# HYBRID MODELING AND VERIFICATION OF EMBEDDED CONTROL SYSTEMS*

**Pieter J. Mosterman**[*,*], **Gautam Biswas**[**] **and Janos Sztipanovits**[***]

*This work is partially supported by grants from PNC, Japan and Hewlett-Packard Co., USA.*

**Center for Intelligent Systems, Box 1679, Sta B, Vanderbilt University, Nashville, TN, 37235,* pjm,biswas@vuse.vanderbilt.edu

***Measurement and Computing Systems Laboratory, Box 6306, Sta B, Vanderbilt University, Nashville, TN, 37235,* sztipaj@vuse.vanderbilt.edu

**Abstract.** Contemporary process control includes continuous time PID controllers for actuation, and at a higher level, supervisory control mechanisms to select appropriate control algorithms for the different modes of system operation in an attempt to achieve optimal or near-optimal control. To model and analyze such control phenomena that include discrete and continuous components requires the use of *hybrid* modeling techniques. This paper presents a hybrid modeling paradigm that identifies and captures the phenomena specific to hybrid system models, and specifies its execution semantics based on the principles of *invariance of state* and *temporal evolution of state*. The modeling methodology is applied to analysis of control behavior in dynamic physical systems, and model verification based on *divergence of time* demonstrates possible applications in design tasks.

**Key Words.** command and control systems; formal methods; hybrid systems; verification

## 1. INTRODUCTION

Complexity of large-scale engineering systems has increased the significance and the need for models that can accurately simulate and verify system behavior. These 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). These embedded control systems, i.e., physical systems that are controlled by digital computers, interact between the signal and power domain where digital control mechanisms coexist with low-level continuous PID control (Garcia *et al.*, 1995). Therefore, complete process control modeling schemes are required to capture both discrete control *and* continuous process characteristics, and are referred to as *hybrid systems*.

Consider the cooling system in Fig. 1. The sodium pump, used to maintain a sufficient flow of coolant, is driven by a synchronous *ac* motor. The flow rate depends on the motor revolutions per minute, which is determined by the frequency of the *ac* signal. To achieve sufficient torque for this flow rate, a continuously operating PID controller controls the power supplied. Actuated valves throughout the loop guard against catastrophic failures. Physical systems are inherently continuous, but the time scale of the opening and closing of a valve is sufficiently small to be modeled as discrete on/off states.

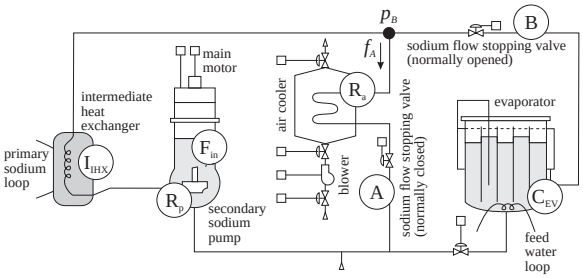


Fig. 1. **Continuous and discrete process control.**

Hybrid modeling techniques can form the basis for a comprehensive study of system performance of embedded control systems, that include the effects of implementation choices such as interfaces, sampling rates, and computation order in software. Previous work developed a modeling theory of dynamic physical systems that is made up of: (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 transferring the system state vector from one operational mode to another through a sequence of transitions (Mosterman and Biswas, 1997*a*; Mosterman and Biswas, 1997*b*).

This paper develops the formal hybrid scheme for modeling control processes and specifies its execution semantics based on the principles of *con-*

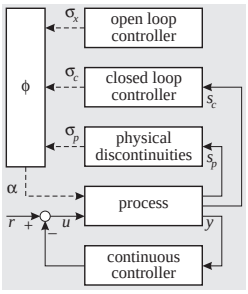


Fig. 2. **A physical process under hybrid control.**

*servation of state*, *invariance of state*, and *temporal evolution of state*. The modeling paradigm applies the principle of *divergence of time* to verify the correctness of an embedded control system model or design.

## 2. HYBRID MODELING FORMALISM

Embedded control of physical processes is assumed to have a continuous physical sub-system, e.g., a chemical plant or a nuclear reactor, and the modeling paradigm must include a hybrid specification.

