Proceedings of the 2003 Winter Simulation Conference
S. Chick, P. J. Sánchez, D. Ferrin, and D. J. Morrice, eds.

MODE TRANSITION BEHAVIOR IN HYBRID DYNAMIC SYSTEMS

Pieter J. Mosterman

The MathWorks, Inc.
3 Apple Hill Dr.
Natick, MA 01760, U.S.A.

ABSTRACT

Physical system modeling benefits from the use of implicit equations because it is often an intuitive way to describe physical constraints and behaviors. To achieve efficient models, model abstraction may lead to idealized component behavior that switches between modes of operation (e.g., an electrical diode may be on or off) based on inequalities (e.g., voltage > 0). In an explicit representation, the combination of these local mode switches leads to a combinatorial explosion of the number of global modes. It is shown how an implicit formulation can be used to formulate these mode switches, thereby circumventing the combinatorial problem. This leads to the use of differential and algebraic equations (DAEs) for each of the modes. In case these DAEs are of high index, jumps in generalized state variables may occur. In combination with the inequalities that define mode switching, this leads to rich and complex mode transition behavior. An overview of this mode switching behavior and an ontology is presented.

1 INTRODUCTION

Modern engineered systems have reached a complexity that requires systematic design methodologies and model based approaches to ensure correct and competitive realization. In addition, the use of digital controllers has become critical. Embedded software, however, has proven to be difficult to manage since small errors may lead to catastrophic failures. Furthermore, the interdependencies in software that implements the control algorithms are difficult to oversee, which only exacerbates with the increasing size of embedded software. Similarly, the interdependencies between controllers scattered about the control system are difficult to distill. Their effects as well as the subtle interaction between controllers and the physical environment are difficult to analyze.

Modeling can be the mortar to combine the controller software and hardware of the controlled system, the *plant*, but different modeling paradigms are used for the different domains. To model the plant, differential and algebraic equations (DAEs) are the method of choice. The controller, on the other hand, is typically modeled by a discrete time or discrete event formalism. In early design stages, continuous models may be preferred because of the analysis and synthesis benefits, but when moving to a software implementation, at one point a discretized version has to be derived. The combined controller/plant analysis then requires mixed continuous/discrete formalisms, or so-called *hybrid dynamic systems* (Benedetto and Sangiovanni-Vincentelli 2001, Lynch and Krogh 2000, Vaandrager and van Schuppen 1999).

In addition, hi-fidelity plant models often include highly nonlinear behaviors that complicate analyses. In a more efficient model, these nonlinearities may be linearized around one or more operating points. Switching between these linearized models then requires a discrete mode switching control structure combined with the continuous models in each of the modes, leading to a hybrid dynamic system as well.

Another ground for using discrete switching effects in plant models is to model perceived physical discontinuities such as valves, overflows, and collisions. In many cases, it is more convenient to model such phenomena as discontinuities although it may require quite some additional conceptual investment compared to a more detailed continuous model (Breedveld 1996).

There is a distinct difference between hybrid dynamic systems that arise from combining controller models with plant models and those that emerge because of including discontinuities in the plant models of physical systems. Controller models are by design of an explicit nature. Plant models, on the other hand, often contain implicit constraints that should be satisfied, without explicitly stating how these are used to generate behavior. For example, Newton's collision law says that the difference of velocities of two colliding bodies after a collision, Δv, equals their difference before, Δv^-, given some coefficient of restitution, ϵ,

$$\Delta v = -\epsilon \Delta v^- \tag{1}$$

This is a general constraint that does not prescribe how the velocities are to be computed.

In contrast to the hybrid dynamic systems that are applied in the control system realm, those that are of an implicit form as found in plant models have much richer behavior, especially with respect to the interaction between the continuous and discrete model parts.

This paper gives a cursory overview of classes of behavior as found in hybrid dynamic systems in general. To this end, Section 2 first presents the modeling of physical systems and how modeled discontinuities may result in complications. Section 3 presents the classes of behavior in a geometric representation. Section 4 then presents these in more detail for linear systems in an algebraic representation. Finally, Section 5 presents the conclusions of this work.

2 PLANT MODELING

To introduce hybrid dynamic systems that result from modeling of physical systems, *plant modeling*, the cylinder of a hydraulic actuator is modeled. Hydraulic actuators are typically used in civil aircraft to position the control surfaces for pitch, yaw, and roll. A simplified representation of such a cylinder is shown in Fig. 1, annotated with the modeled physical phenomena (in circles). At the top, the supply pressure, p_{in}, provides a flow of oil through the intake line, f_{in}. This flow can be interrupted by a valve with resistance, R_{in}.

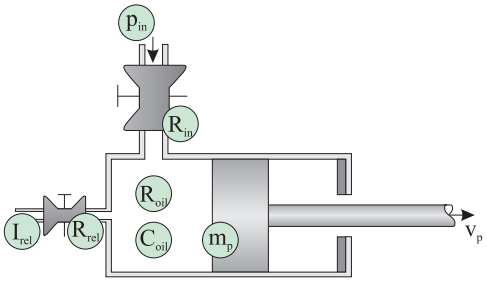


Figure 1: Model Parameters of a Hydraulic Actuator

The oil in the cylinder chamber is modeled by its elasticity, C_{oil}, and its viscosity, R_{oil}, which resembles a spring/damper effect in the mechanical domain. Elasticity can also be attributed to air in the circuit. The chamber pressure works on the cylinder piston with inertia, m_p, and velocity, v_p. The piston is connected to the particular control surface by means of a mechanical linkage (not shown) and so the flow of oil into the cylinder controls the position of the surface.

