PERGAMON

Control Engineering Practice 6 (1998) 511–521

# A hybrid modeling and verification paradigm for embedded control systems

Pieter J. Mosterman[a,1], Gautam Biswas[b,*,2], Janos Sztipanovits[c,3]

[a] Institute of Robotics and System Dynamics, DLR Oberpfaffenhofen, D-82230 Wessling
[b] Department of Computer Science, Vanderbilt University, Nashville, TN 37235
[c] Department of Electrical and Computer Engineering, Vanderbilt University, Nashville, TN 37235

Received 8 October 1997; accepted 9 February 1998

**Abstract**

Contemporary process control includes continuous and discrete components. At the lowest level, continuous PID controllers are used for actuation and control. At a higher level, supervisory control mechanisms are used to select appropriate control algorithms for the different modes of system operation to achieve optimal or near-optimal control. Modeling and analysis of such combined discrete and continuous components requires *hybrid* modeling techniques. This paper presents a hybrid modeling paradigm, and discusses its execution semantics, which are based on the principles of *invariance of state* and *temporal evolution of state*. The modeling and simulation methodology is used to analyze the control behavior of dynamic physical systems, and a model-verification technique based on *divergence of time* demonstrates possible applications in design tasks. © *1998 Elsevier Science Ltd. All rights reserved.*

*Keywords:* Command and control systems; formal methods; hybrid systems; verification; compositional modeling

## 1. Introduction

The complexity of large-scale embedded control systems (Fig. 1) that incorporate computer-based technologies to assist in design, manufacturing, analysis, control, and monitoring has increased the significance and the need for models that can accurately simulate and verify system behavior. Embedded systems typically operate in multiple configurations during normal operation. For example, the Airbus A-320 fly-by-wire system utilizes a number of modes in normal operation: *take off, cruise, approach*, and *go-around* (Sweet, 1995). Process-control operations are implemented with Programming Logic Arrays and software modules. These digital control mechanisms are discrete, and coexist with low-level continuous PID control (Garcia et al., 1995; Mosterman and Biswas, 1997b). Therefore, modeling schemes for process control are required to capture both discrete control and continuous process characteristics. Because of their mixed continuous/discrete nature, these systems are referred to as *hybrid systems*.

Consider the cooling system of a fast breeder reactor, shown in Fig. 2. The main pump, used to maintain an adequate flow of coolant, is driven by a synchronous *ac* motor. The flow rate depends on the motor revolutions per minute (*rpm*), which is determined by the frequency of the *ac* signal. To achieve sufficient torque for this flow rate, a continuous PID controller determines the power supplied to the motor. Excessive pressures may cause leaks in the piping that result in violent chemical reactions because of liquid sodium exposure to air. As a check, actuated valves installed at different points in the cooling loop reduce pressures by activating a bypass alarm loop. These valves also serve to block parts of the system, to minimize loss of coolant in case of an emergency.

Physical systems are inherently continuous. A precise valve model would include complex nonlinear relations. However, opening and closing of the valve can be achieved in seconds, whereas the time constants associated with overall system behaviors are in the order of

*Corresponding author. E-mail: biswas@vuse.vanderbilt.edu

[1] Pieter J. Mosterman is supported by a grant from the DFG Schwerpunktprogramm KONDISK.

[2] Gautam Biswas is supported by grants from PNC, Japan and Hewlett-Packard Labs, USA.

[3] Janos Sztipanovits is with the Measurement and Computing Systems Lab, Vanderbilt University.

*PII* S0967-0661(98)00045-8

minutes. So, the use of continuous valve models introduces overwhelming and unnecessary detail in the analysis, which can be avoided by representing valve behavior as discrete but instantaneous *on–off* switches.

Hybrid modeling techniques provide the basis for a comprehensive study of the performance of embedded control systems that includes the effects of implementation choices such as sampling rates and computation order in software (Wijbrans, 1993). Mosterman and Biswas (1996, 1997b, 1998) have developed a theory of hybrid modeling for dynamic physical systems to simplify the analysis of complex nonlinear behaviors that includes: (i) time-scale abstraction of fast nonlinear behaviors, and (ii) parameter abstraction that ignores small parasitic component parameters. The resultant hybrid models are made up of two components:

1. a differential equation model of continuous system behavior associated with the operational modes of the system, and
2. a discrete-event model, based on finite state automata for handling mode transitions, and correctly transferring the system state vector from one mode to another through a sequence of transitions. The principle of *conservation of state* governs the transfer of the state vector between two modes (Mosterman and Biswas, 1997a).

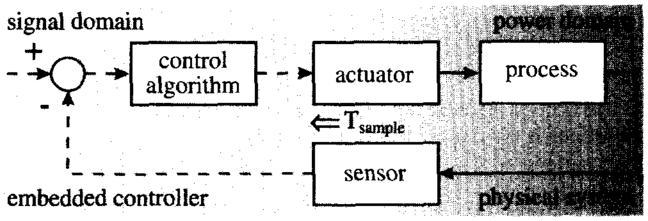

Fig. 1. An embedded control system.

This paper extends the formal hybrid modeling scheme to embedded control systems. Section 2 presents the hybrid system model. Section 3 discusses the set of principles that hybrid models of physical systems have to obey. This includes the principles of *invariance of state* and *temporal evolution of state*. The principle of *divergence of time* verifies that the system does not get stuck in a loop of instantaneous changes. Section 4 presents an application that ensures that the designed control law for the reactor cooling system in Fig. 2 produces desired behaviors. Section 5 presents conclusions of this research.

## 2. Hybrid modeling formalism

Modeling paradigms like finite state machines (Aho et al., 1974; Kohavi 1978), Petri nets (Murata, 1989), Statecharts (Harel, 1987), condition/event systems (Kowalewski and Preußig, 1996), CSP (Davies and Schneider, 1989; Hoare, 1978), and structured methods (Hatley and Pirbhai, 1988; Ward and Mellor, 1985), have been extensively employed for analyzing discrete-event dynamic systems. However, models for embedded control of complex physical systems necessarily require a modeling paradigm that combines continuous differential equation models with these discrete modeling components.

