Implicit Modeling and Simulation of Discontinuities in Physical System Models

Pieter J. Mosterman*
Institute of Robotics and Mechatronics
DLR Oberpfaffenhofen, Germany
Pieter.J.Mosterman@dlr.de
http://www.op.dlr.de/~pjm

Abstract

The design of continuous models of physical systems is often most convenient, less error prone, and intuitive with declarative noncausal constraints that allow an implicit formulation. Discontinuous changes in continuous variables, however, are in general still modeled by explicit reinitialization constructs. This paper shows that it is beneficial to model discontinuous changes implicitly as well by changing sets of algebraic constraints. It also shows that the computation of explicit changes can be automated based on deriving the Kronecker canonical form. An approach that is numerically more stable is presented for index 1 systems based on deriving a pseudo Kronecker canonical form.

Keywords: DAE initialization, DAE modeling, simulation, variable structure systems, hybrid systems

1 Introduction

Object oriented modeling has proven to be a powerful approach to handle the complexity of controlled physical systems [CFM99, MML99]. It relies on object interaction through well-defined ports and allows a noncausal approach to modeling where behavior equations are available in a declarative form. Once the complete set of equations is gathered from each of the constituent behavior specifications along with the connection equations, if the number of equations and variables corresponds, an equation sorting and solving algorithm can derive which equations compute which variables [And94]. Thus, computational causality is assigned and the original implicit formulation is converted into an explicit one. This approach works well for modeling continuous behavior and it is argued that the implicit constraint formulation is indispensable to handle complex systems [CEO96].

In many cases, physical systems are simulated more efficiently if discontinuities in behavior are allowed. Though it is often conceptually easier to design a continuous model, possibly with highly nonlinear behaviors, the phenomena that ensure continuity cause steep gradients in system behavior that are difficult to deal with computationally [Bre96].

This is illustrated by the end stop of a hydraulic cylinder, shown in Fig. 1. In a continuous model, the behavior can be modeled as nonlinear with a steep gradient in the exerted force when the cylinder reaches the end stop ($\Delta x > 0$). In a linear approximation, this characteristic can be modeled as a switch between two modes with linear models. Note that the change in piston velocity (a state variable) is quick but continuous, therefore, this is called a C^0 *hybrid system*, i.e., the 0^{th} time derivative of state variables is continuous. A further abstraction removes the steep gradient and disallows the cylinder to move beyond $\Delta x = 0$. In this case, there is a change in causality of the model, and this causes the piston velocity to change discontinuously.

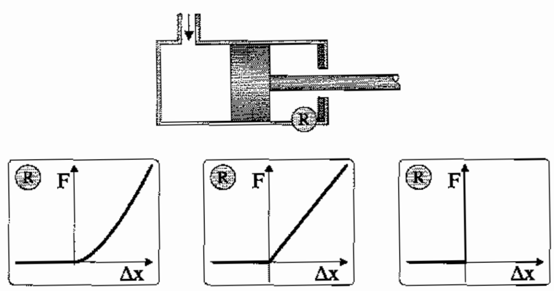

Figure 1: Level of detail of end stop behavior, nonlinear, C^0 hybrid, and hybrid.

In the Modelica^{TM} [E+99] modeling language, different contraints exist for continuous and discrete variables. For continuous variables, the single assignment rule holds, i.e., at each point in time there have to be as many equations as there are unknowns, besides, to have a well formed system, the Jacobian has to be regular. Discrete variables, d_i, on the other hand, have implicit equations $\dot{d}_i = 0$ during continuous integration. At time points when discrete events occur, this equation is not present, and, therefore, any number of equations can be added with an upper bound of the number of unknowns. In case less equations are added, this may lead to underconstrained problems that require heuristics such as a minimum norm projection to achieve an executable specification.

*Pieter J. Mosterman is supported by a grant from the DFG Schwerpunktprogramm KONDISK.

