LUND UNIVERSITY

Index Reduction in Differential-Algebraic Equations Using Dummy Derivatives

Mattsson, Sven Erik; Söderlind, Gustaf

1991

Document Version:
Publisher's PDF, also known as Version of record

Link to publication

Citation for published version (APA):
Mattsson, S. E., & Söderlind, G. (1991). *Index Reduction in Differential-Algebraic Equations Using Dummy Derivatives*. (Technical Reports TFRT-7477). Department of Automatic Control, Lund Institute of Technology (LTH).

Total number of authors:
2

General rights
Unless other specific re-use rights are stated the following general rights apply:
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.
• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal

Read more about Creative commons licenses: https://creativecommons.org/licenses/

Take down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

LUND UNIVERSITY

PO Box 117
221 00 Lund
+46 46-222 00 00

Download date: 07. Oct. 2022

CODEN: LUTFD2/(TFRT-7477)/1-018/(1991)

Index Reduction in Differential-Algebraic Equations Using Dummy Derivatives

Sven Erik Mattsson
Gustaf Söderlind

Department of Automatic Control
Lund Institute of Technology
June 1991

Department of Automatic Control
Lund Institute of Technology
P.O. Box 118
S-221 00 Lund Sweden

Document name
Report

Date of issue
June 1991

Document Number
CODEN:LUTFD2/(TFRT-7477)/1-018/(1991)

Author(s)
Sven Erik Mattsson and Gustaf Söderlind

Supervisor

Sponsoring organisation

Title and subtitle
Index Reduction in Differential-Algebraic Equations Using Dummy Derivatives

Abstract

A new index reduction algorithm for DAEs is developed. In the usual manner, parts of the DAE are differentiated analytically and appended to the original system. For each additional equation, a derivative is selected to be replaced by a new algebraic variable called a *dummy derivative*. The resulting augmented system is at most index 1, but no longer overdetermined. The dummy derivatives are not subject to discretization; their purpose is to annihilate part of the dynamics in the DAE, leaving only what corresponds to the dynamics of a state-space form. No constraint stabilization is necessary in the subsequent numerical treatment. Numerical tests indicate that the method yields results with an accuracy comparable to that obtained for the corresponding state-space ODE.

Key words
Differential-algebraic, index reduction, dummy derivative, constraint stabilization, solution invariant, automatic differentiation

Classification system and/or index terms (if any)

Supplementary bibliographical information

ISSN and key title

ISBN

Language
English

Number of pages
18

Recipient's notes

Security classification

The report may be ordered from the Department of Automatic Control or borrowed through the University Library 2, Box 1010, S-221 03 Lund, Sweden, Telex: 33248 lubbis lund.

1. Introduction

We shall develop a new index reduction technique for differential-algebraic equations

$$F(t, x, \dot{x}) = 0. \tag{1.1}$$

We assume that the problem is *solvable*, Brenan et al. (1989), with a unique, smooth solution when supplied with an appropriate number of consistent initial conditions. The *index* of the problem equals the minimum number of times that all or part of (1.1) must be differentiated with respect to t in order to determine $\dot{x}$ as a continuous function of x and t, op. cit., and is assumed to be constant along the solution. In addition, F is assumed to be sufficiently differentiable to allow the proposed index reduction technique. For alternative definitions of the index, see Hairer et al. (1989) and Gear (1990).

It is well-known that it is numerically difficult to solve a high-index DAE. Index reduction methods provided in the literature, see e.g. Brenan et al. (1989, p. 33), can be used as a remedy. However, it is often less satisfactory to solve the underlying ODE, or UODE, that has been derived from the DAE through index reduction. The reason is that the set of solutions to the UODE is larger than the corresponding set of solutions to the original DAE; the algebraic relations of the DAE are only implicit in the UODE as solution invariants. Unless linear, these invariants are generally not preserved under discretization. As a result, the numerical solution drifts off the algebraic constraints, often leading to instabilities, Führer and Leimkuhler (1991). Consequently, so-called constraint stabilization techniques have been devised, Baumgarte (1972), Gear et al. (1985), Alishenas (1991).

To avoid such difficulties, one may try to obtain a low-index formulation, with a solution set identical to that of the original problem. This can be achieved by augmenting the system as the index reduction proceeds: all original equations, and their successive derivatives are retained in the process. The result is an overdetermined but consistent index 1 DAE. Like invariants, however, consistency is generally lost when the system is discretized. Therefore special projection techniques are required for the numerical solution, see e.g. Führer and Leimkuhler (1990) or Eich et al. (1990).

The index reduction technique proposed in this paper, which was outlined in Mattsson and Söderlind (1990), overcomes the latter complication by introducing a new dependent variable for each new equation generated in the reduction process. This generates an augmented, determined index 1 DAE from the original problem. The technique is applicable to large classes of problems and can be practically implemented, using symbolic manipulation or automatic differentiation, Rall (1981).

We shall outline our method in a simple example. A general nonlinear DAE can be locally transformed to an index 1 or index 0 problem by means of differentiation of equations and nonlinear coordinate changes. The linear problem below, however, is much simpler. Consider the DAE problem

$$\dot{x} = y \tag{1.2a}$$

$$\dot{y} = z \tag{1.2b}$$

$$x = f(t). \tag{1.2c}$$

This is an index three "derivative chain" with solution $x = f(t)$, $y = \dot{f}(t)$ and $z = \ddot{f}(t)$. It may be thought of as prototypical for prescribed-trajectory

problems in mechanics, where one wants to calculate the (usually generalized) forces required for the system to accomplish the desired action. In such an application x, y and z would represent position, velocity and force per unit mass, respectively, and $f(t)$ is the prescribed trajectory for the system.

A sufficient condition for the index to be at most 1, is that we are able to solve for the highest order derivatives in the system, i.e. $\dot{x}$, $\dot{y}$ and z. This is obviously impossible in (1.2). Differentiating (1.2a) once and (1.2c) twice yields, after reordering equations and variables,

$$\ddot{x} = \ddot{f}(t) \qquad (1.2c'')$$
$$\dot{y} = \ddot{x} \qquad (1.2a')$$
$$z = \dot{y}, \qquad (1.2b)$$

which is index 1. Here (1.2a′) denotes the derivative with respect to t of (1.2a).

