Hybrid Automata for Modeling Discrete Transitions in Complex Dynamic Systems

Pieter J. Mosterman*
Institute of Robotics and System Dynamics
DLR Oberpfaffenhofen
D-82230 Wessling. Germany
Pieter.J.Mosterman@dlr.de

Gautam Biswas†
Knowledge Systems Laboratory
Stanford University
Stanford, CA 94305. U.S.A.

Abstract

Hybrid system models combine continuous behavior evolution with discrete mode transitions. These transitions may cause discontinuous changes in the field that defines continuous system behavior and the variable values associated with the continuous state vector. In reality, these discontinuous changes are fast continuous transients. To simplify the analysis of these transients *time scale* and *parameter* abstractions are applied to system models with very different impacts on the analysis of system behavior. We have developed a systematic modeling approach based on hybrid automata which combines *a priori* and *a posteriori* switching values to formally implement switching semantics associated with the abstraction events.

Introduction

The pressure to achieve more optimal and reliable performance on complex systems such as aircraft and nuclear plants, while meeting rigorous safety constraints is leading to more detailed analysis of the embedded controllers for these systems. In embedded systems, the continuous physical process interaction with digital control signals requires modeling schemes that facilitate the analysis of mixed continuous and discrete, i.e., *hybrid* behavior. Discrete phenomena may also occur when modeling abstractions are applied to simplify fast nonlinear continuous process behavior.

Consider the primary aerodynamic control surfaces of an airplane in Fig. 1 (Seebeck, 1998). Modern avionics systems employ electronic signals generated by a digital computer, which are transformed into the power domain by electro-hydraulic actuators. The primary flight control system exemplifies the need for hybrid modeling in embedded control systems. At the lowest level in the control hierarchy, positioning of the rudder, elevators, and ailerons are achieved by continuous PID control. Desired set point values are generated directly by the pilot or by a supervising control algorithm implemented on a digital processor. Digital control may mandate *mode* changes at different stages of a flight plan (e.g., *take-off*, *cruise*, and *go-around*). Detection of failed components may lead to discrete changes in system configuration. Model simplification by discretizing fast nonlinear transients also results in discontinuous variable changes.

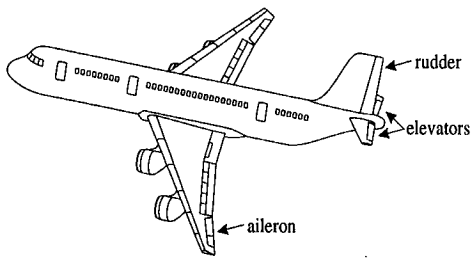

Figure 1: **Aerodynamic control surfaces.**

We have developed a hybrid modeling paradigm that encompasses analysis of embedded systems and modeling abstractions in physical systems. In this paper we present our formalisms for abstracting complex transients into hybrid automata models, and discuss formal semantics for computing the discontinuous changes in the system state vector. The methodology is applied to the elevator positioning subsystem of the primary flight control system of aircraft to demonstrate the correspondence between the model semantics and physical system behavior.

Hybrid Dynamic Systems

Hybrid modeling paradigms (Alur *et al.*, 1993, Guckenheimer and Johnson, 1995, Mosterman *et al.*, 1998b) supplement continuous system description by mechanisms that model discrete state changes resulting in discontinuities in the field description and the continuous state variables. In previous work we have established an ontology of discrete transition types in physical system behavior (Mosterman *at al.*, 1998a).

Differential equations form a common representation of continuous system behavior. The system is described

Supported by a grant from DFG Schwerpunktprogramm KONDISK.

On leave from Dept. of Computer Science, Vanderbilt University, Nashville, TN. Partially supported by a grant from Hewlett Packard Company.