From a modeling perspective, it is desirable to handle discontinuous changes in continuous variables similarly. It will be shown how switching of equations leads to an intuitive approach based on implicit modeling of discontinuities and that it aids the modeling effort and results in a structured approach to the design of hybrid models of physical systems. Because no additional equations are allowed, existing equations have to be removed to prevent an overdetermined system. To avoid the ensuing complexity, at present most modeling and simulation tools require such discontinuities to be explicitly formulated, e.g., in Modelica state variables can only be re-initialized by means of an explicit "reinit" operator.

This paper shows that in many cases the need for such an explicit formulation is inconvenient and hampers the modeling effort. Furthermore, it introduces additional equations so the number of unknown variables and the number of equations does not correspond anymore. This is solved by changing the status of some of the state variables to unknown. For this method to apply, it therefore has to be known exactly which state variable is re-initialized, and, consequently, has to be considered an unknown.

Analogous to the continuous modeling effort, the implicit formulation of discontinuities requires additional model manipulation algorithms to allow simulation. It is shown how such implicit formulations can be systematically treated and an approach is presented for numerical computation of discontinuities in index 1 systems.

2 STATE REINITIALIZATION

Hybrid models of physical systems may contain jumps in state variable values. For example, consider the two colliding bodies in Fig. 2 where m_1 has an initial velocity, $v_1 = v$ while m_2 is at rest. Upon collision, momentum transfers from m_1 to m_2 depending on their masses, and a discontinuous change in the state variable values, the velocities of the masses, takes place.

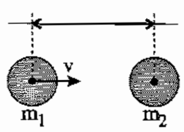

Figure 2: Collision between two bodies.

In many cases, such a system is modeled by a system of differential equations where v_1 and v_2 are re-initialized when collision is detected

$$
\begin{array}{l}
m_1\dot{v}_1 = 0 \\
m_2\dot{v}_2 = 0 \\
\dot{x}_1 = v_1 \\
\dot{x}_2 = v_2 \\
\text{if edge}(x_1 \geq x_2) \text{ then} \\
\quad v_1 = \frac{m_1}{m_1+m_2}(v_1^- + v_2^-) \\
\quad v_2 = \frac{m_2}{m_1+m_2}(v_1^- + v_2^-) \\
\text{endif}
\end{array}
\tag{1}
$$

where the $m_i\dot{v}_i = 0$ equation indicates there is no external force present. The *edge* operator is used to indicate that the if clause is only to be executed once, i.e., when the condition first becomes true. In Modelica, the re-initialization is conveniently facilitated by a special operator, *reinit*. Along with the *pre* operator that returns the *a priori* value of a variable around a discontinuity one can write, e.g., $reinit(v_1, \frac{m_1}{m_1+m_2}(pre(v_1) + pre(v_2))$.

An important characteristic of the reinit operator is that it introduces additional equations, e.g., $v_1 = \frac{m_1}{m_1+m_2}(v_1^- + v_2^-)$. In case of the colliding bodies, the re-initialization equations are added to the existing equations of continuous behavior. Therefore, to arrive at a uniquely determined system of equations, additional unknowns must be introduced as well. To this end, the variables to be initialized (v_1 and v_2) are selected to be unknown at the time of collision although they are state variables otherwise, and, therefore, known.

The need to explicitly model the discontinuous change does not stroke with the principle of implicit modeling for continuous behavior. Because of the explicit nature, the user has to supply the re-initialization equations and select on which state variables this operates. This becomes unwieldy and error-prone, and is inconvenient in case of more complex situations.

Note the difference between the typical treatment of discrete variables, i.e., variables that are constant during continuous integration and only change at event times. For these variables, d_i, no continuous behavior is specified, and, therefore, implicitly it is assumed that $\dot{d}_i = 0$. Because these equations are not explicitly modeled, any time when an equation operating on discrete variables is activated, a discrete variable is set to be unknown. As a result, any number of equations, n_{eq}, can be active as long as it is not more than the number of discrete variables, n_d. In case there are less equations than variables, a minimum norm fit can be used to find values for all variables [KMRW97]. This in contrast to the continuous part, where the number of variables and equations has to always match (except when the reinit operator is used).