A relief path is present for a flow of oil, f_{rel}, back to the sump with reference pressure, $p_{smp} = 0$. This prevents possibly damaging high pressures inside the cylinder chamber. The relief valve is modeled by its resistance, R_{rel}, and inertia, I_{rel}. The inertia is included because of the size of the relief line. In case of a small fluid flow path, the inertia of the moving fluid has to be accounted for (compare spraying with a garden hose by making its opening smaller).

These basic physical phenomena represent a set of constraints that comprise the cylinder model:

$$
\begin{aligned}
&\phi_{in}(f_{in}, p_{R_{in}}, s_{in}) = 0 \\
&f_{R_{oil}} R_{oil} = p_{R_{oil}} \\
&C_{oil}\dot{p}_{C_{oil}} = f_{R_{oil}} \\
&m_p \dot{v}_p = p_{cyl} \\
&\phi_{rel}(f_{rel}, p_{R_{rel}}, s_{rel}) = 0 \\
&I_{rel}\dot{f}_{rel} = p_{rel} \\
&f_{oil} = f_{in} - f_{rel} - v_p \\
&p_{rel} = -p_{R_{rel}} + p_{cyl} \\
&p_{R_{in}} = p_{in} - p_{cyl} \\
&p_{cyl} = p_{R_{oil}} + p_{C_{oil}}
\end{aligned}
\tag{2}
$$

These equations are not written in an explicit form, or, computational causality has not yet been assigned. For example, the equation for the viscosity of the oil, $f_{R_{oil}} R_{oil} = p_{R_{oil}}$, does not capture whether the pressure is computed from the flow or the other way around. This has the distinct advantage that causal changes between modes do not have to be accounted for by the model designer. Plant modeling tools (Andersson 1994) apply sophisticated equation sorting methods to determine the computational causality of the equations, given that the exogeneous variables (here p_{in}) and states (here f_{rel} and v_p) are known.

Given the problem that the model should address, the small oil parameters (stiffness, C_{oil}, and viscosity, R_{oil}) may be abstracted away. Their presence may lead to stiff gradients that operate on a time scale much smaller than the overall behavior to be studied. For example, if the aircraft behavior of interest is the effect of an actuator switch in response to a failure, too detailed an actuator model leads to prohibitively long simulation times (Mosterman, Remelhe, Engell, and Otter 2002).

Note that the two valves are modeled by nonlinear relations, ϕ_{in} and ϕ_{rel}, with additional arguments, s_{in} and s_{rel}, to capture the dimension that determines whether the valves are open or closed. Details of these nonlinear characteristics can be found elsewhere (Mosterman and Biswas 2000b).

Such nonlinear behaviors often cause difficulty in simulation because they may result in stiffness as well. For example, the relief valve changes from enforcing $f_{rel} \approx 0$ to $p_{rel} \approx p_{th}$ over a very short range of f_{rel} and p_{rel}. This causes a very small step size of numerical solvers.

More efficient models replace the nonlinear behavior by a piecewise linear approximation. For example, the

pressure relief valve characteristic, ϕ_{rel}, can be modeled by the following piecewise linear approximation

$$\begin{aligned} f_{rel} R_{rel} &= p_{R_{rel}} \\ R_{rel} &= \textbf{if } s_{rel} \textbf{ then } R_{rel,l} \textbf{ else } R_{rel,h} \end{aligned} \tag{3}$$

where $R_{rel,h}$ is a high resistance when the valve is closed, whereas $R_{rel,l}$ is a low resistance when the valve is open (e.g., $R_{rel,l} = 0$). The Modelica (Elmqvist et al. 1999) notation is used to assign a unique value to R_{rel}, either $R_{rel,l}$ or $R_{rel,h}$ depending on whether s_{rel} is true or false, respectively. Note that this still results in C^0 behavior of the overall system, i.e., the behavior trajectories are continuous but they may not be smooth anymore (the first and higher order derivatives with respect to time may not exist at the switching points). The control signal s_{rel} becomes true as soon as $|p_{rel}| \geq p_{th}$ and remains true from then on, i.e., the relief valve remains open once the threshold pressure has been exceeded until it is reset.

This may still lead to an inefficient simulation model, in particular when $R_{rel,h}$ is a relatively large value it may cause time constants that are several orders of magnitude faster than what is present in the model otherwise.

In a further approximation the valve can be modeled as an ideal switch by

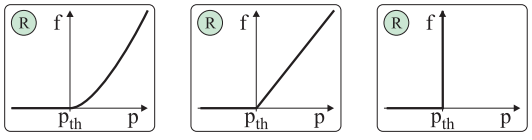
$$\begin{aligned} f_{rel} R_{rel} &= p_{R_{rel}} \\ 0 &= \textbf{if } s_{rel} \textbf{ then } p_{rel} + p_{R_{rel}} - p_{cyl} \textbf{ else } f_{rel} \end{aligned} \tag{4}$$

which replaces

$$\begin{aligned} \phi_{rel}(f_{rel}, p_{R_{rel}}, s_{rel}) &= 0 \\ p_{rel} &= -p_{R_{rel}} + p_{cyl} \end{aligned} \tag{5}$$

in Eq. (2). This, however, may lead to changes in the computational causality during run-time when f_{rel} is forced to be 0, and, therefore at times needs to be treated as an exogeneous variable. So, this approximation eliminates the stiff gradients otherwise found in the model, but it adds complexity as it now needs to be dealt with the changes in computational causality and possibly even changes in the complexity of the system of differential and algebraic equations (DAEs).