The continuous system, a chemical process, aircraft, a nuclear plant, or an automobile engine, may operate in multiple distinct modes (e.g., auto engine controllers may switch between different control programs as a function of the engine *rpm*). The switching or mode changes may be modeled by discrete-event switching logic, but within each mode the system exhibits continuous behavior. In reality, these mode changes are invoked by discrete actuators or physical phenomena that occur when variable values cross prespecified thresholds. Because these are local effects it is advantageous from the modeling and

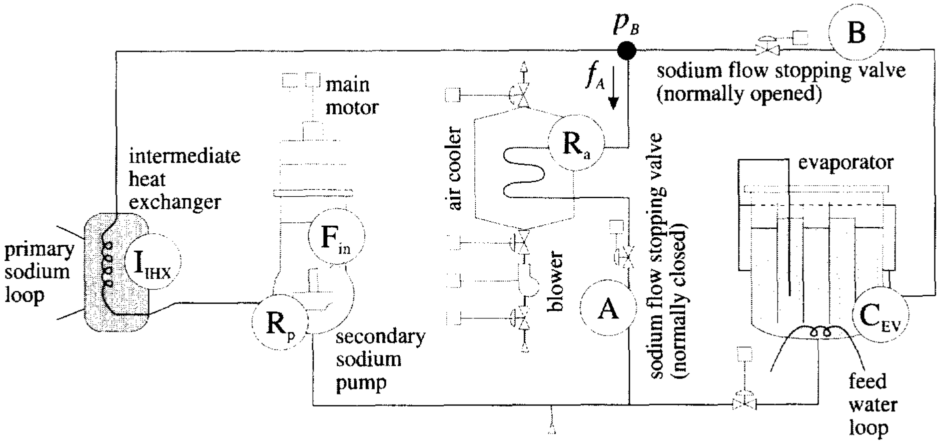

Fig. 2. Continuous and discrete process control.

simulation viewpoints to reduce the complexity of analysis by utilizing a localized discrete modeling paradigm. The states of all the local discrete model parts can then be composed to constitute the global mode of behavior.

Previous work (Mosterman and Biswas, 1996; Nishida and Doshita, 1987; Otter, 1994) has shown that dynamic coupling of system variables may cause a sequence of local discrete state changes before a mode of continuous behavior is arrived at. For example, consider the cooling system in Fig. 2, where valve *A* opens when the pressure $p_B$ exceeds a critical value. If valve *A* is initially closed and valve *B* closes, a quick build-up of pressure causes *A* to open. A continuous model of this dynamic build-up of pressure, though complex, operates on a time scale much smaller than overall behavior of the system. Behavior analysis is simplified by modeling the opening of valve *A* to occur immediately after valve *B* is closed. The mode where valve *A* and *B* are both closed is instantaneously departed from, and assumed not to have a representation in real time. Therefore, it is termed *mythical* (Mosterman and Biswas, 1995; Nishida and Doshita, 1987). The instantaneous changes occur because of abstractions; e.g., the equations governing the opening and closing of the valves, and distributed pipe capacity are ignored. It is critical to assign semantics to the abstract instantaneous phenomena so that the overall system behavior is modeled accurately.

In other work (Alur et al., 1994; Guckenheimer and Johnson, 1995; Nicollin et al., 1991) global switching logic is introduced to model transitions from one continuous mode to another, eliminating the intermediate mythical modes that occur as a chain of instantaneous local switching effects. For small systems, global mode-switching behavior can be specified, but for larger systems all possible permutations of local switching changes have to be considered in computing the global system mode changes. This is computationally intractable.

### 2.1. Architecture

Figure 3 shows the general hybrid architecture of a controlled physical process. The process and its continuous controller represent the continuous components of the system. Configuration changes in the system can be attributed to three phenomena:

(i) physical system signals crossing prespecified threshold values; these are modeling artifacts that can be mainly attributed to time scale and parameter abstractions incorporated in the continuous system model,
(ii) explicit control signals that activate the closed loop controller, and
(iii) external, open-loop control. The events generated by these phenomena are labelled, $\sigma_p$, $\sigma_c$, and $\sigma_x$, respectively, in Fig. 3.

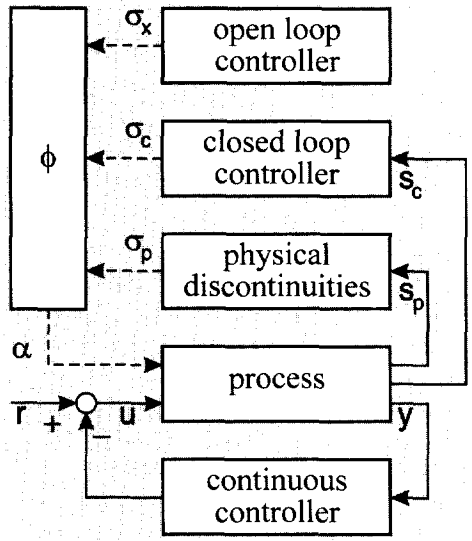

Fig. 3. Hybrid control model of a physical process.

It is assumed that the input signal *u* is continuous. Discontinuous changes in the input (e.g., step input) and changes in the low-level continuous controller are modeled in the open loop-controller by deactivating the current input signal and activating the new input.

### 2.2. The continuous model

Physical system behavior is governed by the principles of conservation of energy and continuity of power (Paynter, 1961). Dynamic systems are typically described by a state-space representation and ordinary differential equations (ODEs),

$$\dot{x}(t) = f(x(t), u(t), t), \tag{1}$$

where $x \in X$ are the system state variables, and $u \in U$ describe the external input to the system. As discussed earlier, the system under consideration can operate in multiple modes of continuous behavior, and each mode is represented by a different behavior model. The continuous system model in operational mode $\alpha \in \aleph$ is defined as: $\dot{x}(t) = f_\alpha(x(t), u(t), t)$, where $t \in \Re$ represents time, $X \in \Re^m$ is the continuous state vector, and $U \in \Re^p$ is the vector of input signals. There is one and only one field, $f_{\alpha_i}$, for each mode of continuous operation $\alpha_i$.

### 2.3. The discrete model

Discrete events can be categorized as time events and state events (Broenink and Weustink, 1996). Time events result from digital control, where discrete actuation is generated by the control algorithm at a point in time. State events are generated by the process. When certain signal values cross prespecified thresholds, mode transitions are invoked. These discrete changes are modeled by a transition function, $\phi$, and transitions are invoked by events in a set $\Sigma$ (see Fig. 3).

Systematic derivation of $\phi$ requires the consideration of a number of independent state machines that control local switching effects. For example, in the liquid sodium cooling system in Fig. 2, the two valves can be modeled by two finite state machines that control their *on* and *off* states. The global mode of the system is determined by considering combinations of the states of the independent state machines. Some of these combined modes may not occur in reality, but they are traversed during behavior generation as the system transitions from one real mode to another. A primary contribution of this paper is the establishment of execution semantics that handle these sequences of mode changes correctly.

The discrete modeling paradigm can be implemented by Petri nets or finite-state machines. It has three components:

- $I = \{\alpha_0, \ldots, \alpha_k\}$, the set of states describing operational modes of the system.
- $\Sigma = \{\sigma_0, \ldots, \sigma_l\}$, the set of events that can cause state transitions. Events are generated by the physical process, the closed-loop controller, and by external, open-loop control signals, i.e., $\Sigma = \Sigma_p \times \Sigma_c \times \Sigma_x$.
- $\phi : I \times \Sigma \to I$, a discrete state-transition function that defines the new mode after an event occurs.