Now, consider the overdetermined but consistent system obtained by augmenting the original system by the successive derivatives of (1.2a) and (1.2c):

$$x = f(t) \qquad (1.3c)$$
$$\dot{x} = \dot{f}(t) \qquad (1.3c')$$
$$\ddot{x} = \ddot{f}(t) \qquad (1.3c'')$$
$$y = \dot{x} \qquad (1.3a)$$
$$\dot{y} = \ddot{x} \qquad (1.3a')$$
$$z = \dot{y}. \qquad (1.3b)$$

For each *differentiated equation* appended to the original system, we need one "new" dependent variable to make the augmented system determined rather than overdetermined. This is achieved by replacing one derivative from each differentiated equation by a new algebraic variable. Thus we eliminate $\ddot{x}$ by substituting a *dummy derivative* x'' for $\ddot{x}$ wherever it occurs in the system (1.3). Although $x'' = \ddot{x}$, *the dummy derivative is a purely algebraic variable and is not subject to discretization.* Similarly we replace $\dot{x}$ by x' and $\dot{y}$ by y'. This yields the augmented but determined system

$$x = f(t) \qquad (1.4a)$$
$$x' = \dot{f}(t) \qquad (1.4b)$$
$$x'' = \ddot{f}(t) \qquad (1.4c)$$
$$y = x' \qquad (1.4d)$$
$$y' = x'' \qquad (1.4e)$$
$$z = y'. \qquad (1.4f)$$

This purely algebraic (hence index 1) system is mathematically equivalent to (1.2). No initial conditions can be imposed, and no discretization is required for the numerical solution. The system is nonsingular since it is possible to solve algebraically for the six unknowns x, x', x'', y, y' and z. Although this example is rather special, it demonstrates the important aspect of the new reduction technique, that the system can be solved numerically without discretizing all derivatives.

In the following sections we shall describe how to proceed in the general case. We first show how to obtain the appropriate differentiated system and then how to select dummy derivatives to make the augmented system determined. We shall also deal with pivoting of the selected set of dummy derivatives, before presenting numerical results.

2. Differentiation and Permutation of Equations

For notational convenience, we shall rewrite (1.1) as an operator equation

$$\mathcal{F}\boldsymbol{x} = 0. \tag{2.1}$$

The dependent variables may appear algebraically or differentiated up to q times. We assume that for some $p \geq q$ the $\mathbb{R}^n$-valued function $\boldsymbol{x} \in \mathcal{C}^p$. Similarly, $\mathcal{F}\boldsymbol{x} \in \mathcal{C}^{p-q}$ is an $\mathbb{R}^n$-valued function.

Let $D = d/dt$ denote the differentiation operator, and let $\nu \in \mathbb{N}^n$ denote a multi-index $\nu = (\nu_1, \nu_2, \ldots, \nu_n)^T$. Then we define $D^\nu = \mathrm{diag}(D^{\nu_1}, \ldots, D^{\nu_n})$. We let $\mu(\mathcal{F}) \in \mathbb{N}^n$ denote a multi-index such that $D^{\mu(\mathcal{F})}\boldsymbol{x}$ are the highest order derivatives appearing in the DAE, i.e. $x_j^{(\mu_j)}$ is the highest order derivative of x_j that appears in $\mathcal{F}\boldsymbol{x} = 0$.

Structural properties and permutations

Following Brenan et al. (1989, p. 21), we call a property of a matrix a *generic* or *structural property* if it holds a.e. in a neighborhood of the particular values of the *nonzero entries* of the matrix. A matrix A is structurally nonsingular if and only if there exists a permutation P_1 such that $P_1 A$ has a nonzero diagonal, often referred to as a *maximum transversal* or an *output set*. A structurally singular matrix is also singular, but the converse is not true. Similar definitions for a nonlinear system $g(v) = 0$ are obtained by requiring that the system's structural Jacobian can be permuted to obtain a nonzero diagonal. Likewise, we call the DAE problem (1.1) structurally nonsingular if there is an output set when we consider $\boldsymbol{x}$ to be unknown and make no difference between algebraic and differentiated appearances of $\boldsymbol{x}$, i.e. differentiation is regarded as an algebraic operation.

We shall make use of Block Lower Triangular (BLT) partitioning in order to decompose a problem into subproblems. By means of a simultaneous row and column permutation of $P_1 A$ the matrix is transformed into $Q^T P_1 A Q = PAQ$, a BLT matrix with the same nonzero diagonal elements as $P_1 A$. Output assignment and BLT partitioning are standard techniques in sparse matrix analysis, cf. Duff et al. (1986).

Differentiations

For a structurally nonsingular DAE, $\mathcal{F}\boldsymbol{x} = 0$, it is always possible to find a differentiated problem $\mathcal{G}\boldsymbol{x} = D^\nu \mathcal{F}\boldsymbol{x} = 0$ with ν finite, such that the differentiated problem is structurally nonsingular with respect to its highest order derivatives $D^{\mu(\mathcal{G})}\boldsymbol{x}$. Pantelides's algorithm, cf. Pantelides (1988), intended for finding consistent initial values for a DAE, establishes the minimum number of times each equation has to be differentiated, i.e. it finds the minimal $\nu(\mathcal{F})$. His algorithm also constructs the output set of interest automatically, and in the structural analysis step we need only construct the desired BLT partition.

By definition the problem $\mathcal{F}\boldsymbol{x} = 0$ is index 0, if it uniquely determines the highest order derivatives $D^{\mu(\mathcal{F})}\boldsymbol{x}$, with all $\mu_j(\mathcal{F}) > 0$, as continuous functions of t and lower derivatives. If the same condition holds with some $\mu_j(\mathcal{F}) = 0$, it is index 1. The fact that it is not possible to solve uniquely for the highest order derivatives does not imply that the index is greater than one: consider the simple index 1 problem

$$\begin{aligned} \dot{x} + \dot{y} &= 1 \\ x - y &= 0. \end{aligned}$$

Thus, Pantelides's algorithm may call for an unnecessary differentiation step (here it would suggest differentiating the second equation once, resulting in an index 0 system) but it is simple to remove such superfluous differentiations. Whether this should be done or not depends on if one prefers to be able to solve for the highest order derivatives or to reduce the index to one while introducing as few new variables as possible.

In our algorithm below, we must be able to solve for the highest order derivatives. Should a singularity be detected (in which case the original problem could actually be singular), we use the differentiated equations to manipulate some of the original equations and restart the reduction procedure for the modified problem.