In Fig. 2 the three different types of valve models are illustrated.



(a) Nonlinear Model (b) C^0 Hybrid Model (c) Hybrid Model

Figure 2: Levels of Abstraction of a Cylinder End-stop

Note how the change in equations in Eq. (4) is specified locally and in an implicit form, so it can be automatically handled by an implicit solver without the need for re-ordering (sorting) the equations (Mosterman 2002).

To illustrate changes in the complexity of the DAEs that may occur, consider the model so derived:

$$\begin{aligned} 0 &= \textbf{if } s_{in} \textbf{ then } p_{in} - p_{R_{in}} - p_{cyl} \textbf{ else } f_{in} \\ f_{in} R_{in} &= p_{R_{in}} \\ m_p \dot{v}_p &= p_{cyl} \\ f_{rel} R_{rel} &= p_{R_{rel}} \\ I_{rel} \dot{f}_{rel} &= p_{rel} \\ v_p &= f_{in} - f_{rel} \\ 0 &= \textbf{if } s_{rel} \textbf{ then } p_{rel} + p_{R_{rel}} - p_{cyl} \textbf{ else } f_{rel} \end{aligned} \tag{6}$$

It contains two *generalized* state variables (Verghese, Lévy, and Kailath 1981), f_{rel} and v_p. These are generalized state variables as they may not be true states because algebraically related to one another.

To clarify, consider the mode where the relief valve is open and the intake valve is not. So, $f_{in} = 0$, and, consequently, $v_p = f_{rel}$. Therefore, these are not independent states, and instead of a second order system, it is of first order. However, since both v_p as well as f_{rel} are in the system of equations as time-derivatives, a modeling & simulation tool has to manipulate the model to arrive at a form where only one state is present. In effect, the generalized state space is of dimension two, but in this mode the system 'lives' in a one-dimensional subspace. This notion of generalized state space and reduced subspaces in which the system evolves is an intrinsic part of plant modeling and constitutes a large part of the continuous/discrete interaction discussed in the rest of this paper.

3 A GEOMETRIC VIEW

An overview of different classes of behavior in hybrid dynamic systems is given in geometric terms.

3.1 The Elementary Case

A hybrid dynamic system evolves continuously in time in a mode, α_i, according to a field, f_{α_i}, that defines a relation $f_{\alpha_i}(\dot{x}, x, u, t) = 0$ between the state, x, its time derivative, $\dot{x}$, the input u, and the time, t.

A mode transition relation $\gamma_{\alpha_i}^{\alpha_{i+1}}(x, u, t) \geq 0$ defines the change from mode α_i to α_{i+1} when true. The state space in a mode α_i consists of two parts: (i) the domain where f_{α_i} is properly defined and (ii) a *patch*, where $\gamma_{\alpha_i}^{\alpha_{i+1}}$ does not invoke a mode change.

Without loss of generality it is first assumed that the explicitly defined state transition function, $x_{\alpha_{i+1}} = g_{\alpha_i}^{\alpha_{i+1}}(x_{\alpha_i}, u_{\alpha_i}, t)$, is the identity function, i.e., $x_{\alpha_{i+1}} = x_{\alpha_i}$.

This is illustrated in Fig. 3 where a trajectory in a hybrid state space is shown. The patches where the state can evolve continuously in each of the modes are shown as white areas. After initialization in mode α_1, the state evolves till the boundary of the patch of α_1 is reached. At this point, x_{α_1}, a mode transition as defined by $\gamma_{\alpha_1}^{\alpha_2}$ is invoked. In the new mode, a patch defined by $\gamma_{\alpha_2}^{\alpha_3}$ is entered in which the state can continue to evolve, now governed by the field f_{α_2}.

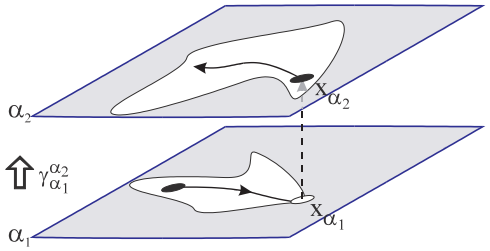


Figure 3: An Elementary Mode Transition

Note that the state behavior is 'left-closed', i.e., each of the abutting intervals of continuous behavior includes its starting point. This satisfies causality requirements (Mosterman 1999a). In general, this left-closedness may (have to) be relaxed, though.

3.2 Discontinuous State Changes

The elementary mode transition behavior in Fig. 3 may become more complex when discontinuous changes in x are present. Instead of only explicitly prescribed jumps in x, in this work discontinuites may also result from changes in the complexity of continuous behavior and the corresponding constraints this puts on the state space accessibility.

In Fig. 3, the domains in α_1 and α_2 where x is accessible are marked by the rectangular surfaces that contain the entire state space. In Fig. 4 a situation arises where in α_2 only a subspace is accessible. Effectively, the two-dimensional generalized state space reduces to a one-dimensional manifold.

