations that should be differentiated are searched and transformed into a suitable form before they are reapplied to the conventional algorithm.

4.2.1. Conventional algorithm
Before searching for the MS subset, Pantelides algorithm is applied to the DAE system once in order to reduce the MS subsets. When Pantelides algorithm is applied to the DAE system, the subset that contains vectors that are linearly independent in \( f \) is differentiated by Pantelides algorithm. Therefore, only the subset that contains vectors that are linearly dependent in \( \hat{f} \) remains without differentiation. Therefore, it is only necessary to search for the subset that includes a vector that is linearly dependent in \( \hat{f} \). As a result, the conventional algorithm is used to reduce the execution time of the entire algorithm.

4.2.2. Update of DAE
The blocks labelled with “Update of DAE” define new system

\[
\hat{f}(\hat{x}, \hat{\dot{x}}, t) = 0. \tag{14}
\]

which collects only the equations that have not been differentiated in the conventional algorithm.

Though the equation which was differentiated is not included in \( \hat{f} \), the obtained differentiated one is included in \( \hat{f} \) instead of one in \( f \). Therefore, \( \hat{f} \) consists of \((n + m)\) equations because of the assumption that the given system (2) has \((n + m)\) equations.

The blocks are placed next to the blocks labelled with “Conventional Algorithm” and “MS Transformation.” The block that is placed next to the block labelled with “Conventional Algorithm” reduces the cost of MS Search because the number of subsets decreases. The block placed next to the block labelled with “MS Transformation” adjusts the number of equations. This block is needed at this time because the conventional algorithm makes the assumption that the number of variables matches the number of equations.

4.2.3. Checking if index \( r \leq 1 \)
From the description of Section 3.1, checking whether

\[
r < n + m, \tag{15}
\]

holds makes it possible to check if index \( r \leq 1 \) holds and to determine whether there is still a hidden constraint for determining the initial condition. If (15) holds, differentiation is still required and MS Search is performed, otherwise the algorithm terminates.

The loop from “Conventional Algorithm” to “MS Transformation” in Figure 2 is repeated until the initial conditions can be obtained correctly. After the algorithm is terminated, all equations are returned.

4.2.4. MS search
MS Search is performed in the subset that is generated from \( \hat{f} \). MS Search is realized by examining \( r \), which is the rank of \( \hat{f} \), for all possible equations in \( f \) in descending order of \( k \). The equations in the subset where \( r < k \) are judged as equations to be differentiated.

Then, the subset containing the equation judged as the equation to be differentiated is excluded from MS search. Therefore, MS subsets could be searched as a result.

In addition, from the description in Section 3.1, there are only \((k - r)\) subsets that, show hidden conditions for determining initial conditions. Therefore, the MS search ends when \((k - r)\) differentiating subsets are found and the number of search can be reduced.

4.2.5. MS transformation
This block differentiates MS subsets and transforms them. MS subsets are differentiated according to the procedure from (6) to (10). Since there is the assumption that MS subsets satisfies condition (12), Equation (10) is always derived by the transformation. Since Equation (10) does not include \( z \), it satisfies the condition (13) and will be differentiated at the following conventional algorithm. Without MS transformation, we may leave the subset that does not satisfy the condition (13) but the condition (12).

5. Example
In this section, we used some examples to evaluate the effectiveness of the proposed algorithm in terms of the reduction of the index and the execution time.

5.1. An example in [4]
We apply the proposed algorithm to example (1) in [4] whose initial conditions cannot be obtained correctly with the conventional algorithm.
We consider the subset \( \tilde{f} \) of the example (1) consisting of the last two equation \((k = 2)\) with \( \tilde{z} = (y_1, y_2)^T \). Then we have
\[
\tilde{f}_z = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}
\] (16)
and
\[
r = \text{rank}(\tilde{f}_z) = 1
\] (17)
Clearly this subset should be differentiated because of condition (12). However, since \( \ell \) is 2, the condition (13) to differentiate the equation used in the conventional algorithm is not satisfied. Therefore, the conventional algorithm terminates without detecting all the subsets which should to be differentiated.

We show how the proposed algorithm works for the example (1). When, the conventional algorithm is applied to the system, no differentiation occurs because example (1) has no subset that satisfies condition (13).

Since the DAE system is not updated,
\[
\hat{f} = \{f_1, f_2, f_3\}, \quad z = [x, y_1, y_2].
\] (18)
Differentiating \( \hat{f} \) with respect to \( z \) generates
\[
\hat{f}_z = \begin{pmatrix} 1 & 2 & 3 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix}
\] (19)
whose rank is 2. Since \( r < k \), the algorithm does not end, and continue to search for the subset to be differentiated. At this stage, since \((k - r) = 1\), the search ends when a singular subset is searched.