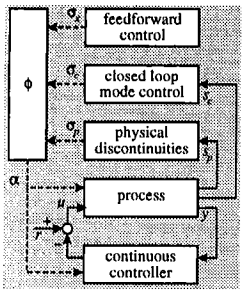

Figure 2: **Hybrid control.**

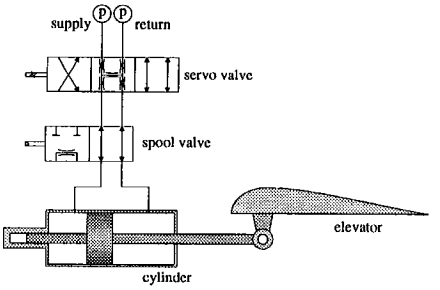

Figure 3: **Hydraulics of one actuator.**

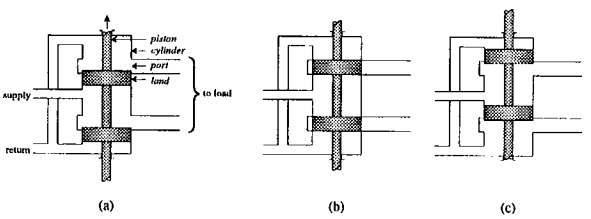

Figure 4: **A typical spool valve.**

by a *state vector*, x, and other variables called *signals*, s, are derived algebraically, $s = h(x)$. Behavior over time is specified by field f. Interaction with the environment is specified by *input* and *output* signals, u and y.

Discrete systems, modeled by a state machine representation, consist of a set of discrete modes, α. Mode changes caused by events, σ, are specified by the *state transition function* ϕ, i.e., $\alpha_{i+1} = \phi(\alpha_i)$. A transition may produce additional discrete events, causing further transitions.

In hybrid dynamic systems, a mode change from α_i to α_{i+1}, may result in a field definition change from f_{α_i} to $f_{\alpha_{i+1}}$. Discontinuous changes in the state vector are governed by an algebraic function g, $x^+ = g_{\alpha_i}^{\alpha_{i+1}}(x)$. Discrete mode changes are caused by an *event generation function* γ associated with the current active mode, α_i, $\gamma_{\alpha_i}(x) \leq 0 \rightarrow \sigma_j$.

The resultant general architecture for hybrid models of embedded control systems appears in Fig. 2 (Mosterman and Biswas, 1997a). Signal value changes (s_p) and closed-loop control active in mode α (s_c) may cause discontinuous changes. The corresponding physical events, σ_p and σ_c, or open loop control generated discrete events σ_x cause mode transitions defined by ϕ.

The Elevator System

Attitude control in an aircraft is achieved by the elevator control subsystem (Seebeck, 1998). This system may consist of two mechanical elevators (Fig. 1) which are positioned by two electro-hydraulic actuators. When a failure occurs, redundancy management switches the actuators and oil supply systems to ensure maximum control. Fig. 3 shows the operation of an actuator. The elevator positioning is controlled by servo valve, which is implemented by a continuous feedback mechanism. When the actuator is *active*, the spool valve is in its *supply* mode and the control signal generated by the servo valve is transferred to the cylinder that positions the elevator. When the actuator is *passive*, the spool valve is in its *loading* mode, and control signals cannot be transferred to the cylinder. Oil flows between the chambers through a loading passageway, otherwise the cylinder would block movement of the elevator, canceling control signals from the redundant *active* actuator.

Consider a scenario where a sudden pressure drop is detected on one of the left elevator actuators. Redundancy control moves the spool valve this actuator from *supply* to *loading* and the spool valve of the other actuator from *loading* to *supply*. This causes transients that are studied in greater detail below.

Modeling the Elevator System

We employ *parameter* and *time scale* abstractions (Mosterman and Biswas, 1997b) to design a simpler but adequate model of the elevator subsystem for control purposes. Parameter abstraction removes small and large parameter values (parasitic dissipative and capacitive elements) from the model. Time scale abstractions collapse the end effect of phenomena associated with very fast time constants to a point in time. We show how these different abstraction types relate back to physical parameters in the real system.

The Spool Valve

A typical spool valve shown in Fig. 4, consists of a piston that moves in a cylinder. A number of cylinder ports connect the supply and return part of the hydraulic system with the load. Fig. 4(a) and (c) show two possible oil flow configurations. When moving from one to the other, the spool valve passes through the configuration where the ports are closed by lands (Fig. 4(b)).