### 2.1. *Architecture*

Fig. 2 shows the general hybrid architecture of a controlled physical process. The process and its continuous controller embody the continuous characteristics of the system. Configuration changes in the system can be attributed to three phenomena: (1) when physical system signals cross threshold values; this can be mainly attributed to abstractions incorporated in the physical system and continuous controller model, (2) explicit signals that activate the closed loop controller to make changes, and (3) external, open loop control. The events generated by these phenomena are called, $\sigma_p$, $\sigma_c$, and $\sigma_x$, respectively.

Note that the input signal $u$ is required to be continuous. Discontinuous changes in the input (e.g., step input) and changes in the low-level continuous controller are modeled by the open loop controller deactivating one input signal and at the same time activating the newly desired input.

### 2.2. *The Continuous Model*

Continuous physical system behaviors embody energic interaction and are typically described by a state space representation with ordinary differential equations (ODEs). Thus, the continuous system model consists of:

- $\dot{x}(t) = f_\alpha(x(t), u(t), t), t \in \Re, \alpha \in \aleph$, defines the continuous behavior of the system in operational mode $\alpha$. There is one and only one field, $f_{\alpha_i}$, for each mode of continuous operation $\alpha_i$.
- $X \in \Re^m$, the continuous state vector.
- $U \in \Re^p$, the vector of input signals.

### 2.3. *The Discrete Model*

Discrete events in embedded control systems stem from (Lennartson *et al.*, 1996):

- discontinuous input produced by idealized discrete actuators,
- discontinuous control which switches operational modes,
- modeling artifacts, where nonlinear behaviors of a system may be abstracted or approximated as piecewise linear behavior, and
- discontinuous output which is the result of measurements made on discrete sensors.

These discrete changes are modeled by a transition function, $\phi$, and transitions are invoked by events in a set $\Sigma$. To systematically derive $\phi$, it may consist of a number of independent state machines that control local switching effects. An operational mode is determined by the combination of individual states of the independent state machines. Some of these modes may not have a physical representation nor meaning, but during behavior generation have to be traversed to a arrive at a new real mode. A main contribution of this paper is to establish execution semantics that handle these sequences of mode changes correctly. The discrete modeling paradigm can be implemented by Petri-nets or finite state automata and is represented as:

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

### 2.4. *Interaction*

Interaction between the continuous and discrete part is specified by (1) discrete events generated by the continuous model, and (2) a change of operational mode by the discrete model:

- $S \in \Re^n$, the signals used for event generation.
- $\gamma : S \rightarrow \Sigma_s$, where $\Sigma_s = \Sigma_p \times \Sigma_c$, generates discrete events from the signal values.
- $h : X \times U \times I \rightarrow S$, returns new signals from the state change.

- $g : X \times I \rightarrow X$, transfers the continuous state vector to the new operational mode, $\alpha \in I$, which may result in discontinuous changes.

The function $\gamma$ generates discrete events when signals, $s \in S$, cross pre-defined threshold values. The output function, $h$, computes the values of these signals from the continuous state vector in an operational mode. The generated events applied to the model may indicate that the system changes its mode of continuous operation. Associated with every mode, $\alpha$, is a field, $f_\alpha$, specifying continuous behavior evolution. When mode switching occurs, the continuous state vector of the system may change. The function $g$ maps the state vector of the continuous model between operational modes and in general may be difficult to derive. In physical systems, this function has to satisfy the principle of conservation of state (charge, momentum, etc.), which allows for an algorithmic mapping of states between operational modes (Mosterman and Biswas, 1997*a*). A natural choice for defining state variables are signal values associated with the energy storage elements in a physical system (e.g., see bond graph modeling techniques (Karnopp *et al.*, 1990)).
In summary, the complete hybrid system model is defined by the 9-tuple (Lennartson *et al.*, 1996)