3 IMPLICIT MODELING OF DISCONTINUITIES

The need to explicitly model discontinuous changes can be circumvented by using additional automated model manipulation methods. This allows discontinuous changes in state variables to be conveniently and systematically modeled by switching algebraic equations that may include algebraic constraints between state variables. In this case, the system is of a high index, and only a subspace of the state space is accessible [Lew92, VLK81]. This subspace and the projection space that contains the discontinuous changes can be systematically computed [GM86].

To illustrate, consider the two colliding bodies in Fig. 2. The differential equation part of this system can be modeled by

$$\begin{bmatrix} m_1 & 0 & 0 & 0 \\ 0 & m_2 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{v}_1 \\ \dot{v}_2 \\ \dot{F}_1 \\ \dot{F}_2 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ F_1 \\ F_2 \end{bmatrix} \tag{2}$$

where F_i is the force acting on body m_i. This system of differential equations is complemented by the algebraic equations

$$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ F_1 \\ F_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \tag{3}$$

i.e., there is no external force acting.

Upon collision, Newton's collision rule

$$v_2 - v_1 = -\varepsilon(v_2^- - v_1^-) \tag{4}$$

becomes active. The values of v_1^- and v_2^- are known as their final value when first $x_1 \geq x_2$ and continuous behavior was halted. Furthermore, upon collision the forces F_1 and F_2 are equal but opposite, which changes the system of equations by replacing Eq. (3) with

$$\begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \\ v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} 0 \\ -\varepsilon(v_2^- - v_1^-) \end{bmatrix}. \tag{5}$$

Using Eq. (2) to solve the equation in the top row of Eq. (5) yields $m_1\dot{v}_1 + m_2\dot{v}_2 = 0$ and this can be integrated over an infinitesimal interval $[t^-, t]$ to

$$m_1(v_1 - v_1^-) + m_2(v_2 - v_2^-) = 0, \tag{6}$$

which embodies the physical conservation of momentum constraint. Combined with Eq. (4) it can be uniquely solved for v_1 and v_2.

The advantage of this approach is that no explicit change in velocity is prescribed. Only Newton's collision rule is activated at the time of collision. The corresponding change in velocity can be automatically computed even when more bodies engage in a collision simultaneously.

Allowing this change of active equations at events corresponds to the mechanism for handling discrete variables, *viz.*, that discrete state variable values are not considered to be known at event times. Continuous states that are not part of algebraic constraints remain unchanged, i.e., the equation $x_i = x_i^-$ holds during discrete changes analogous to the $\dot{d}_i = 0$ equation for discrete variables during continuous behavior.

In general, to compute the new values of state variables when algebraic constraints are activated, a minimum norm approach can be applied. However, it will be shown that only in particular cases this obeys physical principles. Instead, an approach is derived based on the use of the Kronecker Canonical Form (KCF) [DK86].

Consider the system of differential and algebraic equations

$$E\dot{x} = Ax. \tag{7}$$

Using the λ operator to represent differentiation, this can be written as a matrix pencil

$$(\lambda E - A)x = 0 \tag{8}$$

To transform this system of equations into the KCF, the following decomposition is applied

$$Q(\lambda E - A)ZZ^{-1}x = 0 \tag{9}$$

where a change of basis, $y = Z^{-1}x$, is applied. This then yields the system of equations

$$(\lambda\hat{E} - \hat{A})y = 0 \tag{10}$$

where $K = \lambda\hat{E} - \hat{A}$ is in the KCF.