Mode changes in the actuators are facilitated by the spool valve. To enable analysis of behavior during mode changes, four modes of operation are modeled:
(α_0) *loading*: The spool valve operates as a load. Pressure changes generated by the servo valve are blocked. Oil flow between chambers of the elevator positioning cylinder is possible through a loading passageway.
(α_1) *closed*: The spool valve is closed. Pressure changes

generated by the servo valve are blocked. Oil flow between the chambers of the elevator positioning cylinder is not possible. This is a transitional mode between α_0 and α_3.
(α_2) *opening*: The valve is opening. While its lands move past the ports, fluid inertial effects may become active. Depending on the physical construction of the valve, these may have significant effects on transient behavior.
(α_3) *supply*: The spool valve is opened and supplies control power. Pressure changes generated by the servo valve are transferred to the cylinder that positions the elevator. Flow of oil into and out of this cylinder is possible.

Mode changes of the spool valve are controlled by a redundancy management module which monitors a number of critical system variables. In the fault scenario, a sensor reading in actuator1 generates the failure event, σ_f. In response, the redundancy management reconfigures control by generating a sequence of discrete control signals that cause a switch of actuators. The resulting state α_{ij} indicates the state of actuator2, α_i, and actuator1, α_j.
(1) A control event (σ_c) is generated that causes the piston in the *left* spool valve to move from its *supply* to *loading* position at a constant rate of change. Along the trajectory, a number of physical events (σ_p) occur: (i) $\Delta x > -\epsilon \rightarrow \sigma_{close} \Rightarrow \alpha_1$, the overall system mode becomes α_{01} (actuator2 - loading, actuator1 - closed). (ii) $\Delta x > \epsilon \rightarrow \sigma_{open} \Rightarrow \alpha_2$, the overall system mode becomes α_{02} (actuator2 - loading, actuator1 - opening). (iii) $\Delta x > x_{th} \rightarrow \sigma_{load} \Rightarrow \alpha_0$, the overall system mode becomes α_{00} (actuator2 - loading, actuator1 - loading).
(2) A second control event is generated to move the piston in the *right* spool valve to move from its *loading* to *supply* position with a constant rate of change, causing the overall system to switch through modes $\alpha_{00} \rightarrow \alpha_{10} \rightarrow \alpha_{20} \rightarrow \alpha_{30}$.

The values of ϵ and x_{th} are based on physical parameters of the valve, e.g., the shape of ports and lands (Merritt, 1967). For a *critical center* type valve $\epsilon = 0$, and for a *closed center* valve ϵ has a small nonzero value. The value of x_{th} and ϵ determine when inertial effects become active, $\Delta x > \epsilon$ and for how long $\Delta x > x_{th}$.

Model Assumptions

When an actuator moves to its *closed* mode, oil flow into and out of the cylinder that positions the elevator is blocked. This implies that the cylinder piston that controls elevator position cannot move, and, the elevator stops moving as well. In reality, internal dissipation and small elasticity parameters of the oil cause the elevator velocity to change continuously during the transition. The behavior in the continuous transient mode between *supply* and *closed* is shown in Fig. 5. How quickly the system reaches the 0 velocity state in the closed mode depends on the elasticity and internal dissipation parameters chosen for the oil. After a short time in the *closed* mode, the actuator moves to the

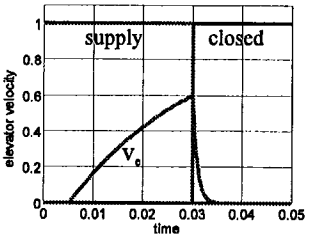

Figure 5: **Continuous transients:** *closed* **mode.**

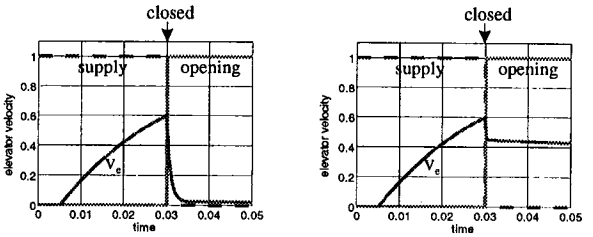

Figure 6: **small Continuous transients:** *supply* $\rightarrow$ *closed* $\rightarrow$ *opening* **for spool valves - (a)** $I = 1$ **and (b)** $I = 100$.