The index reduction procedure for $\mathcal{F}\boldsymbol{x} = 0$ consists of the following steps:

Differentiation, Permutation and Index Reduction Algorithm

1. If the problem is structurally singular then return an error message.
2. *Differentiation.* Use Pantelides's algorithm to obtain
   a. a vector $\nu(\mathcal{F}) \in \mathbb{N}^n$,
   b. a differentiated problem $\mathcal{G}\boldsymbol{x} = \mathcal{F}^\nu \boldsymbol{x} = 0$ with $\mathcal{F}^\nu = D^{\nu(\mathcal{F})}\mathcal{F}$
   c. an output set for $\mathcal{F}^\nu \boldsymbol{x} = 0$ with respect to its highest order derivatives $D^{\mu(\mathcal{G})}\boldsymbol{x}$ as unknowns.
3. *Permutation.* Since $P\mathcal{F}^\nu Q = PD^{\nu(\mathcal{F})}P^T P\mathcal{F}Q = D^{P\nu(\mathcal{F})}P\mathcal{F}Q$, BLT partitioning with respect to unknowns $D^{\mu(\mathcal{G})}\boldsymbol{x}$ yields the permuted
   a. undifferentiated problem $\mathcal{H}\boldsymbol{y} = 0$ with $\mathcal{H} = P\mathcal{F}Q$ and $\boldsymbol{y} = Q^T\boldsymbol{x}$.
   b. differentiated problem $\mathcal{H}^{P\nu}\boldsymbol{y} = 0$ with $\mathcal{H}^{P\nu} = P\mathcal{F}^\nu Q$, i.e. $\nu(\mathcal{H}) = P\nu(\mathcal{F})$, and this problem is BLT with respect to its highest order derivatives $D^{Q^T\mu(\mathcal{G})}\boldsymbol{y}$.
4. *Index Reduction.* Select derivatives to be replaced by dummy derivatives, blockwise as indicated by the BLT partition. If unable to select one dummy for each differentiated equation, manipulate original equations using information from differentiated equations and restart at step 1.

We may without loss of generality view our original problem $\mathcal{F}\boldsymbol{x} = 0$ as correctly permuted initially. In the following section we can therefore concentrate on the central index reduction step, i.e. how to select dummy derivatives.

3. Selection of Dummy Derivatives

Consider the differentiated problem $\mathcal{G}\boldsymbol{x} = \mathcal{F}^\nu \boldsymbol{x} = 0$, in BLT form. For notational simplicity, let $g_i = 0$ represent the i:th block of $\mathcal{G}\boldsymbol{x} = 0$, let z_i denote the vector of highest order derivatives of the unknowns associated with that block, and note that $\dim z_i = \dim g_i$. Let $f_i = 0$ be the corresponding block of the original problem $\mathcal{F}\boldsymbol{x} = 0$. We make the following assumptions:

A1. $\mathcal{G}\boldsymbol{x} = \mathcal{F}^\nu \boldsymbol{x} = 0$ is in BLT form.

A2. The equations in each block have been sorted in descending order with respect to number of differentiations, i.e. $\nu_1(g_i) \geq \nu_2(g_i) \geq \ldots$

A3. The Jacobian $\partial g_i/\partial z_i$ evaluated at the actual point has full rank (the singular case will be treated later).

To obtain an equivalent index 1 problem, the reduction algorithm constructs a sequence of $\nu_1(g_i)+1$ problems indexed by a superscript $[j]$:

Index Reduction Algorithm

Step 1. Initialize:
$$z_i^{[1]} \leftarrow z_i,$$
$$g_i^{[1]}(z_i^{[1]}) \leftarrow g_i(z_i),$$
$$G_i^{[1]} = \partial g_i^{[1]}/\partial z_i^{[1]} \leftarrow \partial g_i/\partial z_i$$
$$j \leftarrow 1.$$

Step 2. If $g_i^{[j]} = 0$ has no differentiated equations, go to Step 6.

Step 3. If $g_i^{[j]} = 0$ has m differentiated equations, let $h_i^{[j]} = 0$ denote its m first equations. The Jacobian $H_i^{[j]} = \partial h_i^{[j]}/\partial z_i^{[j]}$ then equals the first m rows of $G_i^{[j]}$, due to the sorting with respect to number of differentiations.

Step 4. Next, select m columns $l_1, \ldots, l_m$ of $H_i^{[j]}$ to make a square nonsingular matrix $M_i^{[j]}$. Selected columns indicate derivatives to be replaced: from the m equations $h_i^{[j]} = 0$ we select the m components of $\hat{z}_i^{[j]} = (\, z_{i,l_1}^{[j]} \quad \cdots \quad z_{i,l_m}^{[j]} \,)^T$ to be replaced (in Step 6) by dummy derivatives.

Step 5. Now determine how to use the *predecessor* of $h_i^{[j]} = 0$, denoted by $D^{-1}h_i^{[j]} = 0$, thus omitting the last differentiation. New candidates $z_i^{[j+1]}$ for possible replacement are $D^{-1}\hat{z}_i^{[j]}$, i.e. derivatives of one order less than those selected in Step 4. The components of $\hat{z}_i^{[j]}$ are all highest order derivatives in $h_i^{[j]} = 0$ and differentiated at least once. Hence they represent derivatives of the original unknowns $\boldsymbol{x}$, implying that $D^{-1}\hat{z}_i^{[j]}$ is well defined. The Jacobian $\partial g_i^{[j+1]}/\partial z_i^{[j+1]} = M_i^{[j]}$, as will be shown later.

Thus, set
$$g_i^{[j+1]} \leftarrow D^{-1}h_i^{[j]},$$
$$z_i^{[j+1]} \leftarrow D^{-1}\hat{z}_i^{[j]},$$
$$G_i^{[j+1]} \leftarrow M_i^{[j]},$$
$$j \leftarrow j+1,$$
and repeat from Step 2.

Step 6. Let $k_i = j$. We now obtain an index one formulation of $f_i = 0$ by collecting all original as well as all differentiated equations:

$$\begin{pmatrix} g_i^{[k_i]} \\ \vdots \\ g_i^{[1]} \end{pmatrix} = 0.$$

In all equations, introduce a unique *dummy derivative* for each derivative selected in Step 4 (i.e. the variables given by $\hat{z}_i^{[k_i]}, \ldots, \hat{z}_i^{[1]}$), to replace that derivative wherever it occurs. The system now consists of the original equations $f_i = 0$ and the sequence of differentiated equations leading to $g_i = 0$. The unknowns are the original ones and the newly introduced all-algebraic dummy derivatives. □

Before proving that the algorithm converts $f_i = 0$ to an equivalent index 1 formulation, let us first consider its application to a simple linear problem.

EXAMPLE 1
Let our problem $\mathcal{F}\boldsymbol{x} = 0$ be defined as

$$x_1 + x_2 + u_1(t) = 0 \tag{3.1a}$$
$$x_1 + x_2 + x_3 + u_2(t) = 0 \tag{3.1b}$$
$$x_1 + x_4 + u_3(t) = 0 \tag{3.1c}$$
$$2\ddot{x}_1 + \ddot{x}_2 + \ddot{x}_3 + \dot{x}_4 + u_4(t) = 0 \tag{3.1d}$$

where the $u_m(t)$ are known forcing functions. The differentiated problem $\mathcal{G}\boldsymbol{x} = 0$ is

$$\ddot{x}_1 + \ddot{x}_2 + \ddot{u}_1(t) = 0 \tag{3.1a$''$}$$
$$\ddot{x}_1 + \ddot{x}_2 + \ddot{x}_3 + \ddot{u}_2(t) = 0 \tag{3.1b$''$}$$
$$\dot{x}_1 + \dot{x}_4 + \dot{u}_3(t) = 0 \tag{3.1c$'$}$$
$$2\ddot{x}_1 + \ddot{x}_2 + \ddot{x}_3 + \dot{x}_4 + u_4(t) = 0 \tag{3.1d}$$

which was obtained by differentiating (3.1a) and (3.1b) twice and (3.1c) once. The vector of highest order derivatives in $\mathcal{G}\boldsymbol{x} = 0$ is $z_1 = (\ \ddot{x}_1 \quad \ddot{x}_2 \quad \ddot{x}_3 \quad \dot{x}_4\ )^T$.