### 2.4. Interaction

Lygeros et al. (1994) have shown that independent determination and proofs about the continuous and discrete behaviors in a hybrid model do not constitute proofs of the correctness of their combined effects. To enable hybrid system verification, a formal specification of the interaction between the continuous and discrete models has to be established. Interaction between the continuous and discrete parts is specified by (i) discrete events triggered by variable value changes in the continuous model, and (ii) a change of mode by the discrete model. Figure 4 illustrates a block-diagram model of the interaction and the parameters that govern the interaction process. The interaction can be specified by:

- $S \in \Re^n$, the signals that result in event generation.
- $h : X \times U \times I \to S$, computes the signals in a given mode as a function of the state and input variables.
- $g : X \times I \to X^+$, transfers the *a priori* continuous state vector, $X$, immediately before switching to the *a posteriori* state vector, $X^+$, that represents the initial state in the new mode, $\alpha \in I$. This may result in discontinuous changes.
- $\gamma : S \times S^+ \to \Sigma_s$, where $\Sigma_s = \Sigma_p \times \Sigma_c$, generates discrete events from the signal values. These signal values may be computed from the *a priori* state vector, $S$, or the *a posteriori* state vector, $S^+$.

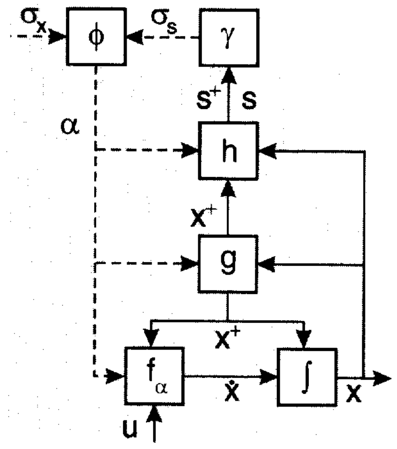

Fig. 4. A general hybrid system.

Figure 4 depicts the interaction process. The function $\gamma$ generates discrete events when signals, $s \in S$, cross prespecified threshold values. The output function, $h$, computes the values of these signals from the continuous state vector in a mode, $\alpha$. The generated events applied to the model may result in a mode change in the continuous operation of the system.

When the system switches modes, the continuous state vector of the system may change. The function $g$ maps the state vector from the last continuous mode to the new continuous mode as a function of the new mode. Mode transitions are assumed to occur at very fast time scales, and this mapping function has to satisfy the principle of conservation of state (charge, momentum, etc.) (Mosterman and Biswas, 1997a). A natural choice for state variables are the signals associated with the energy storage elements in a physical system (e.g., see bond-graph modeling techniques (Karnopp et al., 1990)). If the size of the state vector decreases from one mode to another, implying that state variables have become dependent, discontinuous changes in state variable values may occur, which may lead to a loss of energy in the system (Mosterman and Biswas, 1998). If configuration changes remove dependencies among energy storage elements, the state vector increases in size without discontinuous changes in existing values.

The continuous and discrete model components and their interface can be put together to define the complete hybrid system model as a 9-tuple

$$H = \langle I, \Sigma, \phi, X, U, f_\alpha, g, h, \gamma \rangle. \tag{2}$$

The elements conform to the specification by Lennartson et al. (1996), but the methodology described in this paper provides a more complete definition for the switching function $\gamma$ and its arguments. As a next step, execution semantics are developed for the block diagram structure in Fig. 4 to ensure consistent behavior generation from hybrid system models.

### 2.5. Execution semantics

Continuous system behavior is governed by the active field, $f_{\alpha_k}$, in mode $\alpha_k$. At time $t_s$, signal values corresponding to the continuous state vector $x_{\alpha_k}$ in mode $\alpha_k$ may invoke an event, $\sigma$. At this point, time evolution is suspended, and the transition function $\phi$ is activated to derive a new mode, $\alpha_{k_1}$, (see Fig. 5). The $g$ function derives the *a posteriori* state vector $x^+$ from $x_{\alpha_k}$, given the new mode $\alpha_{k_1}$. The signal values corresponding to $x^+$ with the new mode, $\alpha_{k_1}$, may immediately generate a new event, causing a consecutive mode transition to $\alpha_{k_2}$. The *a posteriori* state vector, $x^+$, is again determined from $x_{\alpha_k}$, but now as a function of $\alpha_{k_2}$. Further mode changes may occur at time point $t_s$ until a mode, $\alpha_m$, with state vector, $x^+$, is arrived at that does not generate another immediate transition. This mode and the corresponding state vector are then established as the new continuous state description and initial condition, respectively, and the evolution of behavior in real time resumes. The new state vector, $x_{\alpha_m} = x^+$, is derived from the original state, $x_{\alpha_k}$; it does not depend on the path of intermediate model configurations. Therefore, $\alpha_m$ is called a "real" mode whereas the $\alpha_{k_i}$'s are termed "mythical" (Mosterman and Biswas, 1995; Nishida and Doshita, 1987).

After the new real mode, $\alpha_m$, is established at a point in time, $t_s$, the state vector is updated to $x^+$, which may result in a new sequence of mode changes (Fig. 5). This sequence terminates in real mode $\alpha_n$ and transitions the system from a point to an interval where the newly activated system description, $f_{\alpha_n}$, describes continuous evolution with $x_{\alpha_n} = x^+$ as the initial condition, and evolution of behavior in real time resumes. In real time, the active real modes ($\alpha_k$, $\alpha_m$, and $\alpha_n$) follow each other immediately.

$$\overbrace{\langle \leftarrow, t_s \rangle}^{\alpha_k}, \overbrace{[t_s]}^{\alpha_m}, \overbrace{\langle t_s, \rightarrow \rangle}^{\alpha_n}. \qquad (3)$$

Mythical modes that were traversed during sequences of discontinuous changes have no representation on the real timeline. Figure 4 shows this as two closed loops, $\phi \rightarrow h \rightarrow \gamma$ and $\phi \rightarrow g \rightarrow h \rightarrow \gamma$, between the continuous and discrete domains.

## 3. Validity of hybrid system behavior

The key to developing useful and correct hybrid system models is to define formal semantics that govern the interaction between the continuous and discrete components of the model so that systematic behavior-generation methodologies that do not violate overall physical system principles can be developed. Previous work developed a set of principles: invariance of state (Mosterman and Biswas, 1995), conservation of state (Mosterman and Biswas, 1997a), temporal evolution of state (Mosterman, 1997), and divergence of time (Mosterman and Biswas, 1996) that govern overall system behavior evolution during discrete transitions from a continuous mode (possibly) through a sequence of instantaneous mythical modes, to a new continuous mode of operation where time continues to evolve along the real time-line. Some key aspects of this work have been (i) projecting local model switches to global configuration changes in the system model, and (ii) computing the initial state in a new continuous mode of operation after discrete transitions have occurred. This paper demonstrates the use of semantics based on invariance of state, temporal evolution of state, and divergence of time to maintain consistency of the interactions.