The change of basis represents a state mapping between the state variable values immediately before the event that caused the change in equations, x^-, and the new values, x. This mapping is derived by integrating the system of equations over an infinitesimal time interval, $[t^-, t]$, during which only impulsive terms contribute. In the KCF these impulsive terms correspond to the variables multiplied by the λ operator.

To illustrate the approach, consider the collision in Fig. 2 in case it is perfectly nonelastic

$$\begin{bmatrix} m_1 & m_2 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{v}_1 \\ \dot{v}_2 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ m_1 m_2 & -m_1 m_2 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \tag{11}$$

Here the E and A matrices are singular. The described approach leads to the transformation

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \lambda m_1 & \lambda m_2 \\ -m_1 m_2 & m_1 m_2 \end{bmatrix} \begin{bmatrix} \frac{1}{m_1+m_2} & -\frac{1}{m_1(m_1+m_2)} \\ \frac{1}{m_1+m_2} & \frac{1}{m_2(m_1+m_2)} \end{bmatrix} \tag{12}$$

to arrive at the KCF

$$\begin{bmatrix} \lambda & 0 \\ 0 & 1 \end{bmatrix} \tag{13}$$

Here Z^{-1} is computed as

$$\begin{bmatrix} m_1 & m_2 \\ -m_1 m_2 & m_1 m_2 \end{bmatrix} \tag{14}$$

which results in

$$\begin{cases} m_1 v_1 + m_2 v_2 = m_1 v_1^- + m_2 v_2^- \\ -m_1 m_2 v_1 + m_1 m_2 v_2 = 0 \end{cases} \tag{15}$$

and this computes the change of momentum

$$\begin{cases} v_1 = \frac{1}{m_1+m_2}(m_1 v_1^- + m_2 v_2^-) \\ v_2 = \frac{1}{m_1+m_2}(m_1 v_1^- + m_2 v_2^-) \end{cases} \tag{16}$$

and conservation of momentum is realized. In other work a minimum norm fit is proposed [KMRW97] in which case the new values are (projection along $v_1 + v_2 = v_1^- + v_2^-$ onto $v_1 = v_2$)

$$\begin{cases} v_1 = \frac{1}{2}(v_1^- + v_2^-) \\ v_2 = \frac{1}{2}(v_1^- + v_2^-) \end{cases} \tag{17}$$

and only if $m_1 = m_2$ momentum is conserved.

4 A Hydraulics Example

The benefits of implicit modeling of discontinuous state changes becomes even clearer in case of a more complex example where the discontinuous changes are not as straightforward. Consider the hydraulic actuator shown in Fig. 3. In normal operation, the input pressure, p_{in}, is used to control the position of the piston with mass, m_p, that may be attached to, e.g., the elevator control surface of an airplane. When the input valve, v_{in}, is open there is a flow of oil through the valve, f_{in}, into the cylinder. In case the valve is closed this flow is 0. When open, the inflow is determined by the pressure drop between the input pressure and the cylinder pressure, p_{cyl}, and the valve resistance, R_{in}. This can be modeled by

$$\text{if } v_{in} \text{ then } f_{in}R_{in} = p_{in} - p_{cyl} \text{ else } f_{in} = 0 \text{ endif} \quad (18)$$

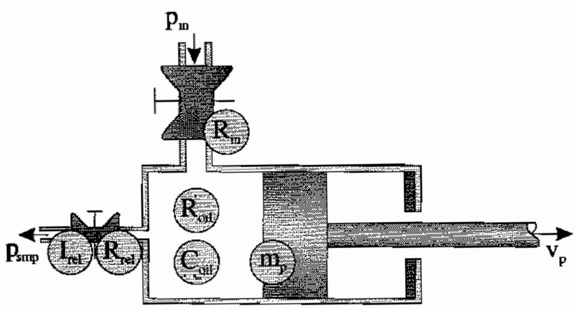

Figure 3: An actuation cylinder and its physical phenomena.