Block triangularization results in one block, $g_1(\boldsymbol{x}) = \mathcal{G}\boldsymbol{x}$, and the differentiated problem is index 1 since the Jacobian with respect to the highest order derivatives

$$\frac{\partial g_1}{\partial z_1} = G_1^{[1]} = \begin{matrix} \\ (\mathrm{a}'') \\ (\mathrm{b}'') \\ (\mathrm{c}') \\ (\mathrm{d}) \end{matrix} \begin{matrix} \begin{matrix} \ddot{x}_1 & \ddot{x}_2 & \ddot{x}_3 & \dot{x}_4 \end{matrix} \\ \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 2 & 1 & 1 & 1 \end{pmatrix} \end{matrix}$$

is nonsingular. This matrix is set up in Step 1. Throughout the steps of the index reduction, we indicate at the Jacobians which equations and variables are presently being considered.

We have three differentiated equations, and Step 3 gives

$$H_1^{[1]} = \begin{matrix} \\ (\mathrm{a}'') \\ (\mathrm{b}'') \\ (\mathrm{c}') \end{matrix} \begin{matrix} \begin{matrix} \ddot{x}_1 & \ddot{x}_2 & \ddot{x}_3 & \dot{x}_4 \end{matrix} \\ \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{matrix}$$

At Step 4 we have two possibilities to select a nonsingular submatrix of $H_1^{[1]}$; columns 1, 3 and 4 or columns 2, 3 and 4. Let us take the first alternative:

$$M_1^{[1]} = \begin{matrix} \\ (\mathrm{a}'') \\ (\mathrm{b}'') \\ (\mathrm{c}') \end{matrix} \begin{matrix} \begin{matrix} \ddot{x}_1 & \ddot{x}_3 & \dot{x}_4 \end{matrix} \\ \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \end{matrix}$$

Thus we use (3.1a$''$), (3.1b$''$) and (3.1c$'$) to replace the derivatives $\ddot{x}_1$, $\ddot{x}_3$ and $\dot{x}_4$. Then, at Step 5 we prepare for the next cycle and consider the predecessor of the present subproblem, omitting one differentiation:

$$G_1^{[2]} = \begin{matrix} \\ (\mathrm{a}') \\ (\mathrm{b}') \\ (\mathrm{c}) \end{matrix} \begin{matrix} \begin{matrix} \dot{x}_1 & \dot{x}_3 & x_4 \end{matrix} \\ \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \end{matrix}$$

Now $j = 2$ and we repeat from Step 2. Since we still have differentiated equations, Step 3 yields

$$H_1^{[2]} = \begin{matrix} \\ (\mathrm{a}') \\ (\mathrm{b}') \end{matrix} \begin{matrix} \dot{x}_1 & \dot{x}_3 & x_4 \end{matrix} \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \end{pmatrix}$$

At Step 4 we have to select the two first columns, and we obtain

$$M_1^{[2]} = \begin{matrix} \\ (\mathrm{a}') \\ (\mathrm{b}') \end{matrix} \begin{matrix} \dot{x}_1 & \dot{x}_3 \end{matrix} \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$$

Thus, we have to use (3.1a′) and (3.1b′) to replace $\dot{x}_1$ and $\dot{x}_3$. Step 5 therefore results in

$$G_1^{[3]} = \begin{matrix} \\ (\mathrm{a}) \\ (\mathrm{b}) \end{matrix} \begin{matrix} x_1 & x_3 \end{matrix} \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$$

Since there are no differentiated equations, we go to Step 6 and collect the pieces. To get a simple and clear notation, we let x_1'' denote the dummy derivative that is substituted for $\ddot{x}_1$, and similarly for other derivatives.

$$x_1 + x_2 + u_1(t) = 0 \tag{3.2a}$$
$$x_1 + x_2 + x_3 + u_2(t) = 0 \tag{3.2b}$$
$$x_1' + \dot{x}_2 + \dot{u}_1(t) = 0 \tag{3.2a′}$$
$$x_1' + \dot{x}_2 + x_3' + \dot{u}_2(t) = 0 \tag{3.2b′}$$
$$x_1 + x_4 + u_3(t) = 0 \tag{3.2c}$$
$$x_1'' + \ddot{x}_2 + \ddot{u}_1(t) = 0 \tag{3.2a″}$$
$$x_1'' + \ddot{x}_2 + x_3'' + \ddot{u}_2(t) = 0 \tag{3.2b″}$$
$$x_1' + x_4' + \dot{u}_3(t) = 0 \tag{3.2c′}$$
$$2x_1'' + \ddot{x}_2 + x_3'' + x_4' + u_4(t) = 0 \tag{3.2d}$$

The nine unknown variables are x_1, x_3, x_1', x_3', x_4, x_1'', x_3'', x_4', and x_2 which all are algebraic except for x_2 which appears differentiated twice.

The problem is now index 1, with the equations ordered as in Step 6. It is BLT with respect to its highest order derivatives. The first block consists of the equations from the last loop of the selection procedure which gave (3.2a) and (3.2b) from which we can solve for x_1 and x_3. The second block consists of the equations from the second last loop which gave (3.2a′), (3.2b′) and (3.2c) from which we can solve x_1', x_3' and x_4. The third and last block consists of the equations from the first loop which gave (3.2a″), (3.2b″), (3.2c′) and (3.2d) from which we can solve x_1'', $\ddot{x}_2$, x_3'' and x_4'. □

Proof of index reduction

We shall prove the crucial parts of Steps 4, 5 and 6.

Step 4: First we will prove by induction that one can select a nonsingular matrix $M_i^{[j]}$ at Step 4. By assumption A3, $G_i^{[1]}$ has full rank. If $G_i^{[j]}$ has full rank and there are differentiated equations, then $H_i^{[j]}$ has full rank. Hence it is possible to select a square nonsingular submatrix $M_i^{[j]}$. Defining $G_i^{[j+1]} = M_i^{[j]}$

then implies that $G_i^{[j+1]}$ has full rank. By induction it follows that it is possible to select a full rank matrix $M_i^{[j]}$ each time we arrive at Step 4.