### 3.1. Invariance of state

Mosterman and Biswas (1998) have demonstrated that a particular continuous state vector of a physical system model, $p_{\alpha_0}$, that represents the stored energy in the physical buffer elements of the system, e.g., springs, tanks, and inertia elements such as masses, is invariant across consecutive, mythical changes in the operational mode of the system.

#### 3.1.1. Example

To illustrate, consider the simple diode-inductor circuit in Fig. 6, whose behavior resembles the operation of the valves in the alarm loop of the secondary sodium cooling system in Fig. 2. The pressure generated by the synchronous *ac* motor is represented by the battery, $V_{in}$. In the cooling system, there is a build-up of flow momentum in the helical coil in the intermediate heat

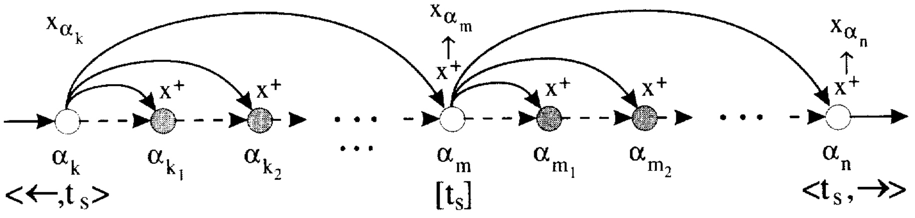

Fig. 5. System state is derived from the original state vector.

exchanger, which corresponds to the inductor, $L$, in the electrical circuit. When the pressure at $p_B$ becomes too large, the valve $A$ opens, represented by the diode, $D$. The pressure drop across the pipes is modeled by resistances $R_1$ and $R_2$.

Initially the switch is closed, the inductor builds up a flux and the diode is inactive (mode $\alpha_{10}$, shown in the top-left corner of the first box in Fig. 6). Figure 7 shows the inductor voltage and current in this mode for $0 \leq t \leq 0.075$ ms with parameter values $V_{in} = 5$ V, $R_1 = 330\,\Omega$, $R_2 = 22\,\Omega$, $L = 5$ mH. When the switch is suddenly flipped to its *off* state at $t = 0.075$ ms, the system configuration changes to mode $\alpha_{00}$ (Figs 6 and 8). This forces the current through the inductor to 0 instantaneously, which would create a large, negative voltage drop across the inductor to release its energy stored in the form of flux. However, when the voltage drop across the diode becomes larger than its threshold voltage (usually 0.6 V), the diode comes on. This causes a model switch to mode $\alpha_{01}$, and the inductor draws current through the diode path, and discharges in a continuous manner. Therefore, no voltage spike occurs in reality. To generate correct behavior, in the mode $\alpha_{01}$ the initial value of current through the diode must equal the current through the inductor at the moment the switch was opened (the last continuous mode). If it were based on the intermediate, mythical configuration, $\alpha_{00}$, where the current through the inductor was forced to 0 because of the sudden open circuit, the initial value of the continuous state vector in the final configuration would indicate that the current through the diode was 0. This would conflict with the real, observed behavior of the system.

The simulation shows that the configuration where the switch is open (OFF) and the diode inactive (OFF) is never achieved in real time (see Fig. 8). The opening of the switch immediately activates the diode through which the inductor discharges. However, this behavior is easier to infer if one goes through the intermediate mythical mode where both components are inactive ($\alpha_{00}$). The alternative approach, the *method of assumed states*, requires exhaustive analysis, and becomes intractable for larger systems (Kassakian et al., 1991). The plot of the inductor voltage in Fig. 7 shows the voltage drop at $t = 0.075$ ms due to the configuration change as a steep slope rather than a discontinuous jump. This is an artifact of the fixed step size used in the simulation. The plot of the inductor current in Fig. 7 illustrates that the initial inductor current in mode $\alpha_{01}$ equals the current just before $\alpha_{10}$ is departed from, which conforms to physical reality.

### 3.1.2. Linear systems

The result illustrated above forms the basis for the general principle of invariance of state, that applies to any state vector for linear systems.

**Conjecture 1.** *For a hybrid physical system model, the particular state vector, $p_{\alpha_0}$, is invariant across mode changes.*

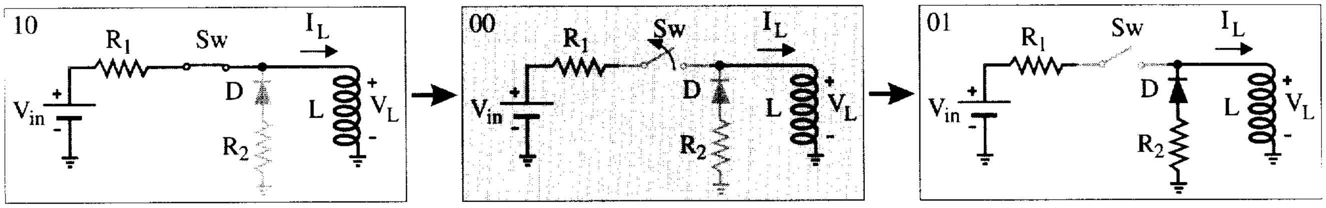

Fig. 6. A mythical mode in a physical system.

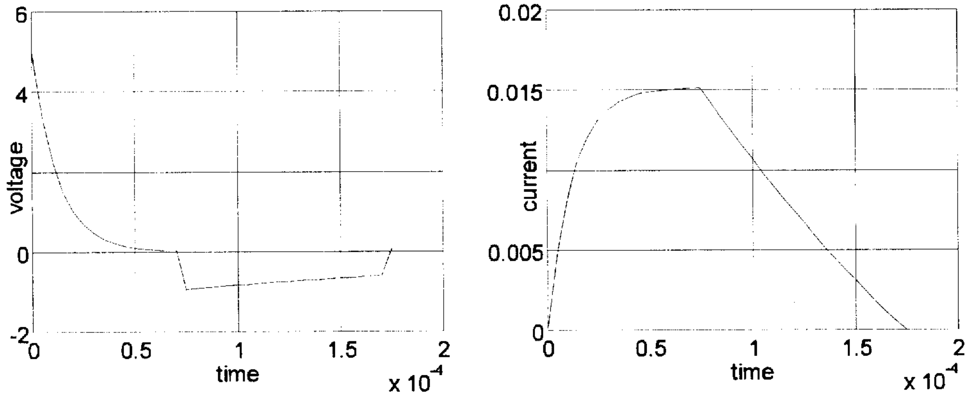

Fig. 7. Inductor voltage and current in the diode-inductor circuit.