If the cylinder pressure exceeds a threshold value, the relief valve, v_{rel}, may open to prevent damage. Similar relations as for the input valve hold, where f_{rel} is the flow of oil through the relief valve out of the cylinder, R_{rel} the valve resistance to flow and p_{sump} the oil pressure of the sump

$$\text{if } v_{rel} \text{ then } I_{rel}\dot{f}_{rel} = p_{rel} \text{ else } f_{rel} = 0 \text{ endif} \quad (19)$$

with

$$p_{rel} = p_{sump} - f_{rel}R_{rel} + p_{cyl} \quad (20)$$

Here, because of the dimensions of the relief valve piping, a fluid inertia, I_{rel}, is attributed to it.

Furthermore, if the small elasticity and dissipation parameters of the oil, C_{oil} and R_{oil}, are not modeled, the net flow of oil into the cylinder, $f_{in} - f_{rel}$, has to correspond to the increased volume because of piston movement $A_p f_p$, or

$$A_p f_p = f_{in} - f_{rel} \quad (21)$$

where A_p is the area of the piston surface. The force acting on the piston, $m_p \dot{f}_p$, depends on the cylinder pressure, p_{cyl},

$$m_p \dot{f}_p = A_p p_{cyl} \quad (22)$$

To derive the KCF, first the vector of system variables, x, is determined as

$$x = [f_p \; f_{rel} \; f_{in} \; p_{cyl} \; p_{rel}]^T \quad (23)$$

Now, the differential equation part that holds in each configuration of closed and open valves is

$$\begin{bmatrix} m_p & 0 & 0 & 0 & 0 \\ 0 & I_{rel} & 0 & 0 & 0 \end{bmatrix} \dot{x} = \begin{bmatrix} 0 & 0 & 0 & A_p & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} x \quad (24)$$

The algebraic equations may change in each configuration. Initially, if the input valve is open and the relief valve is closed they are

$$0 = \begin{bmatrix} A_p & 1 & -1 & 0 & 0 \\ 0 & 0 & -R_{in} & -1 & 0 \\ 0 & -R_{rel} & 0 & 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ p_{in} \\ 0 \end{bmatrix} \quad (25)$$

If during operation the input valve closes, there may be a quick build-up of pressure in the cylinder that causes the relief valve to open. In this configuration the input valve is closed and the relief valve is open, and the algebraic equations are

$$0 = \begin{bmatrix} A_p & 1 & -1 & 0 & 0 \\ 0 & 0 & -R_{in} & 0 & 0 \\ 0 & -R_{rel} & 0 & 1 & -1 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ p_{sump} \end{bmatrix} \quad (26)$$

Note that in both cases the number of equations and unknowns is five.

To compute the state mapping between these two modes, let's first reduce the combined system of differential and algebraic equations to its second order equivalent

$$\begin{bmatrix} m_p & -A_p I_{rel} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{f}_p \\ \dot{f}_{rel} \end{bmatrix} = \begin{bmatrix} 0 & -A_p R_{rel} \\ A_p & 1 \end{bmatrix} \begin{bmatrix} f_p \\ f_{rel} \end{bmatrix} \quad (27)$$

This can be written in the KCF by the following transformations

$$Q = \begin{bmatrix} 1 & \frac{R_{rel} A_p m_p}{A_p^2 I_{rel} + m_p} \\ 0 & 1 \end{bmatrix} \quad (28)$$

$$Z = \begin{bmatrix} \frac{m_p}{A_p^2 I_{rel} + m_p} & \frac{A_p I_{rel}}{m_p} \\ -\frac{A_p m_p}{A_p^2 I_{rel} + m_p} & 1 \end{bmatrix} \quad (29)$$

which yields

$$K = \lambda\hat{E} - \hat{A} = \begin{bmatrix} \lambda m_p + \frac{A_p^2 R_{rel} m_p}{A_p^2 I_{rel} + m_p} & 0 \\ 0 & \frac{A_p^2 I_{rel} + m_p}{m_p} \end{bmatrix} \quad (30)$$