Step 5: Next, we show the formula $G_i^{[j+1]} = M_i^{[j]}$ at Step 5. It implies that $G_i^{[j]} = \partial g_i^{[j]}/\partial z_i^{[j]}$. Consider the element p, q of $G_i^{[j+1]}$. We have

$$G_{i,pq}^{[j+1]} = \partial g_{i,p}^{[j+1]}/\partial z_{i,q}^{[j+1]} = \partial D^{-1} g_{i,p}^{[j]}/\partial D^{-1} z_{i,l_q}^{[j]},$$

and for the same element of $M_i^{[j]}$, we have

$$M_{i,pq}^{[j]} = \partial h_{i,p}^{[j]}/\partial \hat{z}_{i,q}^{[j]} = \partial g_{i,p}^{[j]}/\partial z_{i,l_q}^{[j]}.$$

The formula $G_i^{[j+1]} = M_i^{[j]}$ now follows directly from the fact that for any differentiable function $e(t, v)$ it holds $\partial e(t, v)/\partial v = \partial \dot{e}(t, v, \dot{v})/\partial \dot{v}$.

Step 6: We shall now show that the problem generated by the algorithm is index 1. We first consider the reduction of a block. The equations are $g_i^\star(z_i^\star) = 0$ with

$$g_i^\star = \begin{pmatrix} g_i^{[k_i]} \\ \vdots \\ g_i^{[1]} \end{pmatrix}, \quad z_i^\star = \begin{pmatrix} z_i^{[k_i]} \\ \vdots \\ z_i^{[1]} \end{pmatrix}.$$

Considered as an algebraic problem, $g_i^\star(z_i^\star) = 0$ is BLT, where we can take the jth block to consist of the equations $g_i^{[j]} = 0$ with the unknowns $z_i^{[j]}$. Since the Jacobian $\partial g_i^{[j]}/\partial z_i^{[j]} = G_i^{[j]}$ is nonsingular, we can solve for all the unknowns $z^\star$. The variables of $z_i^{[j]}$ are of two categories. First, $z_i^{[j]}$ contains the variables $\hat{z}_i^{[j]}$ selected for replacement in Step 5 of the jth loop. The remaining variables clearly represent highest order derivatives of the old variables x in the problem generated by the algorithm. Thus $z^\star$ contains all the highest order derivatives of the problem generated by the algorithm, and since we can solve for them the problem is at most index 1.

Finally consider the complete problem. Sort the block subsystems $g_i^{[j]} = 0$ with respect to descending order of the index j, and with respect to descending order of the index i. The unknowns $z_i^{[j]} = 0$ are sorted similarly. The resulting problem is then BLT with nonsingular blocks. Consequently, the complete problem is at most index 1. □

The original problem and the problem resulting from the algorithm are mathematically equivalent in the sense that they have identical solution sets in the original (undifferentiated) dependent variables x. Original algebraic equations are still explicitly present. The advantage of the index reduction technique proposed here is that it excludes some derivatives from discretization; by treating the dummy derivatives as algebraic variables the problem of inconsistency due to discretization is eliminated.

Removing superfluous differentiations

As was mentioned in Section 2, Pantelides's algorithm differentiates some problems to index 0 and others to index 1. If the differentiated problem is index 0, the number of dummy derivatives introduced can be decreased by reintroducing some of them as ordinary derivatives. This also reduces the total number of equations accordingly. The procedure is to modify Step 6 in the following way:

1. Do not include the equations of $g_i^{[1]} = 0$ which are differentiated.
2. Do not replace the corresponding selected components $\hat{z}_i^{[1]}$ with dummy derivatives, but replace them with the first order derivative of the representatives for their predecessors, $D^{-1}\hat{z}_i^{[1]}$, some of which may be dummy derivatives.

EXAMPLE 2
Let us again consider the problem discussed in Example 1. The differentiated problem (3.1) is index 0, since it contains no algebraic variables. This implies that we can obtain a smaller index 1 formulation by disregarding the most differentiated equations, i.e. (3.2a″), (3.2b″) and (3.2c′). The corresponding highest order derivatives $\ddot{x}_1$, $\ddot{x}_3$ and $\dot{x}_4$ are not replaced by dummy derivatives, but with dx_1'/dt, dx_3'/dt and $\dot{x}_4$, respectively. The problem then becomes

$$x_1 + x_2 + u_1(t) = 0 \tag{3.3a}$$
$$x_1 + x_2 + x_3 + u_2(t) = 0 \tag{3.3b}$$
$$x_1' + \dot{x}_2 + \dot{u}_1(t) = 0 \tag{3.3a'}$$
$$x_1' + \dot{x}_2 + x_3' + \dot{u}_2(t) = 0 \tag{3.3b'}$$
$$x_1 + x_4 + u_3(t) = 0 \tag{3.3c}$$
$$2\dot{x}_1' + \ddot{x}_2 + \dot{x}_3' + \dot{x}_4 + u_4(t) = 0. \tag{3.3d}$$

The six unknowns are x_1, x_3, x_1', x_3', x_4 and x_2, which all are dynamic except for the two algebraic variables x_1 and x_3. The problem is index 1, since by differentiating all but the last equation once, we obtain an index 0 problem.
□

This technique may also be applied to some problems that Pantelides's algorithm differentiates to index 1, viz. if the differentiated problem contains index 0 subproblems. Such blocks may then be treated as in Example 2.

4. Pivoting of Dummy Derivatives

The algorithm described above assumes that the Jacobian with respect to the highest order derivatives of the differentiated problem $\mathcal{F}^\nu \boldsymbol{x} = 0$ is nonsingular. If the Jacobian is singular, a more detailed analysis is required. It should be noted that the index reduction algorithm given above transforms the original system *locally* to index 1. Clearly, it could happen that the Jacobian becomes singular along the solution trajectory. Such events may imply that the solution has a turning point, in which case the DAE model may be inappropriate to account for the future evolution of the system. But turning points are not the only possibility. The singularity may be due to a (locally) inappropriate selection of dummy derivatives. Therefore, this selection must in general be dynamic, i.e. we must be prepared to "pivot" the selection of dummy derivatives.

EXAMPLE 3—The inevitable pendulum
Consider a planar pendulum of length L and mass m. In Cartesian coordinates the equations of motion are

$$x^2 + y^2 - L^2 = 0 \tag{4.1a}$$
$$m\ddot{x} + (\lambda/L)x = 0 \tag{4.1b}$$
$$m\ddot{y} + (\lambda/L)y + mg = 0, \tag{4.1c}$$

where g is the gravitational constant and λ is the force in the string; the problem is index 3.

