In *Proc. of the Third International Symposium on Mathematical Modelling*, Vienna, 2000, pp.

SUBSTITUTE EQUATIONS FOR INDEX REDUCTION AND DISCONTINUITY HANDLING

G. Fábián[1] **D.A. van Beek**[2] **J.E. Rooda**[3]
Eindhoven University of Technology, Department of Mechanical Engineering
P.O. Box 513, 5600 MB Eindhoven, The Netherlands
E-mail: {d.a.v.beek[2], g.fabian[1], j.e.rooda[3]}@tue.nl

Abstract Several techniques exist for index reduction and consistent initialization of higher index DAEs. Many such techniques change the original set of equations by differentiation, substitution, and/or introduction of new variables. This paper introduces substitute equations as a new language element. By means of a substitute equation, the value of a continuous variable or its time derivative can be specified by an expression. This expression is evaluated each time that the variable or its time derivative, respectively, is referenced in the model. The advantage of substitute equations is that they enable index reduction and consistent initialization of higher index DAEs without changing the original equations; no existing variables are removed and no new variables are introduced. Substitute equations can also be used to enable the use of general purpose numerical solvers for equations where one or more of the unknowns are discontinuous.

Differential algebraic equations

Many current equation based simulation languages use Differential Algebraic Equations (DAEs) to describe the continuous behaviour of the modelled physical system. DAEs are a set of differential equations with additional algebraic constraints in the form

$$\mathbf{f}(\dot{\mathbf{x}}, \mathbf{x}, \mathbf{y}, t) = \mathbf{0}, \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^n$ is the vector of *differential variables*, $\mathbf{y} \in \mathbb{R}^m$ is the vector of *algebraic variables*, $t \in \mathbb{R}$ is the independent variable and $\mathbf{f} \in \mathbb{R}^{2n+m+1} \to \mathbb{R}^{n+m}$ is the set of DAEs.

In DAEs, not all variables can be freely initialized. The initial values of variables $\mathbf{x}$, $\dot{\mathbf{x}}$ and $\mathbf{y}$, denoted by $\mathbf{x}(0), \dot{\mathbf{x}}(0), \mathbf{y}(0)$, must satisfy equation (1) at time 0:

$$\mathbf{f}(\dot{\mathbf{x}}(\mathbf{0}), \mathbf{x}(0), \mathbf{y}(0), 0) = \mathbf{0}. \tag{2}$$

In DAEs, in many cases, only the differential variables ($\mathbf{x}(0)$) are initialized. The initial values of the algebraic variables ($\mathbf{y}(0)$) and of the time derivatives of the differential variables ($\dot{\mathbf{x}}(\mathbf{0})$) are then calculated from (2).

DAEs are characterized by their (differential) *index* [5]. The index of equation (1) is *m*, if *m* is the smallest number such that the system of equations

$$\begin{aligned} \mathbf{f}(\dot{\mathbf{x}}, \mathbf{x}, \mathbf{y}, t) &= \mathbf{0}, \\ \frac{d\mathbf{f}(\dot{\mathbf{x}}, \mathbf{x}, \mathbf{y}, t)}{dt} &= \mathbf{0}, \\ &\vdots \\ \frac{d^{(m)}\mathbf{f}(\dot{\mathbf{x}}, \mathbf{x}, \mathbf{y}, t)}{d^{(m)}t} &= \mathbf{0}, \end{aligned} \tag{3}$$

can be transformed into an explicit ODE ($\dot{\mathbf{x}} = \mathbf{g}(\mathbf{x}, t)$) by algebraic manipulations. In general, the higher the index, the greater the numerical difficulty one is going to encounter when trying to solve the system numerically. For *higher index DAEs* (systems, with index greater than 1) there is no general purpose stable algorithm. A common feature of higher index DAEs is that there are hidden constraints in the DAEs. These are equations that further restrict the initialization of equation (1). Hidden constraints can be obtained after differentiation and algebraic manipulations [8]. The presence of hidden constraints mean that not all differential variables may be chosen freely; there are dependencies among differential variables. This can be seen, if equations in the form

$$\mathbf{g}(\mathbf{x}, \mathbf{u}, t) = \mathbf{0} \tag{4a}$$
$$\mathbf{u} = \mathbf{h}(t) \tag{4b}$$

are present in equation (1), or can be obtained after differentiations. The differential variables in (4a) are *dependent differential variables*. In fact, hidden constraints may be present in index 1 systems of DAEs too.