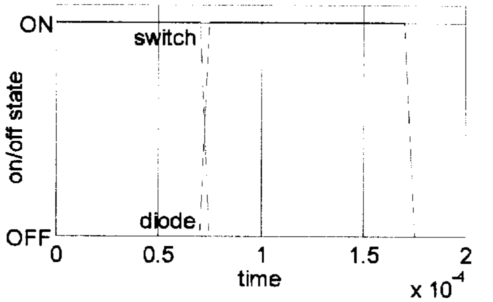

Fig. 8. Local configuration changes in the diode-inductor circuit.

**Lemma 1** (Invariance of State). *Any vector that represents the state of a linear physical system is invariant across mode changes.*

The proof for the lemma is presented elsewhere (Mosterman et al., 1997b).

As an illustration, the continuous state vector for the diode-inductor circuit in Fig. 6 can be defined as the inductor current, $I_L$, or the inductor voltage, $V_L$. There is an algebraic relation between the states defined by the mode of the system

$$T: \begin{cases} V_L = 0 & \text{in } \alpha_{00} \\ V_L = I_L R_L + V_D & \text{in } \alpha_{01} \\ V_L = -I_L R_L + V_{in} & \text{in } \alpha_{10}. \end{cases} \tag{4}$$

The inductor current chosen as the state variable is associated with the primary energy storage element in the circuit for all modes, and, therefore, provides a consistent mapping across mode changes. Mosterman and Biswas (1996) have shown that in such cases it is invariant, i.e., $I_L^+$ can be expressed in terms of $I_{L,0}$, the current before switching, as $I_L^+ = I_{L,0}$ independent of any intermediate modes that the system transitions through.[4]

If the voltage drop across the inductor, $V_L$, is chosen as the state variable, and its value before switching is $V_{L,0}$, then, using invariance of state of the special state variable, $I_L$, it can be established that $I_L^+ = g(I_{L,0}) = I_{L,0}$ and given the mapping $T_{\alpha_{01}}$: $V_L^+ = I_L^+ R_L + V_D$,

$$V_L^+ = I_{L,0} R_L + V_D. \tag{5}$$

If $I_{L,0}$ is expressed in terms of $V_{L,0}$ using the inverse mapping $T_{\alpha_{10}}^{-1}$: $I_{L,0} = -\frac{1}{R_1} V_{L,0} + \frac{1}{R_1} V_{in}$, the value of the new continuous system state can be expressed in terms of variables before switching by $T_{\alpha_{01}} \circ T_{\alpha_{10}}^{-1}$,

$$V_L^+ = \frac{R_2}{R_1} V_{L,0} + \frac{R_2}{R_1} V_{in} + V_D. \tag{6}$$

This illustrates that the value of any continuous state vector in a linear system after a sequence of instantaneous transitions is independent of the intermediate, mythical modes. It is completely determined by the state vector in the original mode and the new mode of continuous operation.

**Conjecture 2.** *This result may be extendable to nonlinear systems, where $T_{\alpha_0}^{-1}$ can be computed uniquely. If that is the case, again the composed function $(T_{\alpha_n} \circ g \circ T_{\alpha_0}^{-1})$ is path-invariant, and depends only on the states $\alpha_0$ and $\alpha_n$.*

In general, it may be hard to prove that $T_{\alpha_0}^{-1}$ exists and is unique for a nonlinear system.

### 3.2. Temporal evolution of state

It is clear that mode changes caused by discrete transitions in the system model may result in configuration changes in the system, as a result of which storage elements may become dependent, i.e., one cannot assign independent initial values to the variable associated with these storage elements. As a result, the order of the system state vector changes. These dependencies may be among variables that were independent before the mode change, or between independent state variables and exogeneous variables. Dependencies among the state variables cause discontinuous changes in variable values that can only occur at well-defined points in time (Mosterman, 1997). Continuity of power requires that the functions on the left and right intervals about the point of discontinuity (see Fig. 5) be well defined. Therefore, configuration changes result in piecewise continuous behaviors with a countable number of *simple* discontinuities with derivable limit values (Rudin, 1976). These observations can be formalized as follows.

**Conjecture 3.** *A hybrid system is piecewise continuous.*

**Lemma 2** (Temporal Evolution of State). *Continuous state variable values have to be continuous in left-closed intervals.*

The proof for this lemma appears in Mosterman et al. (1997b).

The required left closed intervals of state variable values in time determine that discontinuous changes in the state vector can only occur when the system transfers from an interval to a point. Note that this does not prohibit configuration changes from occurring when the system transfers from a point to an interval, as long as the number of degrees of freedom of the system does not decrease. In Fig. 4 this corresponds to the $\phi \rightarrow h \rightarrow \gamma$ loop only because the $\phi \rightarrow g \rightarrow h \rightarrow \gamma$ loop would require

[4] Note that $g(x) = x$ is a special case.

a discontinuous change in $x$ to $x^+$, caused by a reduction in the degrees of freedom in the system.

### 3.3. Divergence of time

Once a transition occurs from a continuous mode, the signals $s^+$ corresponding to the new state vector, $x^+$ (see Fig. 4), may immediately generate a new event that causes another discrete transition to a new mode. This recursive sequence of transitions ends when a new model configuration is established that does not generate a new event. Discontinuous mode changes occur at instants in time, therefore, if the sequence of changes results in a loop, the implication is that the resultant system behavior no longer progresses in real time, a situation that cannot occur in reality. Therefore, divergence of time represents an important principle that has to be satisfied by hybrid system models (Henzinger et al., 1994; Mosterman and Biswas, 1998).

To ensure that hybrid model behavior necessarily diverges in time, it has to be proved that transitions in the discrete model always terminate in a new real mode where a field, $f_\alpha$, governs the continuous evolution of system behavior. This analysis is complicated because the continuous signals that generate discrete events when their values cross prespecified thresholds, may themselves change discontinuously from one mode to another. An algorithm that tracks discontinuous signal value changes and the corresponding events that occur is computationally quite complex.

As discussed, discontinuous mode changes are tracked by invoking the principle of invariance of state, which requires that signal values in the new mode be computed from the state vector in the last real mode. Since this state vector is not affected by future configuration changes, it can be applied to establish a necessary condition for divergence of time. However, the event-generation conditions are typically specified in terms of signals, and, therefore, a mapping has to be applied to express the event conditions in terms of the original state vector, based on the inverse relation of $g$ and $h$.

In general, system verification proceeds by applying $\gamma$ to $\phi$ to determine which conditions cause transitions between modes. Then $h$ is used to express these relations in terms of the continuous state variables, and $g$ is applied to translate the conditions in terms of the switching invariant applied to the original state. This is illustrated next.