When such a collapse of the state space occurs, the original state has to make a discontinuous change to be in the reduced actual state space. In Fig. 4, the state evolves continuously in mode α_1 till it reaches the boudary defined by $\gamma_{\alpha_1}^{\alpha_2}$. The value at this boundary, x_{α_1}, is then transferred to α_2 but now a discontinuous change from $x_{\alpha_2}^-$ to x_{α_2} is required before the state can continue to evolve continuously in mode α_2 as governed by f_{α_2}.

This discontinuous change is determined by a *projection* that takes place in the *instantaneous space* of α_2 as opposed to the *dynamic space* of α_2 that is indicated by the solid line. In Section 4 it will be shown how for a class of linear systems the instantaneous and dynamic spaces can be derived from a matrix pencil, $\lambda E + A$.

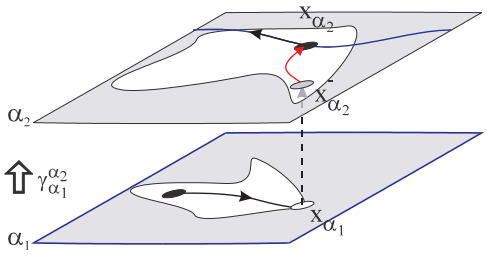
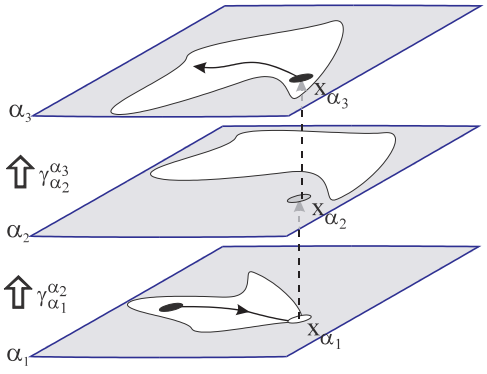


Figure 4: A Mode Transition With Projection

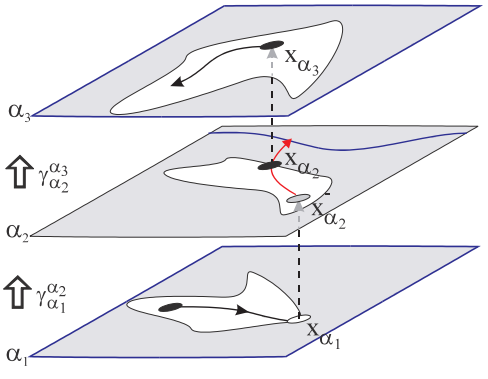
To circumvent the difficulties introduced by the generality of the nonlinearness, a possible implementation to derive the projection may rely on repeated linearization around a point on the projection trajectory, starting at $x_{\alpha_2}^-$, till x_{α_2} is arrived at.

3.3 A Sequence of Mode Transitions

So far, the discrete part of the hybridness has been a single transition between modes. In general, though, a sequence of transitions may occur (Mosterman and Biswas 1998). Two classes of possible behaviors are illustrated in Fig. 5.



(a) Mythical



(b) Pinnacle

Figure 5: Sequences of Mode Transitions

In previous work, the phenomena of *mythical modes* (Nishida and Doshita 1987, Mosterman and Biswas 1996) has been investigated. They emerge when the state reaches a patch boundary and is transitioned into a new mode with a patch that does not include the transferred state. This class of behavior is shown in Fig. 5(a) where the state is initialized inside the patch of mode α_1. The state evolves continuously as governed by f_{α_1} till it reaches the patch boundary defined by $\gamma_{\alpha_1}^{\alpha_2}$. The state is transferred to mode α_2 but here the mode transition $\gamma_{\alpha_2}^{\alpha_3}$ is immediately invoked and so mode α_2 never becomes 'real', i.e., it does not affect the energy distribution (Mosterman and Biswas 1998).

Another class of behaviors occurs when the state is transitioned to a point inside the patch of the new mode, but before it can proceed to evolve continuously, a consecutive mode transition is invoked. This is illustrated in Fig. 5(b). The reduced state space requires $x_{\alpha_2}^-$ to change discontinuously to x_{α_2}. This discontinuous change crosses the patch boundary, though, and $\gamma_{\alpha_2}^{\alpha_3}$ becomes active. In order to properly transfer the state between α_2 and α_3, the point where the patch boundary is crossed needs to be computed and this value, x_{α_2}, needs to be transferred to x_{α_3}.

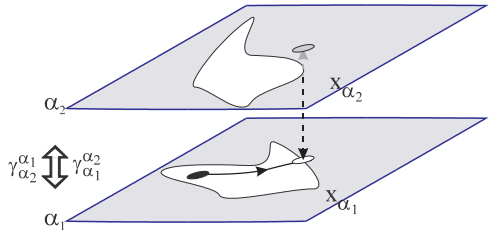
This behavior is called a *pinnacle* as it represents an isolated point in the state trajectory (Mosterman, Zhao, and Biswas 1998). Note that several such pinnacles may follow one another in time before the inside of a patch in a mode α_i is reached that is also contained by the dynamic space of the field f_{α_i}.

Physically, such a sequence of pinnacles cannot occur at the same point in time but when they follow one another in time, the point-interval partitioning of the time line fails. This leads to the system not being defined over infinitesimal periods of time, and gross behavior may violate physical laws such as conservation of momentum.