$$H =< I, \Sigma, \phi, X, U, f_\alpha, g, h, \gamma > . \qquad (1)$$

### 2.5. *Execution Semantics*

Fig. 3 shows a block diagram of the operation of a hybrid system. During continuous evolution, system behavior is determined by the active state description, $f_{\alpha_k}$, in mode $\alpha_k$. At a well-defined point in time, $t_s$, the continuous state vector in mode $\alpha_k$, $x_{\alpha_k}$, has corresponding signal values that may invoke an event, $\sigma$. At this point, real time $t$ is suspended, and the transition function $\phi$ is activated to derive a new mode, $\alpha_{k_1}$, (Fig. 4) that now governs the continuous system behavior evolution. The $g$ function derives the new state vector $x^+$ from $x_{\alpha_k}$. The signal values corresponding to $x^+$ may generate a new event, causing a consecutive mode transition to $\alpha_{k_2}$. The corresponding state vector for this mode, $x^+$, is again determined from $x_{\alpha_k}$. Further mode changes may occur and this continues until an operational mode, $\alpha_m$, with state vector, $x^+$, is arrived at that does not generate any other immediate transitions. Because the new state vector, $x^+$, is derived from the original state, $x_{\alpha_k}$, it does not depend on the path of intermediate model configurations.
After the new operational 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 operational mode changes (Fig. 4) which terminates in mode $\alpha_n$. This sequence transitions

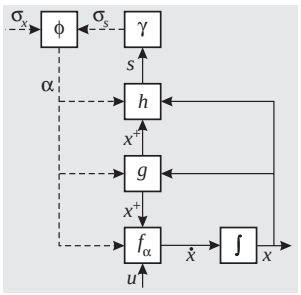


Fig. 3. **A general hybrid system.**

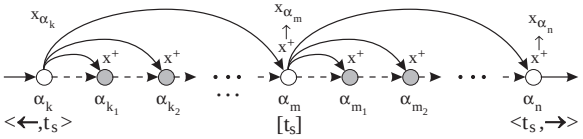


Fig. 4. **System state is derived from the original state vector.**

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 operational modes ($\alpha_k$, $\alpha_m$, and $\alpha_n$) follow each other immediately. Intermediate modes that were traversed during sequences of discontinuous changes have no representation on the real time-line, and, therefore, are termed *mythical*.

## 3. HYBRID SYSTEM VERIFICATION

A key aspect of hybrid system modeling is to define interaction between the continuous and discrete components in an integrated manner, without violating overall physical system principles. It is shown how semantics based on *invariance of state* and *temporal evolution of state* along with *divergence of time* maintain consistency of the interactions.

### 3.1. *Invariance of State*

Previous work shows that a special continuous state vector of a physical system model, $p_{\alpha_0}$, that represents the stored energy in physical buffer elements, e.g., springs and inertias, is invariant across consecutive changes in operational mode (Mosterman and Biswas, 1997*b*).

*Conjecture 3.1* The special state vector, $p_{\alpha_0}$, is invariant across mode changes.

*Lemma 3.1* (Invariance of State) Any vector that represents the state of a linear physical system is invariant across mode changes.

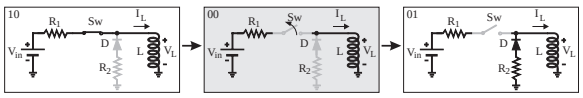


Fig. 5. **A mythical mode in a physical system.**

*Proof:* Let $x_{\alpha_0}$ represent a possible state vector in operational mode $\alpha_0$. Then, for a linear system, there is an algebraic translation $T_{\alpha_0}$ unique to a given operational mode that defines the relation between $x_{\alpha_0}$ and $p_{\alpha_0}$ defined above, i.e., $x_{\alpha_0} = T_{\alpha_0}(p_{\alpha_0})$. Since $p_{\alpha_0}$ is invariant across mode changes, so is $T^{-1}_{\alpha_0}(x_{\alpha_0})$. If $x_{\alpha_n}$ is a vector capturing the system state after the mode changes end, then $x_{\alpha_n} = T_{\alpha_n}(p_{\alpha_n})$ and, since $p_{\alpha_0}$ is invariant across transitions, $x_{\alpha_n}$ can be expressed as, $x_{\alpha_n} = T_{\alpha_n}(g(p_{\alpha_0}))$. So, $x_{\alpha_n} = T_{\alpha_n}(g(T^{-1}_{\alpha_0}(x_{\alpha_0})))$ also is invariant because the function $g$ and the mappings $T_{\alpha_n}$ and $T^{-1}_{\alpha_0}$ are defined by the specific operational mode and no function of how it was achieved, i.e., they are path invariant. ■