Applying Pantelides's algorithm, the length constraint is differentiated twice:

$$x^2 + y^2 - L^2 = 0 \tag{4.1a}$$
$$2x\dot{x} + 2y\dot{y} = 0 \tag{4.1a'}$$
$$2x\ddot{x} + 2\dot{x}^2 + 2y\ddot{y} + 2\dot{y}^2 = 0. \tag{4.1a''}$$

Thus, the differentiated problem is

$$2x\ddot{x} + 2\dot{x}^2 + 2y\ddot{y} + 2\dot{y}^2 = 0 \tag{4.1a''}$$
$$m\ddot{x} + (\lambda/L)x = 0 \tag{4.1b}$$
$$m\ddot{y} + (\lambda/L)y + mg = 0. \tag{4.1c}$$

The Jacobian J of the differentiated problem with respect to the highest order derivatives $(\ \ddot{x} \quad \ddot{y} \quad \lambda\ )^T$ is

$$J = \begin{pmatrix} 2x & 2y & 0 \\ m & 0 & x/L \\ 0 & m & y/L \end{pmatrix}.$$

Since $\det(J) = -2m(x^2 + y^2)/L = -2mL$, the differentiated problem is index 1 if and only if $m \neq 0$ and $L \neq 0$. Assume that this is the case. The index reduction algorithm gives

$$H_1^{[1]} = \begin{array}{c} \\ (\mathrm{a}'') \end{array} \begin{array}{ccc} \ddot{x} & \ddot{y} & \lambda \\ (2x & 2y & 0\ ). \end{array}$$

Neither x nor y is necessarily non-zero for all times. Due to the length constraint, however, they are not simultaneously zero. We may therefore attempt to obtain a "well-conditioned" $M_1^{[1]}$ by choosing $M_1^{[1]} = (2x)$ when $|x| > |y|$ and $M_1^{[1]} = (2y)$ otherwise.

Consider first the case $|x| > |y|$. Selecting $F_1^{[1]} = (2x)$ implies that the differentiated length constraint will be used to replace $\ddot{x}$ and $\dot{x}$. The problem is then transformed to

$$x^2 + y^2 - L^2 = 0 \tag{4.2a}$$
$$2xx' + 2y\dot{y} = 0 \tag{4.2a'}$$
$$2xx'' + 2x'^2 + 2y\ddot{y} + 2\dot{y}^2 = 0 \tag{4.2a''}$$
$$mx'' + (\lambda/L)x = 0 \tag{4.2b}$$
$$m\ddot{y} + (\lambda/L)y + mg = 0. \tag{4.2c}$$

When $|x| \leq |y|$, we select $F_1^{[1]} = (2y)$, implying that the differentiated length constraint will be used to replace $\ddot{y}$ and $\dot{y}$. Then we obtain the index 1 problem

$$x^2 + y^2 - L^2 = 0 \tag{4.3a}$$
$$2x\dot{x} + 2yy' = 0 \tag{4.3a'}$$
$$2x\ddot{x} + 2\dot{x}^2 + 2yy'' + 2y'^2 = 0 \tag{4.3a''}$$
$$m\ddot{x} + (\lambda/L)x = 0 \tag{4.3b}$$
$$my'' + (\lambda/L)y + mg = 0. \tag{4.3c}$$

Changing the decision whether the equations (4.1a″) and (4.1a′) should be used to replace $\ddot{x}$ and $\dot{x}$ or $\ddot{y}$ and $\dot{y}$, can be interpreted as a pivoting operation, and we refer to it at as *dummy pivoting*. From this example, it can be seen that dummy pivoting corresponds to a (locally necessary) change of mathematical models, cf. Leimkuhler (1989). Switching from one model to the other is a simple operation, since in the augmented index 1 problem we solve also for the dummy derivatives. Therefore, the initial values necessary to continue the numerical integration with the new model are readily available. □

A very simple first approach to handle dummy pivoting is to make a selection at the start and use the resulting equations as long as the numerical DAE solver is able to function properly, and pivot only when necessary. In general, however, it is preferable to pivot so that the matrices $M_i^{[j]}$ remain well-conditioned. It is therefore desirable to have a rather close integration of the index reduction method and the numerical DAE solver. This suggests that index reduction computations should be incorporated into the solver, rather than being considered as a separate preprocessing tool.

EXAMPLE 4—Lagrangian equations of motion
The Lagrangian equations of the first kind for a constrained mechanical system are, written as a second order equation,

$$M(p)\ddot{p} = F(p,\dot{p}) - G^T(p)\lambda \tag{4.4a}$$
$$0 = g(p), \tag{4.4b}$$

where p is an n-vector representing the system's position, M is the nonsingular mass matrix, F applied forces, and λ is the m-vector of Lagrange multipliers associated with the m constraints (4.4b), assumed to have a full-rank constraint matrix $G(p) = \partial g/\partial p$. Thus the system has $n - m$ degrees of freedom, and the system of equations (4.4) is index 3. Two differentiations of the constraint equation results in

$$M(p)\ddot{p} = F(p,\dot{p}) - G^T(p)\lambda \tag{4.4a}$$
$$0 = g(p) \tag{4.4b}$$
$$0 = G\dot{p} \tag{4.4b′}$$
$$0 = \dot{G}\dot{p} + G\ddot{p}. \tag{4.4b″}$$

Here $\dot{G} = G'\dot{p}$. We can solve for the highest order derivatives $\ddot{p}$ and λ, since the regularity assumptions imply that $GM^{-1}G^T$ is regular. Selecting dummy derivatives implies choosing m dummies among the n variables $\dot{p}$, and m among $\ddot{p}$, using equations (4.4b′) and (4.4b″). Then we have a total of $n + 3m$ equations, $n + m$ original variables p and λ, and $2m$ dummy derivatives (selected elements of $\dot{p}$ and $\ddot{p}$).

It is also instructive to apply the index reduction algorithm to the corresponding first order system

$$\dot{p} = v \tag{4.5a}$$
$$M(p)\dot{v} = F(p,v) - G^T(p)\lambda \tag{4.5b}$$
$$0 = g(p), \tag{4.5c}$$

where the n-vector v represents velocity. Applying Pantelides's algorithm now requires also that (4.5a) is differentiated once, and as before, the resulting differentiated problem is second order. Apart from the same dummies as

those selected in (4.4), one will have to select n dummy derivatives $v' = \dot{v}$. However, this variable merely accounts for a trivial action of substitution and can immediately be eliminated. If a first order system is required, it can easily be obtained after a few more similar operations; it is identical to the first order system that can be obtained from (4.4). □

Let us finally comment the singular case. Then, if the index of the differentiated problem is greater than one, it is possible to derive an equation in which none of the unknown highest order derivatives appears. However, since we need the whole sequence of differentiated equations, we shall manipulate the original undifferentiated system so that the corresponding differentiated problem has the desired structure with a nonsingular Jacobian.