Next, the MS search follows. All the possible subsets in this example are
\[
\{f_1\}, \{f_2\}, \{f_3\}, \{f_1, f_2\}, \{f_1, f_3\}, \{f_2, f_3\}, \{f_1, f_2, f_3\}\.
\] (20)
We sort the subsets in ascending order of number of equations. and examine \( \hat{f}_z \) in this order to find the subset to be differentiated.

Since the subset consisting of \( f_2 \) and \( f_3 \) satisfies of \( r < k \), the subset is the condition to be differentiated. Since MS subset was found, we do not have to check \( \{f_1, f_2, f_3\} \).

Next, the MS subset
\[
0 = x + y_1 + y_2 + 1 \cdots f_2
\]
\[
0 = 2x + y_1 + y_2 \cdots f_3
\] (21)
is differentiated to give
\[
\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 2 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ x \end{pmatrix} = 0.
\] (22)
Using elementary row operation, Equation (22) is transformed into,
\[
\begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ x \end{pmatrix} = 0
\] (23)
i.e.
\[
y_1 + y_2 + \dot{x} = 0 \cdots f_4
\]
\[
\dot{x} = 0 \cdots f_5
\] (24)
Here, \( f_5 \) corresponds to Equation (10) and is the hidden constraint, which was not obtained by the conventional algorithms.

Since, \( f_2 \) and \( f_3 \) is differentiated, \( \hat{f} \) is updated to
\[
\hat{f} = \{f_1, f_4, f_5\} \quad z = [x, y_1, y_2].
\] (25)
When we apply the conventional algorithm to this subset, \( f_1 \) and \( f_5 \) are differentiated and the following equations are derived.
\[
\dot{x} = x + 2y_1 + 3y_2 \cdots f_6
\]
\[
\dot{x} + y_1 + y_2 = 0 \cdots f_4
\]
\[
\dot{x} = 0 \cdots f_5
\] (26)
and, \( \hat{f} \) is updated to
\[
\hat{f} = \{f_6, f_4, f_5\} \quad z = [x, y_1, y_2]
\] (27)
and \( \hat{f}_z \) becomes
\[
\hat{f}_z = \begin{pmatrix} 1 & -2 & -3 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{pmatrix}
\] (28)
Since the rank of \( \hat{f}_z \) is equal to the number of equations, the algorithm ends.

5.2. Implementation and application

We implemented the proposed algorithm as several functions on Maxima [10]. Figure 3 shows the execution result on Maxima when they are applied to example (1). (%i 18) defines example (1) as \( F \).
“DAEInitialization()” is the function in which the proposed algorithm is implemented. (§6.19) outputs
\[
\begin{align*}
y_2 + y_1 + x + 1 & = 0 \\
y_2 + y_1 + 2x & = 0 \\
3y_2 + 2y_3 - \dot{x} + x & = 0 \\
\dot{x} & = 0 \\
y_2 + y_1 + \dot{x} & = 0 \\
3y_2 + 2y_1 - \dot{x} + \dot{x} & = 0 \\
\dot{x} & = 0
\end{align*}
\]
which is the DAE system obtained by the proposed algorithm. We confirmed that Equation (29) are identical with the result by hand.

5.3. Execution time

We consider the execution time of the proposed algorithm from two points of view. First, we considered the time for determining whether the equations output from Pantelides algorithm is still to be differentiated. Second, we consider the effect of the number of equations on execution time. When the number of equations of the DAE system is \((n + m)\), MS Search examines \(2^{n+m} - 1\) Jacobi matrices at worst. The execution time is approximately doubled for each number of equations at worst.

Table 1 shows the execution environment.

| Processor  | Intel core i5     |
|------------|-------------------|
| Memory     | 32GB              |
| OS         | Windows 10 64-bit |
| Maxima     | Maxima 5.43.2     |

The execution time is measured by Maxima’s “time” function [10]. This is the function that returns the time to calculate the output. The time is not the elapsed time but the estimate of the internal computation time. In this paper, the average of 1000 executions is shown below as the execution time.

5.3.1. Checking rank of Jacobian

We consider the time for determining whether the equations outputed from Pantelides algorithm are still to be differentiated. We prepare an example that conventional algorithm,

We use
\[
\begin{align*}
\dot{x}_1 & = x + 2y_1 + 3y_2 \\
0 & = x + y_1 + y_2 + 1 \\
0 & = 2x + 2y_1 + y_2.
\end{align*}
\]

We measure the difference between the execution time of conventional algorithm and that of the proposed one. Table 2 shows the execution time of the conventional algorithm and the proposed one. The extra time to determine if the differentiation is needed is less than 10%.