To illustrate, consider the simple diode-inductor circuit in Fig. 5 that resembles the operation of the secondary sodium cooling system. Initially, the switch is closed, the inductor builds up a flux and the diode is inactive (mode $\alpha_{10}$). In steady state the voltage across the inductor is 0 and it draws a constant current. If the switch is opened, the system changes configuration to $\alpha_{00}$, the current through the inductor is forced to 0 instantaneously and a large, negative, voltage drop over the inductor is induced to release its energy. When the voltage drop across the diode becomes larger than its threshold voltage, say, $0.6V$, the diode comes on, the system switches to $\alpha_{01}$, and the inductor draws its current through the diode path. To generate correct behavior, the eventual current drawn through the diode is determined based on the current drawn by the inductor at the moment the switch was opened. If it were based on the intermediate, mythical, configuration $\alpha_{00}$ where the current through the inductor was forced to 0, the continuous state vector in the final configuration would show a 0 current through the diode, which is in conflict with real behavior.
The continuous state vector of the diode-inductor circuit can be chosen as either the inductor current, $I_L$, or the inductor voltage, $V_L$. The algebraic translation between the states depends on the operational mode

$$T : \begin{cases} V_L = 0 & \text{if } \alpha_{00} \\ V_L = I_L R_L + V_D & \text{if } \alpha_{01} \\ V_L = -I_L R_L + V_{in} & \text{if } \alpha_{10} \end{cases} \quad (2)$$

If the inductor current is chosen to represent system state it is associated with the same energy storage element in each mode, and, therefore, provides a consistent mapping. Previous work shows that in this case 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 the intermediate configurations (Mosterman and Biswas, 1996*a*).[1] If the state vector is defined in terms of $V_L$, and its value before switching begins is $V_{L,0}$, then, using invariance of state of the special state variable, $I_L$, $I_L^+ = g(I_{L,0}) = I_{L,0}$ and $T_{\alpha_{01}} : V_L^+ = I_L^+ R_L + V_D$, $V_L^+ = I_{L,0} R_L + V_D$. If $I_{L,0}$ is expressed in terms of $V_{L,0}$ by using $T^{-1}_{\alpha_{10}} : I_{L,0} = -\frac{1}{R_1} V_{L,0} + \frac{1}{R_1} V_{in}$ the new continuous system state can be expressed in its value before switching by $T_{\alpha_{01}}(T^{-1}_{\alpha_{10}})$,

$$V_L^+ = \frac{R_2}{R_1} V_{L,0} + \frac{R_2}{R_1} V_{in} + V_D \quad (3)$$

This illustrates that, in general, the new value of any continuous state vector is independent of the intermediate operational modes that are mythical. It is completely determined by the original and new modes of continuous operation only.

### 3.2. *Temporal Evolution of State*

Configuration changes can cause dependencies among state variables and exogeneous variables and change the number of degrees of freedom of the system. This produces discontinuous changes in values at points in time. Continuity of power requires well-defined functions on the left and right intervals about the point of discontinuity (Fig. 4), therefore, configuration changes cause piecewise continuous behaviors with a countable number of *simple* discontinuities which have a limit value. We show that the state vector at the point of discontinuity is the limit value of the state in the *new* operational mode.

*Conjecture 3.2* A hybrid system is piecewise continuous.

*Lemma 3.2* (Temporal Evolution of State)
Continuous state variable values have to be continuous in left-closed intervals.

*Proof:* For simple discontinuities, limit values exist at a time of switching, $t_s$, $x_{\alpha_k}(t_s^-) = \lim_{t \uparrow t_s} x_{\alpha_k}(t)$, and $x_{\alpha_n}(t_s^+) = \lim_{t \downarrow t_s} x_{\alpha_n}(t)$ (Fig. 4). In case of a jump, $x_{\alpha_k}(t_s^-) \neq x_{\alpha_n}(t_s^+)$. Because the state vector exists for all points on the real time-line, there is a state vector $x_{\alpha_m}(t_s)$ determined at $t_s$. If the state vector at $t_s$, $x_{\alpha_m}(t_s) \neq x_{\alpha_n}(t_s^+)$ then the system continues to evolve in a left open interval, $< t_s, \rightarrow >$, after configuration changes have occurred, starting with $\lim_{t \downarrow t_s} x_{\alpha_n}(t)$. However, causality requires that the initial state in the new configuration be a function solely of $x_{\alpha_m}(t_s)$ and the new configuration, and, therefore, the state vector has to evolve in left-closed intervals, $[t_s, \rightarrow >$. ■

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

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.

### 3.3. *Divergence of Time*

When a sequence of mode changes invokes a previously traversed mode, a loop emerges. Because discontinuous changes happen at instants in time, a loop of discontinuous changes invoked by the discrete model part would result in system behavior no longer progressing 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 systems (Henzinger *et al.*, 1994; Mosterman and Biswas, 1997*b*).