The change of basis $y = Z^{-1}x$ requires

$$Z^{-1} = \begin{bmatrix} 1 & -\frac{A_p I_{rel}}{m_p} \\ \frac{A_p m_p}{A_p^2 I_{rel} + m_p} & \frac{m_p}{A_p^2 I_{rel} + m_p} \end{bmatrix} \quad (31)$$

and because $y_1 = y_1^-$ and $y_2 = 0$, [1]

$$\begin{cases} f_p - \frac{A_p I_{rel}}{m_p} f_{rel} = f_p^- - \frac{A_p I_{rel}}{m_p} f_{rel}^- \\ A_p m_p f_p + m_p f_{rel} = 0 \end{cases} \tag{32}$$

which gives the correct, momentum conserving, state mapping

$$\begin{cases} f_p = -\frac{1}{A_p^2 I_{rel} + m_p}(A_p I_{rel} f_{rel}^- - m_p f_p^-) \\ f_{rel} = \frac{A_p}{A_p^2 I_{rel} + m_p}(A_p I_{rel} f_{rel}^- - m_p f_p^-) \end{cases} \tag{33}$$

Note that R_{rel} does not play a role in this mapping because it only affects continuous behavior.

In case of explicit re-initialization the modeler has to identify f_{rel} and f_p as state variables and calculate the re-initialization formula. The implicit approach automates these computations and allows a much more intuitive, flexible, and elegant but also systematic model specification.

5 Implementation for Index 1 Systems

Because of the numerical difficulty to compute the KCF, a pseudo canonical form is used. The implementation of all operations that are described relies on LAPACK [ABB+95] and BLAS calls.

First, the original system

$$E\dot{x} + Ax + Bu = 0 \tag{34}$$

is tranformed by the decomposition

$$QEZZ^{-1}\dot{x} + QAZZ^{-1}x + QBu = 0 \tag{35}$$

to arrive at the form

$$\begin{bmatrix} \tilde{E}_{11} & \tilde{E}_{12} \\ 0 & \tilde{E}_{22} \end{bmatrix} \begin{bmatrix} \dot{\tilde{x}}_1 \\ \dot{\tilde{x}}_2 \end{bmatrix} + \begin{bmatrix} \tilde{A}_{11} & \tilde{A}_{12} \\ 0 & \tilde{A}_{22} \end{bmatrix} \begin{bmatrix} \tilde{x}_1 \\ \tilde{x}_2 \end{bmatrix} + \begin{bmatrix} \tilde{B}_1 \\ \tilde{B}_2 \end{bmatrix} \begin{bmatrix} u \end{bmatrix} = 0 \tag{36}$$

where the top and bottom rows correspond to the finite and infinite eigenvalues, respectively. In case of an index 1 system, the matrix $\tilde{E}_{22} = 0$ [VLK81], and, therefore, $\tilde{x}_2$ can be computed explicitly to be $\tilde{x}_2 = -\tilde{A}_{22}^{-1}\tilde{B}_2 u$. However, substitution in the finite part to compute $\tilde{x}_1$ requires the time derivative of u because of the $\tilde{E}_{12}$ cross coupling. Therefore, a further coordinate transformation $\bar{x} = P^{-1}\tilde{x}$ is performed with

$$P = \begin{bmatrix} I & P_{12} \\ 0 & I \end{bmatrix} \tag{37}$$

to arrive at the form

$$\begin{bmatrix} \bar{E}_{11} & 0 \\ 0 & \bar{E}_{22} \end{bmatrix} \begin{bmatrix} \dot{\bar{x}}_1 \\ \dot{\bar{x}}_2 \end{bmatrix} + \begin{bmatrix} \bar{A}_{11} & \bar{A}_{12} \\ 0 & \bar{A}_{22} \end{bmatrix} \begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} \begin{bmatrix} u \end{bmatrix} = 0 \tag{38}$$

[1] Note that the $\frac{A_p^2 R_{rel} m_p}{A_p^2 I_{rel} + m_p}$ term in the K matrix does not contribute when integrated over an infinitesimal time interval.