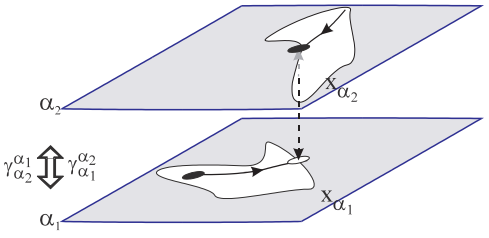
3.4 Revisiting Modes

Once sequences of mode changes may occur, models can be constructed that contain loops of changes, i.e., a previously visited mode is re-visited.

Two classes of behavior are illustrated in Fig. 6. In Fig. 6(a), the pathological case is shown that violates the *divergence of time* principle (Mosterman and Biswas 1998). Here, the state is initialized inside of the patch in mode α_1. It evolves continuously till it reaches the patch boundary as defined by $\gamma_{\alpha_1}^{\alpha_2}$. When the state x_{α_1} is then transferred to mode α_2, it is outside of the patch as defined by $\gamma_{\alpha_2}^{\alpha_1}$ (note the exchange in subscripts of α). This causes the state to be transferred back to α_1 where it is outside of the patch as defined by $\gamma_{\alpha_1}^{\alpha_2}$. Thus, a loop of discrete changes between modes arises. Note that a loop may involve any finite number of modes. Because these are instantaneous, no time elapses, and, therefore, the model stops evolving in time. In other words, time does not diverge. Since this behavior is not observed in physical systems, such models are considered anomalous.



(a) Non-divergent



(b) Sliding

Figure 6: Loops of Mode Transitions

Similar but different behavior is illustrated in Fig. 6(b). Here, after reaching the patch boundary in α_1, the state transfers onto the patch boundary in α_2 as defined by $\gamma_{\alpha_2}^{\alpha_1}$ (note again the exchange in subscripts of α). Because it is the patch boundary, the state transfers back to α_1 after an infinitesimal step in time. This step results in a value x_{α_1} that may be immediately inside the patch in α_1 as defined by $\gamma_{\alpha_1}^{\alpha_2}$ and so another infinitesimal step will transfer the state back to α_2. Note how left-closedness is violated in this particular instance of behavior. In general, an infinitesimal 'hysteresis' effect may be present to guarantee left-closedness again.

Far-fetched and pathological as it may seem, this behavior, referred to as *chattering* or *sliding mode* behavior, is actually aimed for by robust control design methodologies (Utkin 1992) as it is relatively insensitive to plant model parameter variations (e.g., it is used in anti-lock braking systems). Unlike the behavior in Fig. 6(a), here the state does continue to evolve in time and the divergence of time principle is satisfied. To efficiently derive the actual behavior along the *switching surface* as defined by the patches in mode α_1 and α_2 two methods exist: (i) equivalence of control (Utkin 1992) and (ii) equivalence of dynamics (Filippov 1960, Mosterman, Zhao, and Biswas 1999). Though there are classes of models for which these 'regularizations' result in the same behavior, in general they may differ.

Finally, another class of pathological behaviors can be identified, namely Zeno behavior, named after the Greek philosopher Zeno who studied the relation between points

and intervals, i.e., is an interval an infinite collection of points. Behaviors that are Zeno do progress in time by a non-infinitesimal value each time a mode transition occurs. However, this time reduces upon each transition as a converging series. For example, in case the time is halved upon each transition, the transition series converges to a limit value $t_f = \Sigma_i 2^{-i}$. Therefore, though time diverges locally, it does not do so globally.

It is now possible to compare the three mode re-visiting behaviors.

- *Divergence of time*: infinitely many discrete steps in zero time. Time remains the same.
- *Chattering*: infinitely small time steps. Evolves past any value in time.
- *Zeno*: infinitely many time steps in a finite, nonzero, time interval. Does not evolve past a limit point in time.

Unfortunately, the general hybrid dynamic systems literature is loose in its use of these terms (e.g., behavior that is locally not divergent in time is often called Zeno as well).

4 AN ALGEBRAIC VIEW

The general geometric overview in Section 3 will now be restricted to linear systems and illustrated by studying the behavior of the hydraulic cylinder presented in Section 2.

4.1 Projection Equations

The generalized state space of the cylinder model is two dimensional with states the flow through the relief valve, f_{rel}, and the velocity of the cylinder piston, v_p. Because the small oil parameters have been abstracted away, when at least one of the two valves is open the system has only one degree of freedom, i.e., the actual state space is one-dimensional. The subspace that consitutes this one-dimensional state space may change between the modes. In case both valves are closed, there is no flow of oil possible nor movement of the piston, and, therefore, the state space becomes of zero order.

To compute the projections between the different reduced dimension state spaces the following procedure is applied (Mosterman 2001, Mosterman 2000). For a general system of equations of the form

$$E\dot{x} + Ax + Bu = 0, \tag{7}$$

transformation derives a pseudo Weierstrass normal form to index 2 (since the nilpotency of the submatrix

$$\begin{bmatrix} 0 & \bar{E}_{22,12} \\ 0 & 0 \end{bmatrix}$$

is 2)