To ensure discontinuous changes occur instantaneously and at that time diverges, it has to be proven that transitions in the discrete model always terminate in a new operational mode where a field, $f_\alpha$, again governs continuous evolution of system behavior. This analysis is complicated because continuous signals that generate discrete events by crossing threshold values may change discontinuously themselves between operational modes. An algorithm that tracks the signal values that change discontinuously between operational modes and the corresponding events this may generate increases in computational complexity rather quickly.

This problem is best addressed by invoking the principle of invariance of state. Signal values may change discontinuously between operational modes, but their values in the newly arrived mode are always determined by the state vector of the last continuous operational mode. Since this state vector is not affected by future configuration changes, it is invariant and 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 based on the newly found state vector, 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 operational 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 original state.

## 4. THE SECONDARY COOLING SYSTEM

Fig. 1 shows a schematic representation of a cooling system. Pump losses are represented by the dissipation parameter, $R_p$. The cooling fluid is pumped through a coil in an intermediate heat-exchanger which has an inertia value, $I_{IHX}$, responsible for building up flow momentum. An evaporator vessel with capacitance $C_{EV}$ transports heat from the heat-exchanger to a steam water loop that 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 fluid completely.

### 4.1. *Specifying the System*

The continuous variables defining a state vector in this system are the flow momentum, $x_1$, in the coil of the intermediate heat exchanger, $I_{IHX}$ and the stored fluid, $x_2$, in the evaporator, $C_{EV}$, $x = [x_1\ x_2]^T$. The input to the system is the input flow, $F_{in}$, $u = F_{in}$. The discrete model is determined by the two valves in the system which results in four operational modes, $\alpha_{00} = \{A_{closed}, B_{closed}\}$, $\alpha_{01} = \{A_{closed}, B_{open}\}$, etc., so $Q = \{\alpha_{00}, \alpha_{01}, \alpha_{10}, \alpha_{11}\}$. The initial mode is $\alpha_{01}$. 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 $\sigma_{B\rightarrow off}$, the fluid flow becomes 0 abruptly and a large pressure is induced. To prevent this pressure from becoming too high and causing damage to the piping, the closed loop control makes a release path available by generating $\sigma_{A\rightarrow on}$ when $p_B > p_{th}$, which opens valve $A$. Over time, the pressure falls below $p_{th}$, and the controller closes the valve $A$, $\sigma_{A\rightarrow off}$. This results in the complete event set $\Sigma = \{\sigma_{A\rightarrow on}, \sigma_{A\rightarrow off}, \sigma_{B\rightarrow off}\}$. The closed loop controller generates events $\Sigma_c$, which are specified by $\gamma$

$$\gamma : \begin{cases} p_B > p_{th} \rightarrow \sigma_{A\rightarrow on} \\ p_B \leq p_{th} \rightarrow \sigma_{A\rightarrow off} \end{cases} \tag{4}$$

and the mode changes are executed by $\phi$

$$\phi : \begin{cases} \sigma_{A\rightarrow on} \rightarrow \alpha_{10} & \text{if } \alpha_{00} \\ \sigma_{B\rightarrow off} \rightarrow \alpha_{00} & \text{if } \alpha_{01} \\ \sigma_{A\rightarrow off} \rightarrow \alpha_{00} & \text{if } \alpha_{10} \end{cases} \tag{5}$$

The instantaneous change in flow to 0 when valve $B$ closes represents a reduction in the size of the continuous state vector, captured by $g$ in $x^+ = g \cdot x$,

$$g : \begin{cases} [0\ 1] & \text{if } \alpha_{00} \\ [1\ 1] & \text{if } \alpha_{01} \\ [1\ 1] & \text{if } \alpha_{10} \end{cases} \tag{6}$$

The function $h$ translates the state variables $[x_1\ x_2]^T$ into signal values $s = [p_B\ f_A]^T$ that are used by $\gamma$. Note that when both valves are closed, the pressure $p_B$ is determined by a derivative relation, $p_B = F_{in}R_p - I_{IHX}\frac{dx_1}{dt}$, which is approximated by a Dirac pulse, $\delta$, for discontinuous changes in $x_1$. Let the function *sign* be defined as

$$sign(x) = \begin{cases} -1 & \text{if } x < 0 \\ 0 & \text{if } x = 0 \\ 1 & \text{if } x > 0 \end{cases} \quad (7)$$