5.3.2. Number of equations

The effect of the number of equations in the DAE system on the execution time is discussed. We prepare some examples that conventional algorithm cannot deal with and have different numbers of equations, and compare their execution times to discuss the effects.

As for the system with three equations, we use example (1). As for the system with four equations, we use
\[
\begin{align*}
\dot{x} & = x + 2y_1 + 3y_2 + 4y_3 \\
0 & = x + y_1 + y_2 + y_3 \\
0 & = 3x + y_1 + y_2 + 2y_3 \\
0 & = 5x + y_1 + y_2 - y_3.
\end{align*}
\]

As for the system with five equations, we use
\[
\begin{align*}
\dot{x} & = x + 2y_1 + 3y_2 + 4y_3 + 5y_4 \\
0 & = 2x + y_1 + y_2 + y_3 + y_4 + 1 \\
0 & = 3x + y_1 + 2y_2 + 2y_3 + 2y_4 \\
0 & = 5x + y_1 + 2y_2 + 3y_3 + 3y_4 \\
0 & = 7x + 2y_1 + 2y_2 + y_3 + y_4.
\end{align*}
\]

Table 3 shows the execution time of three examples. The execution time is approximately doubled for each number of equations.

6. Conclusion

We proposed an algorithm to modify the solution obtained by Pantelides algorithm, which is used in many software tools, to deal with a wider class of DAE systems. We implemented the proposed algorithm as the functions on Maxima and evaluated the algorithm by using some examples.

The proposed algorithm can reduce the index of the DAE system, which could not be dealt with by the conventional algorithm in Equation (2), by using the necessary and sufficient condition. MS search in the proposed algorithm sorts the equations so that the number of searches could be as small as possible, but the execution time can be approximately at worst doubled for each number of the equations.

Table 2. Execution time of the two algorithms.

| Algorithm               | Execution time [ms] |
|-------------------------|---------------------|
| Conventional algorithm  | 1.23                |
| Proposed algorithm      | 1.32                |

Table 3. Execution time of three examples.

| DAE       | Number of equation | Execution time [ms] |
|-----------|--------------------|---------------------|
| (1)       | 3                  | 4.953               |
| (31)      | 4                  | 9.250               |
| (32)      | 5                  | 17.86               |
Since this paper deals with the same problem formulation as Pantelides algorithm, it may be applicable to other researches based on Pantelides algorithm. For example, reference [11] reports that the algorithm has been modified to allow it to be applied to the delayed differential algebraic equations. The proposed algorithm may be applicable to such works.

Disclosure statement
No potential conflict of interest was reported by the author(s).

Notes on contributors

Keisuke Shimako
He received his B.E. degrees from Kyushu Institute of Technology, Japan, in 2020. In 2020, he joined Kyushu Institute of Technology Graduate School. His research interests include control systems design and numerical computation.

Masanobu Koga
He received his B.E., M.E., and D.E. from Tokyo Institute of Technology, Japan, in 1988, 1990, and 1993, respectively. In 2000, he joined Kyushu Institute of Technology, where he is currently a Professor of the Department of Intelligent and Control Systems. His research interests include control systems design and numerical computation. He is a member of IEEE.

References

[1] Ascher UM, Petzold LR. Computer methods for ordinary differential equations and differential-algebraic equations. Philadelphia: SIAM; 2006.
[2] The MathWorks Inc, Simulink. Available from: http://jp.mathworks.com/products/simulink/.
[3] WOLFRAM, Wolfram. Available from: http://www.wolfram.com.
[4] Pantelides CC. The consistent initialization of differential-algebraic systems. SIAM J Sci Stat Comput. 1988;9(2):213–231.
[5] Pryce JD. A simple structural analysis method for DAEs. BIT Numer Math. 2001;41(2):364–394.
[6] Chowdhry S, Krendl H, Linninger AA. Symbolic numeric index analysis algorithm for differential algebraic equations. Ind Eng Chem Res. 2004;43(14):3886–3894.
[7] Scholz L, Steinbrecher A. Structural-algebraic regularization for coupled systems of DAEs. BIT Numer Math. 2016;56:777–804.
[8] Tan G, Nedialkov NS, Pryce JD. Conversion methods for improving structural analysis of differential-algebraic equation systems. BIT Numer Math. 2017;57:845–865.
[9] Shimako K, Koga M. An extension of pantelides algorithm for consistent initialization of differential-algebraic equations using minimally singularity. In: 2020 59th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE), Chiang Mai, Thailand. 2020. p. 1676–1681.
[10] Sourceforge, Maxima, a Computer Algebra System. Available from: http://maxima.sourceforge.net/.
[11] Ahrens I, Unger B. The Pantelides algorithm for delay differential-algebraic equations, arXiv:1908.01514, 2020.