EXAMPLE 5—A singular system
We consider the following problem, Brenan et al. (1989, p. 23):

$$\dot{x} + t\dot{y} = f_1 \tag{4.6a}$$
$$x + ty = f_2. \tag{4.6b}$$

The problem is solvable and the unique solution is $x = f_2 - t(\dot{f}_2 - f_1)$, $y = \dot{f}_2 - f_1$. Pantelides's algorithm differentiates the second equation once and yields

$$\dot{x} + t\dot{y} = f_1 \tag{4.6a}$$
$$\dot{x} + t\dot{y} + y = \dot{f}_2. \tag{4.6b$'$}$$

It is not index 1, since it is singular in $\dot{x}$ and $\dot{y}$.

The approach is now to consider the most differentiated equations and use them and their predecessors to eliminate variables in the other original equations. Equation (4.6b$'$) depends on $\dot{x}$ and $\dot{y}$. Try to solve (4.6b) for x or y. Solving for x, we obtain $x = f_2 - ty$ and $\dot{x} = \dot{f}_2 - y - t\dot{y}$. Substitution into (4.6a) yields $y = \dot{f}_2 - f_1$, in which neither $\dot{x}$ nor $\dot{y}$ appears. Let this equation replace (4.6a) and consider

$$y = \dot{f}_2 - f_1 \tag{4.7a}$$
$$x + ty = f_2 \tag{4.7b}$$

as the new original problem. The modified problem is then simpler than the original problem. In this particular case, it is algebraic, hence index 1, and can be solved immediately. □

The example above demonstrates a more difficult type of index reduction. In contrast to the operations performed by the index reduction algorithm of Section 3, problems having a rank-deficient Jacobian with respect to the highest order derivatives will also require that equations are solved symbolically or semi-numerically.

5. Numerical Results

We shall now present numerical results for the pendulum problem of Example 3 using the well-known solver DASSL. Brenan et al. (1989, pp. 150–157) discusses the numerical solution for various formulations. The evolution of the pendulum equations is computed for a short time interval covering less than a full period. Here we are concerned with the stability and numerical

drift of constraints and invariants, thus requiring simulations long enough to reveal secular effects; in all cases the problem was run for well over a hundred periods. In all the numerical results we use $g = 1$, $m = 1$ and $L = 1$.

In the discussion below, we follow the convention in Brenan et al. (1989) that the positive direction of the gravitational force is opposite to the direction of the y-axis in the Cartesian coordinate system. Note, however, that for their numerical results to be correct, op. cit. pp. 155–157, we must assume that they have accidentally switched the direction of the y-axis, cf. the obvious misprint in the state-space equation (6.2.5) on p. 151.

The state-space form of the planar pendulum equations is

$$\ddot{\phi} + \frac{g}{L}\sin\phi = 0 \tag{5.1}$$

where $x = L\sin\phi$ and $y = -L\cos\phi$. This model will be used as a reference model for comparing the numerical results.

Besides studying how well the length constraint is preserved in the numerical solutions, it is also of a great interest to study the invariant total energy, E, which is the sum of the kinetic energy and the potential energy. The "rest" state ($\phi = 0$ and $\dot{\phi} = 0$) is taken as the reference level with $E = 0$. This gives

$$E = \frac{m}{2}L^2\dot{\phi}^2 + mgL(1 - \cos\phi).$$

A numerical solution should preserve the energy and keep E constant within numerical accuracy. The expression for the energy can also be used to calculate the amplitude ϕ_M of the oscillations.

We shall study two cases:

1. Small oscillations with $\phi(0) = 0.1$ and $\dot{\phi}(0) = 0.0$ which gives the amplitude $\phi_M = 0.1$, period time $T \approx 6.29$ and energy $E = 1 - \cos 0.1$.
2. Large oscillations with $\phi(0) = \pi/2$ and $\dot{\phi}(0) = -1.0$ (Brenan et al. (1989) use these initial values) which gives the amplitude $\phi_M = 2\pi/3$, period time $T \approx 8.63$ and energy $E = 1.5$.

The first case is almost linear, whereas the second, with its large amplitude, is strongly nonlinear and will require four dummy pivotings per full period. Whenever DAEs are solved below, the initial conditions are taken to be consistent.

The two problems will be solved for the following formulations:

a. State-space reference model (5.1).

b. Differentiated index 1 model (4.1a″, b, c).

c. Dummy derivative index 1 model (4.3), case 1, and (4.2–3) for case 2.

d. Stabilized constraint index 2 model (Brenan et al., 1989, pp. 154–155).

In all cases the problems were run for 1000 units of time corresponding to approximately 159 periods in case 1, and 116 periods in case 2. Tolerance levels were kept sharp: the parameters ATOL and RTOL in DASSL were taken to be 10^{-9} for all components of the solution except for the model *d*, which cannot be solved with such requirements unless a looser tolerance is used for the two Lagrange multipliers λ and μ. These tolerances were set to 10^{-2}, which effectively corresponds to excluding these algebraic variables from the error tests, cp. Brenan et al. (1989, p. 156). Numerical Jacobians were used.

The results for case 1 are displayed in Table 1. Apart from run statistics, we show the deviation ΔE in total energy at the end of the integration interval,

Table 1: Numerical results for small oscillations (case 1) with models a–d.

Model	steps	f-evals	Jac.	ΔE	E-drift	ΔL	L-drift
a	19323	38654	19	$-2.9 \cdot 10^{-7}$	linear	—	—
b	26451	56157	85	$3.3 \cdot 10^{-6}$	quad.	$-3.5 \cdot 10^{-6}$	quad.
c	27338	62167	1291	$-1.1 \cdot 10^{-7}$	linear	$\sim 10^{-11}$	none
d	26697	54774	337	$-1.5 \cdot 10^{-7}$	linear	$\sim 10^{-11}$	none

Table 2: Numerical results for large oscillations (case 2) with models a–d.

Model	steps	f-evals	Jac.	ΔE	E-drift	ΔL	L-drift
a	44267	88524	27	$-2.9 \cdot 10^{-6}$	linear	—	—
b	68933	222313	28997	$2.2 \cdot 10^{-3}$	quad.	$1.6 \cdot 10^{-3}$	quad.
c	108731	240161	4800	$7.9 \cdot 10^{-7}$	linear	$\sim 10^{-11}$	none
d	84087	203850	6545	$1.9 \cdot 10^{-5}$	linear	$\sim 10^{-10}$	none

and similarly the deviation ΔL in the length constraint. E-drift and L-drift refer to the drifts in E and L, respectively. Note that the length constraint is explicitly present in models *c* and *d*, but it is only an implicit invariant in model *b*. Thus, the latter model shows a drift, and the deviation in the table refers to the error at the end of the integration interval. For models *c* and *d* there is no drift and the error corresponds to typical deviations throughout the integration. The energy, on the other hand, is an implicit invariant in all four models, resulting in a quite regular drift. For model *b* the drift is quadratic in t, whereas the other models exhibit a linear drift. It is to be noted that the drift is less pronounced for models *c* and *d* than for the state-space model *a*.