## 4. The secondary cooling system

An analysis of the alarm control of the secondary sodium cooling system in Fig. 2 shows that even simple control laws may contain fallacies not recognized during design. Application of the invariance of state and divergence of time principles helps detect the problem.

### 4.1. Specifying the system

The main motor of the cooling system drives a pump which establishes a flow-rate $F_{in}$, and a continuous controller ensures sufficient torque is available to maintain the desired flow rate for the liquid sodium coolant. Pump losses are represented by the dissipation parameter, $R_p$. The coolant is pumped through a coil in an intermediate heat exchanger with inertia, $I_{IHX}$, and this is responsible for building up flow momentum. An evaporator vessel with capacitance $C_{EV}$ transports heat from the heat exchanger to a steam water loop, where it drives a turbine to produce electricity. A discrete controller acts on the two valves, $A$ and $B$. In normal operation, $A$ is closed and $B$ is open. In an alarm situation, valve $B$ may be closed by supervisory control and the closed-loop controller is required to activate the alarm path with resistance $R_a$ by opening valve $A$ until it is safe to stop the flow of coolant completely.

As an exercise, the hybrid model of the system is developed and the verification mechanism is applied to ensure that the model is consistent. A possible state vector in this system is made up of the flow momentum, $x_1$, in the coil of the intermediate heat exchanger, $I_{IHX}$ and the stored coolant, $x_2$, in the evaporator, $C_{EV}$, i.e.,

$$x = [x_1 \; x_2]^T. \tag{7}$$

The input to the system is the input flow, $F_{in}$, i.e.,

$$u = F_{in}. \tag{8}$$

The discrete model is defined by the states of the two valves in the system resulting in four modes, $I = \{\alpha_{00}, \alpha_{01}, \alpha_{10}, \alpha_{11}\}$, where

$$\begin{aligned}\alpha_{00} &= \{A_{closed}, B_{closed}\},\\ \alpha_{01} &= \{A_{closed}, B_{open}\},\\ \alpha_{10} &= \{A_{open}, B_{closed}\},\\ \alpha_{11} &= \{A_{open}, B_{open}\}.\end{aligned} \tag{9}$$

In each of these modes, behavior is governed by a system of differential equations. When both valve $A$ and valve $B$ are closed ($\alpha_{00}$), there are no independent states, i.e., the state vector $x = \emptyset$. Otherwise,

$$f_{\alpha_{01}}: \dot{x} = \begin{bmatrix} -\dfrac{R_p}{I_{IHX}} & -\dfrac{1}{C_{EV}} \\ \dfrac{1}{I_{IHX}} & 0 \end{bmatrix} x + \begin{bmatrix} R_p \\ 0 \end{bmatrix} u \tag{10}$$

$$f_{\alpha_{10}}: \dot{x}_1 = \left[ -\dfrac{R_p + R_a}{I_{IHX}} \right] x_1 + [R_p] u \tag{11}$$

$$f_{\alpha_{11}}: \dot{x} = \begin{bmatrix} -\dfrac{R_p}{I_{IHX}} & -\dfrac{1}{C_{EV}} \\ \dfrac{1}{I_{IHX}} & -\dfrac{1}{R_a C_{EV}} \end{bmatrix} x + \begin{bmatrix} R_p \\ 0 \end{bmatrix} u. \tag{12}$$

### 4.2. Designing the control law

In normal operation, $\alpha_{01}$, valve $A$ is closed and valve $B$ is open. Valve $A$ is controlled by a closed-loop discrete controller, and valve $B$ by an open-loop discrete controller. When the open loop control closes $B$ by generating $\sigma_{B\to off}$, the coolant flow in that segment of the pipe becomes 0 abruptly, causing a large pressure value for $p_B$. To prevent damage to the piping, this large increase in pressure is kept in check by the closed-loop controller, which opens valve $A$ (i.e., $\sigma_{A\to on}$) to create a release path when $p_B > p_{th}$. As time progresses, the pressure falls below $p_{th}$, and the controller closes the valve $A$ (i.e., $\sigma_{A\to off}$). The complete event set is

$$\Sigma = \{\sigma_{A\to on}, \sigma_{A\to off}, \sigma_{B\to off}\}. \quad (13)$$

The closed-loop controller generates events $\Sigma_c$, which are specified by $\gamma$

$$\gamma: \begin{cases} p_B > p_{th} \to \sigma_{A\to on} \\ p_B \le p_{th} \to \sigma_{A\to off}. \end{cases} \quad (14)$$

and the mode changes are defined by $\phi$

$$\phi: \begin{cases} \sigma_{A\to on} \to \alpha_{10} & \text{in } \alpha_{00} \\ \sigma_{B\to off} \to \alpha_{00} & \text{in } \alpha_{01} \\ \sigma_{A\to off} \to \alpha_{00} & \text{in } \alpha_{10}. \end{cases} \quad (15)$$

The instantaneous change in flow to 0 when valve $B$ closes represents a reduction in the size of the continuous state vector. The function $g$ can be specified in a vector equation, $x^+ = g \cdot x$, as

$$g: \begin{cases} [0\ \ 1] & \text{in } \alpha_{00} \\ [1\ \ 1] & \text{in } \alpha_{01} \\ [1\ \ 1] & \text{in } \alpha_{10}. \end{cases} \quad (16)$$

The function $h$ expresses the signal values, used in $\gamma$,

$$s = [p_B\, f_A]^T \quad (17)$$

in terms of the state variables $[x_1\ \ x_2]^T$. When both valves are closed, the pressure $p_B$ is determined by a derivative relation, $p_B = F_{in}R_p - \mathrm{d}x_1/\mathrm{d}t$. For a discontinuous change in $x_1$, $\mathrm{d}x_1/\mathrm{d}t \to \infty$. This situation is approximated by a Dirac pulse, $\delta$. Let the function *sign* be defined as

$$sign(x) = \begin{cases} -1 & \text{if } x < 0 \\ 0 & \text{if } x = 0 \\ 1 & \text{if } x > 0. \end{cases} \quad (18)$$

Then, a discontinuous change results in $p_B = F_{in}R_p - \mathrm{d}x_1/\mathrm{d}t = F_{in}R_p - sign(x_1^+ - x_1)\delta$, which yields

$$h: \begin{cases} p_B = F_{in}R_p - sign(x_1^+ - x_1)\delta, & f_A = 0 & \text{in } \alpha_{00} \\ p_B = \dfrac{x_2^+}{C_{EV}}, & f_A = 0 & \text{in } \alpha_{01} \\ p_B = R_a \dfrac{x_1^+}{I_{IHX}}, & f_A = \dfrac{x_1^+}{I_{IHX}} & \text{in } \alpha_{10}. \end{cases} \quad (19)$$

and the specification is complete. Note that because the discontinuity in behavior is the result of parameter abstraction, the switching condition has to be specified in terms of *a posteriori* conditions embodied in $x^+$ (Mosterman and Biswas, 1997b).