$$0 = \begin{bmatrix} \bar{E}_{11} & 0 & 0 \\ 0 & 0 & \bar{E}_{22,12} \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{\bar{x}}_1 \\ \dot{\bar{x}}_{2,1} \\ \dot{\bar{x}}_{2,2} \end{bmatrix} + \begin{bmatrix} \bar{A}_{11} & \bar{A}_{12,1} & \bar{A}_{12,2} \\ 0 & \bar{A}_{22,11} & \bar{A}_{22,12} \\ 0 & 0 & \bar{A}_{22,22} \end{bmatrix} \begin{bmatrix} \bar{x}_1 \\ \bar{x}_{2,1} \\ \bar{x}_{2,2} \end{bmatrix} + \begin{bmatrix} \bar{B}_1 \\ \bar{B}_{2,1} \\ \bar{B}_{2,2} \end{bmatrix} \begin{bmatrix} u \end{bmatrix} \tag{8}$$

where $\bar{E}_{11,11}$, $\bar{A}_{22,11}$, and $\bar{A}_{22,22}$ are of full rank. This allows computation of the initial conditions as

$$\begin{aligned} \bar{x}_1 &= \bar{x}_1^- + \bar{E}_{11}^{-1}\bar{A}_{12,1}\bar{A}_{22,11}^{-1}\bar{E}_{22,12}(\bar{x}_{2,2} - \bar{x}_{2,2}^-) \\ \bar{x}_{2,1} &= -\bar{A}_{22,11}^{-1}(\bar{B}_{2,1}u + \bar{E}_{22,12}\dot{\bar{x}}_{2,2} + \bar{A}_{22,12}\bar{x}_{2,2}) \\ \bar{x}_{2,2} &= -\bar{A}_{22,22}^{-1}\bar{B}_{2,2}u \end{aligned} \tag{9}$$

where $\bar{x}^-$ are the final values that are achieved in the previous mode. The values for $\bar{x}$ can then be transformed back to obtain initial values for x that are consistent with the subspace of the dynamic behavior, and thus the projection is determined.

4.2 The Hydraulic Cylinder

A model of the hydraulic cylinder is given in Eq. (6). Mode changes occur when the logical variables that model the state of the valves (either open or closed), s_{in} and s_{rel}, change their verity. As there are two valves, four modes ensue: in α_{00} both valves are closed, in α_{01} the relief valve is open and the intake valve is closed, in α_{10} the relief valve is closed and the intake valve is open, and in α_{11} both valves are open.

The projections in each of the modes can be symbolically computed. For example, when the intake valve is closed and the relief valve is open, mode α_{01}, there is an algebraic dependency between the generalized states v_p and f_{rel}, $-v_p - f_{rel} = 0$. This constitutes a subspace into which the state has to be projected if $-v_p - f_{rel} \neq 0$ and it is switched to this mode. The corresponding discontinuous change in v_p then becomes

$$v_p = \frac{1}{m_p + I_{rel}}(m_p v_p^- - I_{rel} f_{rel}^-). \tag{10}$$

The other projections are listed in Table 1.

The discontinuous change in mode α_{00}, however, may require an additional explicit change in state (Mosterman and Biswas 2000b). Consider the scenario illustrated in Fig. 7 where an initial control pressure, p_{in}, is applied and the piston starts moving. At a time, t_s, the intake valve closes and in the detailed model with the oil parameters (R_{oil} and C_{oil}), a quick pressure build-up results. This behavior operates on a time scale much faster than gross behavior, hence the double arrow heads in Fig. 7. The initial jump in

Table 1: Mode Specification Table

mode	projection
α_{00}	$v_p = 0$ $f_{rel} = 0$
α_{01}	$v_p = \frac{1}{m_p + I_{rel}}(m_p v_p^- - I_{rel} f_{rel}^-)$ $f_{rel} = \frac{1}{m_p + I_{rel}}(m_p v_p^- - I_{rel} f_{rel}^-)$
α_{10}	$f_{rel} = 0$
α_{11}	

pressure at $v_p = 4\ [m/s]$ is due to the abrupt change of oil flow into the cylinder when the flow through the intake valve jumps to 0. After this, the oil elasticity results in a further quick pressure build-up. This pressure build-up affects the velocity of the piston which ultimately would become 0 (conform the projections in Table 1). However, during the fast transient, the pressure may exceed the threshold pressure, p_{th}, and the relief valve may open.

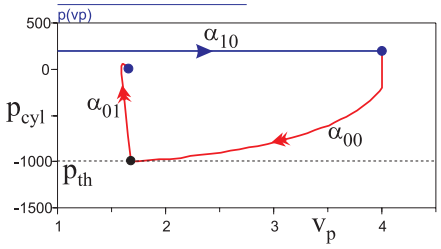


Figure 7: The Detailed Fast Transient May Cause a Further Mode Switch

In case the small oil parameters are abstracted away, the detailed information about whether the threshold is crossed and how much this affects v_p is not present anymore. Therefore, it has to be explicitly captured in a discontinuous state change function.

In addition, the switching logic has to be adapted as the pressure in the cylinder becomes of an impulsive nature when the detailed oil phenomena are abstracted away and there is a discontinuous change in v_p since $m_p \dot{v}_p = p_{cyl}$.

These modified specifications can be derived from the more detailed model with the oil parameters. The switching area in α_{00} was computed using the Symbolic Math Toolbox of `Matlab`® (`Matlab` 2003) and is shown in Fig. 8. First, a Taylor series approximation of the detailed 2^{nd} order continuous behavior was derived and then the maxima of this function were found. If the maximum absolute pressure exceeded the threshold, p_{th}, the switch occurred. This was done for a number of p_{cyl}^- and v_p^- data points to derive the patch sketched in Fig. 8. The actual function $v_p = g_{\alpha_{00}}^{\alpha_{01}}(v_p^-, p_{cyl}^-)$ to compute v_p from v_p^- and p_{cyl}^- is given in previous work (Mosterman and Biswas 2000b) and will not be re-iterated here. This relation is depicted in the phase space in Fig. 8. An interesting phenomena in this model is that the state transition function, $g_{\alpha_{00}}^{\alpha_{01}}$, does not equal the identity function (as assumed throughout Section 3).