It is well known, that the mathematical models of several physical systems have a high differential index. The usual technique is that through differentiation and algebraic manipulations, the index is lowered to 0 or 1, and the resulting system is solved with available ODE or DAE solvers. In the literature, several algorithms can be found for index reduction. From these, the algorithm of Gear and Petzold [6], the constraint stabilization technique of Gear [5], and the algorithm of Bachmann et al. [1] all differentiate (parts of) the system of equations and use substitution. The algorithm of Pantelides [8] to reveal hidden constraints in DAEs, can also be used for index reduction. Furthermore, Mattsson describes an index reduction technique in [7], which uses dummy algebraic variables.

There are simulators, where (some of) the above mentioned index reduction techniques are implemented. After the model of a physical system has been specified, the equations are analyzed symbolically and the index is reduced by subsequent steps of differentiations and algebraic manipulations (substitution). In this way, the equations are changed. Different ways of index reduction may thus lead to different sets of variables that can be initialized. This may lead to a modelling problem. Since not all differential variables may be freely chosen in such a system, it must be clear for the modeller which ones may be initialized, and which ones are calculated from the equations. Even more, the modeller may want to choose himself the dependent differential variables that he wants to initialize.

In those simulators, where index reduction is not implemented, only low index systems (index 0 or 1) of equations may be entered. Therefore, the modeller has to perform index reduction, and has to re-formulate the system of equations. In this case, a new equation set is obtained that is usually less expressive than the original one. Also, variables may be eliminated from the equations due to index reduction. Therefore, each time the values of these variables are needed in the model, they must be re-calculated. This also reduces the readability of models.

To overcome the problems of the two approaches, in the χ language [10, 4] substitute equations are used.

Substitute equations

In the χ language, substitution can be specified explicitly by means of *substitute equations* in two forms. The simple form is

$$\begin{array}{lll} S & ::= & v \leftarrow E \mid v' \leftarrow E \\ E & ::= & e \end{array}$$

where S and E are nonterminals, v is a continuous variable, v' is the time derivative of a continuous variable and e is a numerical expression. This specification is equivalent to replacing all occurrences of v (v') by e in the model. The guarded form is

$$E \quad ::= \quad [\ b_1 \longrightarrow e_1 \ [] \ \ldots \ [] \ b_n \longrightarrow e_n]$$

where b_i is a boolean guard and e_i is a numerical expression ($i = 1 \ldots n$). In this case, variable v (v') is substituted dynamically, depending on the values of the guards. If b_i is true, variable v (v') is substituted by e_i. If more guards are true at the same time, one alternative is chosen nondeterministically and all occurrences of v (v') are calculated from this alternative.

All variables occurring in the right-hand-side of substitute equations (in expressions e, e_i and b_i, $i = 1 \ldots n$) must be well-defined, either by another substitute equation or by normal, non-substitute equations. Substitute equations are evaluated recursively; if a substituted variable occurs on the right-hand-side of a substitute equation, first, its value is re-calculated by substitution. Therefore, substitute equations can be specified in arbitrary order; the only requirement is that they may not contain circular dependencies. Variables defined in this way may not be assigned.

Substitution facilitates index reduction; it can also be used to reveal hidden constraints in index 1 DAEs and to model discontinuities. This is illustrated below.

Index reduction

As an example for a higher index DAE, take the following PID (proportional integral differential) controller. A horizontal force F is applied to a body of mass m on a flat surface, without friction. The position of the body is denoted by x. The control objective is to keep the body at a given position x_{set}. The unknowns are x, v, i, e, u. Variable F is an input variable (depending only on time), x_{set}, m, k_P, k_D and k_I are constants.

$$\dot{x} = v \tag{5a}$$
$$\dot{v} = \frac{F-u}{m} \tag{5b}$$
$$\dot{i} = e \tag{5c}$$
$$e = x - x_{set} \tag{5d}$$
$$u = k_P e + k_D \dot{e} + k_I i \tag{5e}$$