Then, a discontinuous change results in $p_B = F_{in}R_p - I_{IHX}\frac{dx_1}{dt} = F_{in}R_p - sign(x_1^+ - x_1)\delta$, which yields

$$h : \begin{cases} p_B = sign(x_1^+ - x_1)\delta,\ f_A = 0 & \text{if } \alpha_{00} \\ p_B = \frac{x_2^+}{C_{EV}},\ f_A = 0 & \text{if } \alpha_{01} \\ p_B = R_a\frac{x_1^+}{I_{IHX}},\ f_A = \frac{x_1^+}{I_{IHX}} & \text{if } \alpha_{10} \end{cases} \quad (8)$$

and the specification is complete.

### 4.2. *Verification of the Cooling System*

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

$$\begin{cases} F_{in}R_p - sign(x_1^+ - x_1)\delta > p_{th} & \text{if } \alpha_{00} \\ R_a\frac{x_1^+}{I_{IHX}} \le p_{th} & \text{if } \alpha_{10} \end{cases} \quad (9)$$

To verify consistency, these conditions have to be expressed in terms of the switching invariant, i.e., the state variables before switching $[x_1\ x_2]^T$, $[x_1^+\ x_2^+]^T = g \cdot [x_1\ x_2]^T$, yields

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

Therefore, closed loop switching events are generated when $F_{in}R_p - sign(-x_1)\delta > p_{th}$ and

$$x_1 \le \frac{I_{IHX}}{R_a}p_{th}. \quad (11)$$

Considering that $\delta$ approaches infinity, there is an area $0 < x_1 \le \frac{I_{IHX}}{R_a}p_{th}$ where the system switches between modes $\alpha_{00}$ and $\alpha_{10}$ indefinitely. For this flow momentum, the system is not consistent as it is not determined which of the operational mode is reached.[2] Note that, if $x_1 = 0$ then $F_{in}R_p > p_{th}$ causes inconsistency, which is true if $F_{in} \ge \frac{p_{th}}{R_p}$. The interaction between the discrete and continuous domain shows that a control algorithm based on pressures is insufficient, the flow momentum that causes the build-up of pressure needs to be considered as well. If this momentum has fallen 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 specify these constraints, $x_1 \le I_{IHX}f_{th}$ is added to the precondition for $\sigma_{A\to off}$ and $x_1 > I_{IHX}f_{th}$ to $\sigma_{A\to on}$. Now, a unique operational mode is specified for the complete hybrid system if $F_{in} < \frac{p_{th}}{R_p}$. Note that the added condition is of an *energic* nature, since it is based on the flow momentum of the system.

[2] This is also true if the system could be modeled as non-deterministic. A loop of operational modes is different from a unique mode that is not deterministically arrived at.

## 5. CONCLUSIONS

This paper formulates a hybrid modeling paradigm for embedded control systems and presents the corresponding execution semantics. Under the assumption that the continuous characteristics are attributed to physical systems, invariance of state applies to any state vector for linear hybrid systems and can be used to verify divergence of time. The verification procedure does not rely on the form of the continuous field, $f$, in any particular operational mode. The usefulness of modeling discrete/continuous interaction is demonstrated by designing an alarm control specification for a cooling system.

## 6. REFERENCES


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

Henzinger, Thomas A., X. Nicollin, J. Sifakis and S. Yovine (1994). Symbolic model checking for real-time systems. *Information and Computation* **111**, 193–244.

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

Lennartson, B., M. Tittus, B. Egardt and S. Pettersson (1996). Hybrid systems in process control. *IEEE Control Systems* pp. 45–56.

Mosterman, P. J. and G. Biswas (1996*a*). A formal hybrid modeling scheme for handling discontinuities in physical system models. In: *AAAI-96*. Portland, OR. pp. 985–990.

Mosterman, P. J. and G. Biswas (1997*a*). Hybrid modeling specifications for dynamic physical systems. In: *ICBGM'1997*. Phoenix, AZ. pp. 162–167.

Mosterman, P. J. and G. Biswas (1997*b*). A theory of discontinuities in dynamic physical systems. *Journal of the Franklin Institute*. in press.

Sweet, W. (1995). The glass cockpit. *IEEE Spectrum* pp. 30–38.