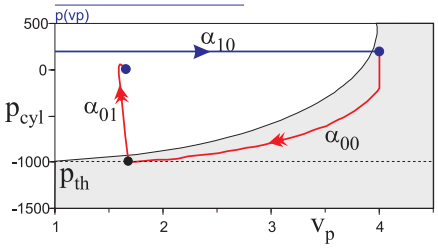


Figure 8: The Area in Which $\gamma_{\alpha_{00}}^{\alpha_{01}} > 0$

Note that this implies the user defined discontinuous change in state, v_p, takes place while the system is not in a patch. This differs from implicit discontinuities that have their jumps aborted when the patch boundary is crossed. In previous work (Mosterman and Biswas 2000a), the difference between these phenomena has been attributed to two different abstraction types: (i) *time scale* abstraction collapses the behavior during a relatively short period of time into a point of discontinuous change, (ii) *parameter* abstraction simply removes very small or large parameters. The explicitly defined discontinuous change then is the result of a time scale abstraction where the implicit discontinuity results from a parameter abstraction.

Once in mode α_{01}, there is an algebraic dependency between v_p and f_{rel}, and the system lives in a one-dimensional state space again. However, the value for the state, $x_{\alpha_{01}}^-$, as computed by the function $g_{\alpha_{00}}^{\alpha_{01}}$ may not be in this space and so a projection is required.

4.3 Semantics of x^-

In general, the use of x^- for re-initialization requires careful implementation and well-defined semantics. Typically, x^- is considered the *a priori* value around a discontinuous change and used as part of the input in Eq. (8). When no discontinuous change occurs, x^- equals the current value.

In order to design models with proper behavior, a mode with re-initialization using x^- should be immediately departed since no proper continuous behavior can be generated.

4.4 Putting it Together

The complete mode transition behavior can now be graphically depicted as illustrated in Fig. 9. Note that the pressure p_{cyl} is impulsive when v_p changes abruptly in the idealized model of Eq. (6). These impulses are not shown in Fig. 9 and so the value $p_{cyl} = -1000$ that is shown is never achieved. However, it provides the conceptual mapping between Fig. 9 and Fig. 7.

This scenario illustrates how several modes may be traversed before the state can proceed to evolve continu-

ously. It also illustrates how user defined discontinuous state changes may be part of a model and how these interact with the implicit discontinuous state changes that result from the required projections.

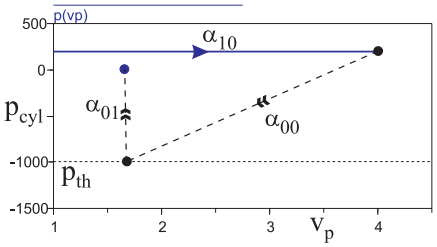


Figure 9: The Fast Transients as Instantaneous Changes

5 CONCLUSIONS

All this exemplifies the richness and complexity of mode transition behavior in hybrid dynamic systems. It illustrates that though an explicit representation such as hybrid automata (Alur, Courcoubetis, Henzinger, and Ho 1993) may be a powerful vehicle for analyses, it is by no means trivial to design such a hybrid automata for complex physical system models. It means that the expressiveness of the formalism is achieved by a rather significant conceptual investment by the model designer. As such, part of the analyses burden is put squarely on the shoulders of the model designer, making it difficult to gain acceptance in communities where the featured classes of complex behaviors are being dealt with on a day to day basis.

To eliminate this reluctance, future research efforts should focus on automated model complexity reduction and transformation into an underlying hybrid automata representation that may be hidden from the user.

6 ACKNOWLEDGMENTS

The author wishes to acknowledge extensive discussions with Gautam Biswas and Feng Zhao on the topic.

REFERENCES

Alur, R., C. Courcoubetis, T. A. Henzinger, and P.-H. Ho. 1993. Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems. In *Lecture Notes in Computer Science*, ed. R. Grossman, A. Nerode, A. Ravn, and H. Rischel, Volume 736, 209–229. Springer-Verlag.

Andersson, M. 1994. *Object-oriented modeling and simulation of hybrid systems*. PhD dissertation, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.

Benedetto, M. D. D., and A. L. Sangiovanni-Vincentelli. (Eds.) 2001, March. *Hybrid systems: Computation and control*, Volume 2034 of *Lecture Notes in Computer Science*. Springer-Verlag.

Breedveld, P. C. 1996, July. 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: Ecole Centrale de Lille.

Elmqvist, H., *et al*. 1999, December. Modelica^{tm}–a unified object-oriented langauge for physical systems modeling: Language specification. version 1.3.

Filippov, A. F. 1960. Differential equations with discontinuous right-hand sides. *Mathematicheskii Sbornik* 51 (1).

Lynch, N., and B. Krogh. (Eds.) 2000, March. *Hybrid systems: Computation and control*, Volume 1790 of *Lecture Notes in Computer Science*. Springer-Verlag.