This is an index 2 system of DAEs. Differentiation of equation (5d) yields

$$\dot{e} = \dot{x}. \tag{6}$$

After differentiating equations (5a, 5e), and differentiating equation (6) a second time, and then substituting in $\dot{u} = k_P\dot{e} + k_D\ddot{e} + k_I\dot{i}$: v for $\dot{e}$, $\frac{F-u}{m}$ for $\ddot{e}$ ($\ddot{e} = \ddot{x} = \dot{v} = \frac{F-u}{m}$), and $x - x_{set}$ for $\dot{i}$; the ODE form can be obtained. Typically to higher index systems, equations (5) contain a hidden constraint: equation (6) must hold at time 0. This is because e and x are dependent differential variables, as can be seen from (5d). Therefore, only one of them can be initialized freely. Index reduction algorithms differ in the way they choose the variables to substitute. By choosing (5d) for differentiation and $\dot{e}$ for substitution, the system is specified in χ as follows

$$\begin{array}{ll} & x' = v \\ , & v' = (F-u)/m \\ , & e \;\; = x - x_{set} \\ , & i' \;\, = e \\ , & u \;\; = k_P e + k_D e' + k_I i. \\ , & e' \leftarrow x' \end{array}$$

Variable e is a so-called *prime substituted differential variable*. Variables of these category cannot be freely initialized. In this model, x can be freely initialized, but the value of e depends on x. The actual set of equations solved by numerical solvers obtained after substitution is

$$\dot{x} = v \tag{7a}$$
$$\dot{v} = \frac{F-u}{m} \tag{7b}$$
$$e = x - x_{set} \tag{7c}$$
$$\dot{i} = e \tag{7d}$$
$$u = k_P e + k_D \dot{x} + k_I i. \tag{7e}$$

Note that for the solvers, $\dot{e}$ is not present in the equations, e is thus an algebraic variable.

The advantage of using substitute equations is that the process of substitution is transparent; the original form of the equations is preserved and the additional information used ($\dot{e} = \dot{x}$) is made explicit. Also, references to the substituted variable in the discrete-event part of the model need not be altered; variable e is by definition a differential variable, and its time derivative can be referenced in any discrete statement.

The index of the example system of equations (5) can also be reduced by removing variable e from the equations. In this case, both e and $\dot{e}$ are calculated by substitution. The χ specification of the equations is as follows

$$\begin{array}{ll} & x' = v \\ , & v' = (F-u)/m \\ , & i' \;\, = e \\ , & u \;\; = k_P e + k_D e' + k_I i. \\ , & e \;\; \leftarrow x - x_{set} \\ , & e' \leftarrow x' \end{array}$$

In this case, variable e is a so-called *differential base-prime substituted* variable. Wherever e and $\dot{e}$ occur in the model they are substituted by the right-hand-side expressions of the respective substitute equations. The equation set actually solved by numerical solvers obtained by substitution is

$$
\begin{aligned}
\dot{x} &= v \\
\dot{v} &= \frac{F-u}{m} \\
\dot{i} &= x - x_{set} \\
u &= k_P(x - x_{set}) + k_D\dot{x} + k_I i.
\end{aligned}
$$

Variable e has disappeared from the actual equations. What is left are four equations in four unknowns. This is an index 1 problem. Again, in this χ specification only the value of x can freely be chosen, after which e is automatically initialized to $x - x_{set}$ via substitution. Also, the value of $\dot{e}$ is set automatically to $\dot{x}$.

Consistent initialization of index 1 systems

In the previous two approaches, the problems of higher index DAEs and hidden constraints in the equations have been solved simultaneously. In this section, the situation is addressed where only the problem of a hidden constraint is present. This is the case when the system of DAEs is of index 1 and there are dependent differential variables. Again, the solution is substitution.

The index of the PID controller example can also be reduced if variable u is replaced by a derivative of a dummy variable z. The set of equations now is

$$\dot{x} = v \tag{8a}$$

$$\dot{v} = \frac{F-\dot{z}}{m} \tag{8b}$$

$$e = x - x_{set} \tag{8c}$$

$$\dot{i} = e \tag{8d}$$

$$\dot{z} = k_P e + k_D\dot{e} + k_I i. \tag{8e}$$

This is an index 1 problem, because after differentiating equation (8c), the equations can be re-arranged into an ODE. Yet, e and x remain dependent differential variables so that they cannot be freely initialized. As a consequence, there is a hidden constraint in equation (8c), which appears after differentiation of the equation. The initialization problem can easily be solved in χ as before, by adding a substitution equation for $\dot{e}$ (or for both e and $\dot{e}$).

Modelling discontinuities

Another application area for substitute equations is the modelling of discontinuous functions. General purpose DAE and ODE solvers cannot usually integrate discontinuous functions [2]. The usual approach is that discontinuities are specified by so called *switching functions*. When the sign of the switching function changes, a discontinuity occurs. Integration stops, and is re-started again after the discontinuity. For more on numerical methods with respect to discontinuities we refer to [3].