For this transformation, P_{12} is computed from the requirement that $\tilde{E}_{11}P_{12} + \tilde{E}_{12} = 0$. Straightforward computation shows that $\bar{E}_{11} = \tilde{E}_{11}$, $\bar{E}_{22} = \tilde{E}_{22}$, $\bar{A}_{11} = \tilde{A}_{11}$, $\bar{A}_{22} = \tilde{A}_{22}$, and $\bar{A}_{12} = \tilde{A}_{11}P_{12} + \tilde{A}_{12}$.

To compute the initial values, the Laplace transform ($\dot{x} = sX - x^-$) is applied. Because only the initial conditions in the finite part can be chosen freely, this yields

$$(s\bar{E}_{11} + \bar{A}_{11})\bar{X}_1 + \bar{A}_{12}\bar{X}_2 + \bar{B}_1 U = \bar{E}_{11}\bar{x}_1^- \tag{39}$$

The Laplace tranform of the system in original coordinates is

$$(sQE + QA)X + QBU = QEx^- \tag{40}$$

and equating the initial conditions leads to

$$\bar{x}_1 = \bar{E}_{11}^{-1} QEx^- \tag{41}$$

where $\bar{E}_{11}$ ($= \tilde{E}_{11}$) is of full rank. Combined with $\bar{x}_2 = -A_{22}^{-1}B_2 u$, the initial conditions of the transformed system are determined, and, therefore, the consistent initial values of the original system can be computed by

$$x = ZP\bar{x} = ZP \begin{bmatrix} \bar{E}_{11}^{-1} QEx^- \\ -A_{22}^{-1}B_2 u \end{bmatrix} \tag{42}$$

These computations are applied to a model of the actuator cylinder that is automatically generated from a hybrid bond graph model by HyBrSim [MB99] with variables $x = [f_p \; f_{rel} \; p_{R_{rel}} \; p_{R_{in}}]^T$ where $p_{R_{rel}}$ and $p_{R_{in}}$ are the pressure drop across the resistances R_{rel} and R_{in}, respectively. For parameters $m_p = 5, A_p = 1, I_{rel} = 10$, $R_{in} = 5$, and $R_{rel} = 5$, in the mode when the supply valve is closed and the relief valve open, the following output is generated (u does not affect the computations and is not listed):

$$x^- = \begin{bmatrix} 3 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{43}$$

$$E = \begin{bmatrix} 0 & 0 & 0 & 0 \\ -1 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{44}$$

$$A = \begin{bmatrix} -5 & 0 & -0.999911 & 0 \\ 0 & 0 & 0.2 & 0 \\ -1 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \tag{45}$$

$$\tilde{E} = \begin{bmatrix} 0.577303 & -0.0498672 & 0 & 2.15968 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{46}$$

$$\tilde{A} = \begin{bmatrix} 0.192451 & -0.0166239 & 0 & -0.0518278 \\ 0 & -4.68643 & 0 & -2.19267 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1.10876 \end{bmatrix} \tag{47}$$

$$Q = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 0.957826 & 0 & 0.287348 & 0 \\ 0 & 0 & 0 & 1 \\ -0.287348 & 0 & 0.957826 & 0 \end{bmatrix} \tag{48}$$

$$Z = \begin{bmatrix} 0.192434 & 0.881135 & 0 & 0.431937 \\ -0.192434 & 0.465501 & 0 & -0.863874 \\ -0.962257 & 0.0831195 & 0 & 0.259139 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{49}$$

$$P = \begin{bmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{50}$$

$$x = \begin{bmatrix} 1 \\ -1 \\ -5.00044 \\ 0 \end{bmatrix} \tag{51}$$

Note that $\bar{E}$ is not explicitly computed. Also note that the computed values in Eq. (51) given the initial values in Eq. (43) comply with the symbolic computations in Eq. (33).