### 4.3. Verification of the cooling system behavior

The control law specified in Section 4.2, when applied to the hybrid model, indicates that the flow momentum in $I_{IHX}$ drops towards 0 when valve $B$ is closed because $f_A = 0$. The high pressure triggers the opening of valve $A$ and immediately after opening, if invariance of state is not observed, there would be no continued flow. Now the alarm valve, $A$, can be closed as well. This would indicate that the control operates as desired, which, in fact, is not true. In the hybrid model, small time constants due to pipe capacity and the time associated with opening and closing of the valves are considered negligible, and, therefore, abstracted away. If all these phenomena were included in the model, there would not be an instantaneous build-up of infinitely large pressure in the system. Instead, the pressure $p_A$ would begin to increase at a very fast rate. When that pressure exceeded its threshold value, the alarm loop would become active, causing the pressure to decrease, and when it dropped sufficiently, the alarm valve would close. However, this would again cause the pressure to increase, and the alarm valve would open. The implication is that the valve chatters (it opens and closes at a very fast rate), and this causes the system to exhibit a sliding-mode behavior in phase space (Mosterman et al., 1997a). Certain systems are designed to operate in sliding regimes, but sliding-mode behavior is not a desirable feature in this application. Therefore, the control-law specification needs to be modified.

To verify consistency in terms of divergence of time, the closed-loop switching specifications in $\gamma$ for which further mode changes occur are detected. Using $\phi$ to establish conditions for further switching, $\gamma$ combined with $h$ shows that this occurs when

$$\begin{cases} F_{in}R_p - sign(x_1^+ - x_1)\delta > p_{th} & \text{if } \alpha_{00} \\ R_a \dfrac{x_1^+}{I_{IHX}} \le p_{th} & \text{in } \alpha_{10}. \end{cases} \quad (20)$$

To verify that no immediate transition back to a previous mode occurs, these conditions have to be expressed in terms of the switching invariant, i.e., the state variables before switching $[x_1\ x_2]^T$. Applying $[x_1^+\ x_2^+]^T = g \cdot [x_1\ x_2]^T$, yields

$$\begin{cases} F_{in}R_p - sign(-x_1)\delta > p_{th} & \text{in } \alpha_{00} \\ R_a\dfrac{x_1}{I_{IHX}} \leq p_{th} & \text{in } \alpha_{10}. \end{cases} \tag{21}$$

Therefore, closed-loop switching events are generated when

$$F_{in}R_p - sign(-x_1)\delta > p_{th} \tag{22}$$

and

$$x_1 \leq \frac{I_{IHX}}{R_a} p_{th}. \tag{23}$$

Since $\delta \to \infty$, Eq. 22 simplifies to $x_1 > 0$.[5] The two conditions plotted in Fig. 9 show the area $0 < x_1 \leq (I_{IHX}/R_a)p_{th}$, where the system can switch between modes $\alpha_{00}$ *and* $\alpha_{10}$ indefinitely. The implication is that for flow momentum values in the above range, the continuous mode cannot be determined. In reality, small time constants that arise from pipe capacitance imply quick changes between the two modes, i.e., chattering. To avoid this behavior in the cooling system the control law needs to be modified.

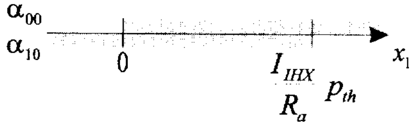

Fig. 9. An area of inconsistency in phase space.

The interaction between the discrete and continuous domains shows that a control algorithm that uses the pressure values to switch modes is insufficient. To establish consistent control, the flow momentum that causes the build-up of pressure needs to be considered as well. If this momentum falls below a safe threshold value, $f_{th}$, build-up of pressure does not exceed the critical value and the alarm valve can be closed safely. To explicate these constraints, $x_1 \leq I_{IHX}f_{th}$ is added to the precondition for $\sigma_{A \to off}$ and $x_1 > I_{IHX}f_{th}$ to $\sigma_{A \to on}$. This results in a unique operational mode for the complete hybrid system if $F_{in} < p_{th}/R_p$. Note that the added condition is of an *energy* nature, since it is based on the energy in the system. If the energy value is less than a critical value, the power from $p_{th} \cdot f_{th}$ is insufficient to keep valve $A$ open, and, therefore, it closes. This forms a guideline for selecting the numerical value of $f_{th}$ in the more detailed design stages.

## 5. Conclusion

This paper presents a hybrid modeling paradigm with formal execution semantics for the analysis of embedded control systems. The resultant modeling framework is based on physical principles resulting in behavior generation that closely matches true system behaviors (Mosterman, 1997). Other approaches to the specification and analysis of hybrid systems (e.g., Alur et al. (1994), Guckenheimer and Johnson (1995), Nicollin et al. (1991)) focus more on formalisms that require the specification of a global mathematical model of physical system behavior, which is hard to define, especially for larger systems. Some of these approaches rely on a discretization of the continuous phase space so that variable dynamics have to be defined by constant rates of change. In contrast, this paper focuses on a systematic compositional modeling paradigm, where system models are constructed from local component/concept specifications (Borst, 1997; Breunese, 1996).

The principle of conservation of state governs the state transition function between modes of continuous behavior, and the principle of invariance of state ensures the correct initial state in a new mode when the system transitions through one or more mythical modes. The invariance of state principle is general, and applies to any state vector definition for a linear hybrid system. The principle of temporal evolution of state ensures that the system behavior does not violate principles of causality. Invariance of state forms the core for verifying divergence of time in system behavior, a necessary condition for establishing the correctness of the system model. The verification procedure is independent of the continuous field, $f$, in any particular mode.

This paper has specifically focused on the application of hybrid modeling techniques to embedded control systems that are typically quite complex. Therefore, compositional modeling and global behavior generation from localized discrete effects provides a distinct advantage over schemes that require more global mathematical models for analysis. From a control viewpoint, this paper establishes that localized discrete effects originate from physical abstractions imposed on system models to simplify their behavior specifications, closed-loop control, and open-loop control. Using this modeling paradigm to support the design of the corresponding control laws to ensure the desired hybrid system behavior, makes it essential that model behaviors do not violate physical principles. From a design viewpoint, applying the systematic principles associated with the modeling paradigm helps identify and eradicate fallacies in the control regimen in the design phase. This was adequately demonstrated in the modeling and analysis of the discrete/continuous interactions of the alarm control specifications

[5] For the boundary condition, $x_1 = 0$, $F_{in} > p_{th}/R_p$ causes inconsistency because $\delta = 0$.

for the reactor cooling system. Future work will focus on a more-detailed study and analysis of sliding-mode behaviors, and the application of this modeling paradigm to monitoring and fault isolation in dynamic, embedded systems.