A discontinuity in a variable that is used by the solver can be avoided in cases where the discontinuous variable can be expressed in a closed form. This variable can then be calculated by substitution, and thereby, it is removed from the equation set that is actually solved by numerical solvers. As an example, consider a tank described in [9], where overflow occurs if the level h of its contents reaches a maximum height h_{max}. The incoming and outgoing flows are denoted by Q_i and Q_o, respectively; the area of the tank by A, and the overflow by Q_x. The system described by a conditional equation is

$$
\begin{array}{l}
[\ h < h_{max} \vee Q_i < Q_o \longrightarrow Ah' = Q_i - Q_o,\ Q_x = 0 \\
[]\ h \geq h_{max} \wedge Q_i \geq Q_o \longrightarrow Ah' = 0,\ Q_x = Q_i - Q_o \\
]
\end{array}
$$

The general form of a conditional equation is: $[\, b_1 \longrightarrow DAEs_1\ []\ \ldots\ []\ b_n \longrightarrow DAEs_n\,]$, where $DAEs_i$ $(1 \leq i \leq n)$ represents one or more DAEs separated by commas. Boolean expression b_i denotes a guard. At any time, (at least) one of these guards must be open (true), so that the DAE(s) associated with the open guard (after the arrow of the open guard) is (are) activated. The discontinuous variable Q_x can be removed from the equations by substitution

$$
\begin{array}{ll}
 & Ah' = Q_i - Q_o - Q_x \\
, & Q_x \leftarrow [\ h < h_{max} \vee Q_i < Q_o \longrightarrow 0 \\
 & \quad [] \ h \geq h_{max} \wedge Q_i \geq Q_o \longrightarrow Q_i - Q_o \\
 & \quad]
\end{array}
$$

In this case, only the first equation, $Ah' = Q_i - Q_o - Q_x$ is solved by integration. The fact that variable h has a discontinuous first derivative is usually not a problem for numerical integrators.

Conclusions

Substitute equations make the mechanism of index reduction transparent to users. The original equation set is unchanged, so that substituted variables do not disappear from the model; they can still be used in discrete-event statements. In this way, expressiveness of the models is preserved. Furthermore, the use of substitute equations makes it clear which variables can be chosen freely, and which ones are calculated. Substitute equations can also be used to reveal hidden constraints in index 1 DAEs. Finally, substitute equations enable the use of general purpose numerical solvers for equations where an unknown variable is discontinuous and can be expressed in a closed form.

References

1. Bachmann, R., Brüll, L., Mrziglod, T., and Pallaske, U., On methods for reducing the index of differential algebraic equations. Computers and Chemical Engineering, 14 (1990), 1271 – 1273.

2. Cellier, F. E., Elmqvist, H., Otter, M., and Taylor, J. H., Guidelines for modeling and simulation of hybrid systems. In: IFAC 12th Triennial World Congress, Sydney, 1993, 1219 – 1225.

3. Eich-Soellner, E. and Fuehrer, C., Numerical Methods in Multibody Dynamics. Teubner, Stuttgart, 1998.

4. Fábián, G., A Language and Simulator for Hybrid Systems. Ph.D. thesis, Eindhoven University of Technology, 1999.

5. Gear, C. W., Differential-algebraic equation index transformations. SIAM. J. Sci. Stat. Comp., 9 (1988), 39 – 47.

6. Gear, C. W. and Petzold, L. R., ODE methods for the solution of differential/algebraic systems. SIAM Journal on Numerical Analysis, 21 (1984), 716 – 728.

7. Mattsson, S. E. and Söderlind, G., Index reduction in differential-algebraic equations using dummy derivatives. SIAM J. Sci. Comput., 14 (1993), 677 – 692.

8. Pantelides, C. C., The consistent initialization of differential-algebraic systems. SIAM. J. Sci. Stat. Comput., 9 (1988), 213 – 231.

9. Van Beek, D. A. and Rooda, J. E., Specification of discontinuities in hybrid models. In: Hybrid Dynamical Systems—Proc. of 3rd International Conference on Automation of Mixed Processes, Reims, 1998, 415 – 420.

10. Van Beek, D. A. and Rooda, J. E., Languages and applications in hybrid modelling and simulation: Positioning of Chi. Control Engineering Practice, 8 (2000), 81 – 91.