`Matlab` 2003. `Matlab` *User's Guide*. 3 Apple Hill Drive, Natick, MA: The MathWorks, Inc.

Mosterman, P. J. 1999a, March. An Overview of Hybrid Simulation Phenomena and Their Support by Simulation Packages. In *Hybrid Systems: Computation and Control*, ed. F. W. Vaandrager and J. H. van Schuppen, Volume 1569, 164–177: Lecture Notes in Computer Science; Springer-Verlag.

Mosterman, P. J. 2000. Implicit Modeling and Simulation of Discontinuities in Physical System Models. In *The 4th International Conference on Automation of Mixed Processes: Hybrid Dynamic Systems*, ed. S. Engell, S. Kowalewski, and J. Zaytoon, 35–40.

Mosterman, P. J. 2001. `MAsim` – a hybrid dynamic systems simulator. Technical Report DLR-IB-515-01/07, Institute of Robotics and Mechatronics, DLR Oberpfaffenhofen, P.O. Box 1116, D-82230 Wessling, Germany.

Mosterman, P. J. 2002. `HyBrSim` – a modeling and simulation environment for hybrid bond graphs. *Journal of Systems and Control Engineering* 216: 35–46.

Mosterman, P. J., and G. Biswas. 1996, August. A Formal Hybrid Modeling Scheme for Handling Discontinuities in Physical System Models. In *AAAI-96*, 985–990. Portland, Oregon.

Mosterman, P. J., and G. Biswas. 1998, January. A theory of discontinuities in dynamic physical systems. *Journal of the Franklin Institute* 335B (3): 401–439.

Mosterman, P. J., and G. Biswas. 2000a. A comprehensive methodology for building hybrid models of physical systems. *Artificial Intelligence* (121): 171–209.

Mosterman, P. J., and G. Biswas. 2000b. Towards Procedures for Systematically Deriving Hybrid Models of Complex Systems. In *Hybrid Systems: Computation and Control*, ed. N. Lynch and B. Krogh, 324–337. Lecture Notes in Computer Science.

Mosterman, P. J., M. A. P. Remelhe, S. Engell, and M. Otter. 2002. Simulation for analysis of aircraft elevator feedback and redundancy control. In *Modelling, Analysis, and Design of Hybrid Systems*, ed. S. Engell, G. Frehse, and E. Schnieder, 369–390. Berlin: Springer-Verlag.

Mosterman, P. J., F. Zhao, and G. Biswas. 1998, July. An Ontology for Transitions in Physical Dynamic Systems. In *AAAI98*, 219–224.

Mosterman, P. J., F. Zhao, and G. Biswas. 1999. Sliding mode model semantics and simulation for hybrid systems. In *Hybrid Systems V*, ed. P. Antsaklis, W. Kohn, M. Lemmon, A. Nerode, and S. Sastry, 218–237. Springer-Verlag. Lecture Notes in Computer Science.

Nishida, T., and S. Doshita. 1987. Reasoning about discontinuous change. In *Proceedings AAAI-87*, 643–648. Seattle, Washington.

Utkin, V. I. 1992. *Sliding modes in control and optimization*. Springer-Verlag.

Vaandrager, F. W., and J. H. van Schuppen. (Eds.) 1999, March. *Hybrid systems: Computation and control*, Volume 1569 of *Lecture Notes in Computer Science*. Springer-Verlag.

Verghese, G. C., B. C. Lévy, and T. Kailath. 1981, August. A generalized state-space for singular systems. *IEEE Transactions on Automatic Control* 26 (4): 811–831.

AUTHOR BIOGRAPHY

PIETER J. MOSTERMAN is a Senior Research Scientist in Real-time and Modeling & Simulation Technologies at The MathWorks, Inc., in Natick, MA. Before, he held a research position at the German Aerospace Center (DLR) in Oberpfaffenhofen. He has a Ph.D. degree in Electrical and Computer Engineering from Vanderbilt University in Nashville, TN, and a M.Sc. degree in Electrical Engineering from the University of Twente, Netherlands. His primary research interests are in Hybrid Dynamic Systems and Computer Automated Multi-Paradigm Modeling (CAMPAM) with principal applications in training systems and fault detection, isolation, and reconfiguration. For this, he designed several simulation environments such as the *Electronics Laboratory Simulator* (nominated for The Computerworld Smithsonian Award), a first version of *Transcend*, *HyBrSim* (a paper on which received the Donald Julius Groen Prize), and `MAsim`. Dr. Mosterman is co-chairing the *14th International Workshop on Principles of Diagnosis*, he has served on the industrial advisory board and program committee of several conferences and workshops, and he has organized special sessions at the *2001 IEEE Conference on Control Applications*, the *2000 IEEE International Symposium on Computer Aided Control System Design* (CACSD), and *Eurosim '98*. He is currently a member of the *IFAC Technical Committee on CACSD*, chair of the *IEEE CSS Action Group on Hybrid Dynamic Systems for CACSD*, Mechatronics area editor of *Simulation*, and associate editor of *IEEE Transactions on Control Systems Technology* and the *International Journal of Applied Intelligence*. He is also guest editor of special issues of *ACM Transactions on Modeling and Computer Simulation* and *IEEE Transactions on Control Systems Technology* on the topic of CAMPAM. His e-mail address is <pieter_j_mosterman@mathworks.edu> and his web page is <http://www.xs4all.nl/~mosterma>.