6 Conclusions

Continuous modeling of complex physical systems is supported by the use of implicit modeling techniques. Automated symbolic processing derives the explicit form of the implicit system of equations. It is shown that the same principle of implicit modeling applies to discontinuous state changes when model configuration changes occur. By conveniently changing algebraic constraints, a new system of differential and algebraic equations is formed that may require projection of state variable values onto a subspace of behavior. This projection is computed by transforming the matrix pencil into the Kronecker canonical form (KCF).

A disadvantage of the approach is the difficulty of obtaining a KCF in general. A pseudo canonical form that allows numerical stability is shown to be applicable for index 1 systems.

7 Acknowledgement

Thanks to Andras Varga (DLR Oberpfaffenhofen) for his help in developing the calculations and implementing the numerics.

References

[ABB+95] E. Anderson, Z. Bai, J. Bishop, J. Demmel, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov, and D. Sorensen. *LAPACK User's Guide, Second Edition.* SIAM, Philadelphia, 1995.

[And94] M. Andersson. *Object-Oriented Modeling and Simulation of Hybrid Systems.* PhD dissertation, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden, 1994.

[Bre96] Peter C. Breedveld. The context-dependent trade-off between conceptual and computational complexity illustrated by the modeling and simulation of colliding objects. In *CESA '96 IMACS Multiconference*, Lille, France, July 1996.

[CEO96] F.E. Cellier, H. Elmqvist, and M. Otter. Modelling from physical principles. In W.S. Levine, editor, *The Control Handbook*, pages 99–107. CRC Press, Boca Raton, FL, 1996.

[CFM99] Emanuele Carpanzano, Luca Ferrarini, and Claudio Maffezzoni. An object-oriented model for hybrid control systems. In *1999 IEEE International Symposium on Computer Aided Control System Design*, pages 132 – 137, Kohula Coast Island of Hawai'i, Hawai'i, August 1999.

[DK86] James Demmel and Bo Kågström. Stably computing the Kronecker structure and reducing subspaces of singular pencils $A - \lambda B$ for uncertain data. In J. Cullum and R. A. Willoughby, editors, *Large Scale Eigenvalue Problems*. Elsevier Science Publishers B.V. (North-Holland), 1986.

[E+99] Hilding Elmqvist *et al.* Modelica™–a unified object-oriented langauge for physical systems modeling: Language specification, December 1999. version 1.3, http://www.modelica.org/.

[GM86] Eberhard Griepentrog and Roswitha März. *Differential-Algebraic Equations and Their Numerical Treatment.* BSB Teubner, Leipzig, 1986.

[KMRW97] Peter Kunkel, Volker Mehrmann, Werner Rath, and Jörg Weickert. A new software package for linear differential-algebraic equations. *SIAM Journal of Scientific Computing*, 18(1):115–138, January 1997.

[Lew92] F. L. Lewis. A tutorial on the geometric analysis of linear time-invariant implicit systems. *Automatica*, 28:119–137, 1992.

[MB99] Pieter J. Mosterman and Gautam Biswas. A Java Implementation of an Environment for Hybrid Modeling and Simulation of Physical Systems. In *ICBGM99*, pages 157–162, San Francisco, January 1999.

[MML99] Dieter Moormann, Pieter J. Mosterman, and Gert-Jan Looye. Object-oriented computational model building of aircraft flight dynamics and systems. *Aerospace Science and Technology*, 3:115–126, 1999.

[VLK81] George C. Verghese, Bernard C. Lévy, and Thomas Kailath. A generalized state-space for singular systems. *IEEE Transactions on Automatic Control*, 26(4):811–831, August 1981.