Results for case 2 are shown in Table 2. Here model *c* yields the most accurate results, with an energy drift approximately four times smaller than for *a*, and 25 times smaller than for model *d*. From the efficiency point of view, the state-space model *a* is clearly preferable. However, given the rather significant difference in accuracy between models *c* and *d*, the modest performance disadvantage of model *c* seems to be of minor importance.

The results reported here are typical for the pendulum problem. A few other problems from applications in mechanics and electrical engineering have also been tried with good results using the dummy derivative technique.

6. Implementation Aspects and Conclusions

When the proposed index reduction technique is to be implemented there are two main possibilities. One could either perform all operations completely using symbolic computations in a preprocessing step, or one could employ automatic differentiation, Rall (1981), to obtain a closer integration of the index reduction process and the subsequent numerical treatment. The latter approach seems particularly attractive, since in the augmented system, function evaluations corresponding to differentiated equations must be performed using analytical derivatives that must be continually reevaluated. In addition, the

subsequent numerical solution could take advantage of analytical Jacobians. The choice of differentiation technique is probably the most important decision in an implementation of the reduction method.

A second important issue is the choice of the matrices $M_i^{[j]}$ by selecting m linearly independent columns from $H_i^{[j]}$. We should aim for well-conditioned matrices, and hence the columns should be selected carefully. An obvious approach is to use a Gram-Schmidt procedure to successively find a set of "maximally" linearly independent columns. However, this approach will depend on the choice of the initial column selected, and it will in general not be possible to find the optimal set.

As far as the practical issues of dummy pivoting is concerned, it is required that one monitors the condition of the matrices $M_i^{[j]}$. Dummy pivoting corresponds to replacing one or more columns of $M_i^{[j]}$ by new columns from $H_i^{[j]}$. It is as yet unclear how this can be carried out inexpensively in large systems. Most likely, it is more demanding to continually monitor the condition than to actually perform the pivoting operation. It should be noted that well-conditioned matrices $M_i^{[j]}$ are needed to ensure that the selected dummy derivatives cancel the exact amount of dynamics in the augmented DAE system, leaving only what corresponds to the dynamics of a state-space form. If the selection is rank-deficient, there is a risk that the numerical integration method gets stuck in a singular point. Such effects can readily be seen e.g. in the pendulum problem; if one inappropriately uses (4.2) for small amplitudes (i.e. when $|x| \ll |y|$) it may very well happen that the integration method cannot get past $x = 0$.

In many cases of practical interest, one will not need a complete automatic reduction procedure, but only the handling of dummy derivatives. Thus, the structure (4.4) is common to all systems described by the Lagrangian equations of the first kind. Therefore, it is sufficient in such applications to select dummy derivatives properly.

The merits of the index reduction technique proposed in this paper lie in the fact that the dummy derivatives are identified and excluded from discretization. As a result, one avoids "over-discretization" of the DAE, and the differentiations inherent in a high-index DAE are carried out analytically rather than numerically. Since the algebraic equations are still present in their original form, there will be no numerical drift away from the solution manifold of the DAE, thus eliminating the need for constraint stabilization. As for invariants that are implicit in both the DAE and the state-space ODE (e.g. energy), the drift is very similar in both formulations. To sum up, the numerical experiments show that, by using the dummy derivative formulation, an accuracy comparable to that of solving a state-space formulation of the problem is obtained. Thus, the technique can be considered a viable alternative not only to constraint stabilization but even to state-space formulations whenever the latter are difficult to obtain.

7. References

ALISHENAS, T. (1991):, Ph.D. thesis to appear, Royal Inst. of Tech., Stockholm, Sweden.

BAUMGARTE, J. (1972): "Stabilization of constraints and integrals of motion

in dynamical systems," *Comp. Meth. in Appl. Mech.*, **1**, 1–16.

BRENAN, K.E., S.L. CAMPBELL and L.R. PETZOLD (1989): *Numerical solution of initial-value problems in differential-algebraic equations*, North-Holland.

DUFF, I.S. and C.W. GEAR (1985): "Computing the structural index," Math. and Comp. Sci. Div. Tech. Memo. No 50, March 1985, Argonne National Laboratory, Univ. of Chicago.

DUFF, I.S., A.M. ERISMAN and J.K. REID (1986): *Direct methods for sparse matrices*, Clarendon Press, Oxford.

EICH, E., C. FÜHRER, B.J. LEIMKUHLER and S. REICH (1990): "Stabilization and projection methods for multibody dynamics," Research Report A281, Institute of Mathematics, Helsinki University of Technology, Espoo, Finland.

FÜHRER, C. and B.J. LEIMKUHLER (1990): "A new class of generalized inverses for the solution of discretized Euler-Lagrange equations," in E. Haug and R. Deyo (Eds.): *Real-time integration methods for mechanical system simulation*, NATO ASI Series Vol. F 69, Springer, pp. 143–154.

FÜHRER, C. and B.J. LEIMKUHLER (1991): "Numerical solution of differential-algebraic equations for constrained mechanical motion," *Numerische Mathematik*, **59**, 55–69.

GEAR, C.W. (1990): "Differential algebraic equations, indices, and integral algebraic equations," *SIAM J. Num. Anal.*, **27**, 1527–1534.

GEAR, C.W., G.K. GUPTA and B.J. LEIMKUHLER (1985): "Automatic integration of the Euler-Lagrange equations with constraints," *J Comp. Appl. Math.*, **12–13**, 77–90.

LEIMKUHLER, B.J. (1989): "Some notes on pertubations of differential-algebraic equations," Research Report A266, Institute of Mathematics, Helsinki University of Technology, Espoo, Finland.

HAIRER, E., C. LUBICH and M. ROCHE (1989): *The numerical solution of differential-algebraic systems by Runge-Kutta methods*, Lecture Notes in Mathematics, 1409, Springer.

MATTSSON, S.E. and G. SÖDERLIND (1990): "A New technique for the solution of high index differential-algebraic equations," The 1990 Conference on the Numerical Solution of Ordinary Differential Equations, Helsinki, Finland, June 18–22.

PANTELIDES, C.C. (1988): "The consistent initialization of differential-algebraic systems," *SIAM J. Sci. Stat. Comput.*, **9**, 2, 213–231.

RALL, L.B. (1981): *Automatic differentiation—Techniques and applications*, Lecture Notes in Computer Science, 120, Springer.