## References

Aho, Alfred John, V., Hopcroft, E., Jeffrey D. Ullman, 1974. The Design and Analysis of Computer Algorithms. Addison-Wesley Publishing Company. Reading, Massachusetts. ISBN 0-201-00029-6.

Alur, R., Courcoubetis, C., Halbwachs, N., Henzinger, T.A., Ho, P.-H., Nicollin, X., Olivero, A., Sifakis, J., Yovine, S., 1994. The algorithmic analysis of hybrid systems. In: Bakkers, J.W., Huizing, C., de Roeres, W.P., Rozenberg, G., (Eds.), Proceedings of the 11th International Conference on Analysis and Optimization of Discrete Event Systems. Springer-Verlag, pp. 331–351. Lecture Notes in Control and Information Sciences 199.

Borst, Pim. 1997. Construction of Engineering Ontologies for Knowledge Sharing and Reuse. PhD dissertation. University of Twente. The Netherlands. ISBN 90-365-0988-2.

Breunese, A.P.J., 1996. Automated Support in Mechatronic Systems Modeling. PhD dissertation. University of Twente. The Netherlands.

Broenink, Jan F., Paul B.T. Weustink, 1996. A combined system simulator for mechatronic systems. In: Proceedings of ESM 96. Budapest, Hungary.

Davies, Jim., Steve Schneider, 1989. An introduction to Timed CSP. Oxford University Computing Laboratory. Oxford, UK.

Garcia, H.E., Ray, A., Edwards, R.M., 1995. A reconfigurable hybrid system and its application to power plant control. IEEE Transactions on Control Systems Technology.

Guckenheimer, John., Stewart Johnson, 1995. Planar hybrid systems. In: Panos Antsaklis, Wolf Kohn, Anil Nerode, Shankar Sastry, (Eds.), Hybrid Systems II. Vol. 999. Springer-Verlag, pp. 202–225. Lecture Notes in Computer Science.

Harel, David. 1987. Statecharts: A visual formalism for complex systems. Science of Computer Programming 8, 231–274.

Hatley, Derek, J., Imtiaz Pirbhai, 1988. Strategies for Real-Time Systems Specification. Dorset House Publishing Co., New York, New York.

Henzinger, Thomas, A., Xavier Nicollin, Joseph Sifakis, Sergio Yovine, 1994. Symbolic model checking for real-time systems. Information and Computation 111, 193–244.

Hoare, C.A.R., 1978. Communicating sequential processes. Communications of the ACM 21(8), 666–677.

Karnopp, D.C., Margolis, D.L., Rosenberg, R.C., 1990. Systems Dynamics: A Unified Approach. 2 ed. John Wiley and Sons. New York.

Kassakian, John, G., Martin F. Schlecht, George C. Verghese, 1991. Principles of Power Electronics. Addison-Wesley Publishing Company. Reading, Massachusetts. ISBN 0-201-09689-7.

Kohavi, Zvi. 1978. Switching and Finite Automata Theory. McGraw-Hill, Inc.. New York.

Kowalewski, Stefan., Jorg Preußig, 1996. Verification of sequential controllers with timing functions for chemical processes. In: 13th IFAC World Congress. San Francisco, CA.

Lennartson, Bengt., Michael Tittus., Bo Egardt., Stefan Pettersson, 1996. Hybrid systems in process control. IEEE Control Systems, pp. 45–56.

Lygeros, John., Datta Godbole., Shankar Sastry, 1994. Simulation as a tool for hybrid system design. In: 1994 AIS Conference on Distributed Interactive Simulation Environments.

Mosterman, Pieter J., 1997. Hybrid Dynamic Systems: A hybrid bond graph modeling paradigm and its application in diagnosis PhD dissertation. Vanderbilt University.

Mosterman, Pieter J., Gautam Biswas, 1995. Modeling Discontinuous Behavior with Hybrid Bond Graphs In: Qualitative Reasoning Workshop. University of Amsterdam, Amsterdam, pp. 139–147.

Mosterman, Pieter J., Gautam Biswas, 1996. A Formal Hybrid Modeling Scheme for Handling Discontinuities in Physical System Models In: AAAI-96. AAAI Press, 445 Burgess Drive, Menlo Park, CA, 94025. Portland, Oregon, pp. 985–990.

Mosterman, Pieter J., Gautam Biswas, 1997a. Formal Specifications from Hybrid Bond Graph Models In: Qualitative Reasoning Workshop. Cortona, Italy, pp. 131–142.

Mosterman, Pieter J., Gautam Biswas, 1997b. Principles for Modeling, Verification, and Simulation of Hybrid Dynamic Systems In: Fifth International Conference on Hybrid Systems. Notre Dame, Indiana.

Mosterman, Pieter J., Gautam Biswas, 1998. A theory of discontinuities in physical system models. Journal of the Franklin Institute, 335B, 409–439.

Mosterman, Pieter J., Feng Zhao, Gautam Biswas, 1997a. Model semantics and simulation for hybrid systems operating in sliding regimes. In: AAAI Fall Symposium on Model Directed Autonomous Systems.

Mosterman, Pieter J., Gautam Biswas, Janos Sztipanovits, 1997b. Hybrid modeling and verification of embedded control systems. In: Proceedings of the 7th IFAC CACSD '97 Symposium. Gent, Belgium, pp. 21–26.

Murata, Tadao. 1989. Petri nets: Properties, analysis and applications. Proceedings of the IEEE 77(4), 541–580.

Nicollin, Xavier. Joseph Sifakis. Sergio Yovine. 1991. From atp to timed graphs and hybrid systems. In: Bakkers, J.W.. Huizing. C., de Roeres, W.P., Rozenberg, G., (Eds.) Lecture Notes in Computer Science Vol. 600. Mook, The Netherlands, pp. 549–571. Real Time: Theory and Practice.

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

Otter, Martin. 1994. Objektorientierte Modellierung mechatronischer Systeme am Beispiel geregelter Roboter. PhD dissertation. Ruhr-Universitat Bochum.

Paynter, Henry M., 1961. Analysis and Design of Engineering Systems. The M.I.T. Press. Cambridge, Massachusetts.

Rudin, W., 1976. Principles of Mathematical Analysis. 3 ed. McGraw-Hill. New York.

Sweet, William. 1995. The glass cockpit. IEEE Spectrum, pp. 30–38.

Ward Paul T., Stephen J. Mellor, 1985. Structured Development for Real-Time Systems. Prentice-Hall. Englewood Cliffs, New Jersey.

Wijbrans, K.C.J., 1993. Twente Hierarchical Embedded Systems Implementation by Simulation: a structured method for controller realization. PhD dissertation. University of Twente. CIP-DATA Koninklijke Bibliotheek, Den Haag, The Netherlands. ISBN 90-9005933-4.