opening mode, and the inertial effects become active. Fig. 6 illustrates the continuous transients involved in the transition. The inertial parameter determines the final elevator velocity, v_e. In the *opening* mode, the inertial effect decreases as the clearance between port and land increases. After some time its value becomes negligible, and the actuator operates as a simple load (*loading* mode). This is shown in Fig. 7 for an inertial parameter with two different values.

The continuous transients described above are not of much interest to the modeler for analysis and control (see Fig. 7 where the transients in the opening mode are still clearly visible but the continuous transients in the closed mode are not). Model simplification results in removal of small elasticity and inertial effects but Fig. 7 illustrates that depending on their magnitude, they may have a distinct impact on the overall system behavior.

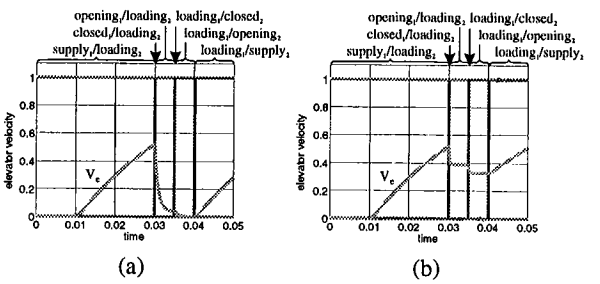

Figure 7: **Continuous transients:** *loading* **mode - (a)** $I = 1$ **and (b)** $I = 100$.

Abstraction Types

We apply previous work on model simplification by abstraction (Mosterman and Biswas, 1997b) to analyze the elevator control subsystem.

Time Scale Abstraction. In the *opening* mode, fluid inertia and dissipative effects in the clearance between land and port, cause a second order build-up of fluid flow. Though the fluid flow velocity and its time derivative are 0 initially, the velocity of the elevator and the driving piston are not. This results in a pressure build-up in the cylinder governed by the elasticity coefficient of the oil which causes a rapid increase of fluid flow through the land/port opening. The pressure also causes the elevator velocity to decrease rapidly resulting in the transient in Fig. 6(a). The initial transient from moving into the *closed* mode is replaced by the transient moving into the *opening* mode. The difference is best seen by comparing Fig. 5 with Fig. 6(a). The final value of the velocity after this transient depends on the dissipative effects and starting point and duration of the *opening* mode.

If the elastic and inertial effects are abstracted away, the *closed* and *opening* modes are traversed instantaneously in sequence into the *loading* mode. However, the inertial element has a distinct derivative effect on system behavior, and the influence occurs over a small time interval. This is an example of time scale abstraction, where mode change phenomena is expressed at *a point* in time. An important implication is that the state vector has to be modified through the sequence of mode changes. An algebraic relation is derived to compute the elevator velocity to correspond to the fast transient behavior in the mode transitions (Fig. 5 and Fig. 6(a)).

Parameter Abstraction. When dissipation in the land/port clearance dominates the inertial effect, a much faster response in fluid flow velocity occurs because dissipation does not introduce a time derivative effect. The flow of oil into and out of the cylinder is fast, and the pressure build-up in the cylinder is small. As a result, elevator velocity remains almost unchanged as the model switches from *closed* to *opening* (Fig. 6(b)). Small parameter values are abstracted away, and the transitions through the *closed* and *opening* modes are instantaneous (no time derivative effects are present). For small parameter values (Fig. 5 and Fig. 6(b)), the transients to *opening* (Fig. 6) may result in very different behavior from transients into *closed* (Fig. 5). When a discontinuous jump occurs, the eventual elevator velocity is not computed by first executing the jump to *closed* and then to *opening*, but immediately to *opening*. Otherwise, *closed* would have set the velocity to 0, which would also be the value in the *opening* mode. For parameter abstractions the intermediate steps are completely abstracted away.

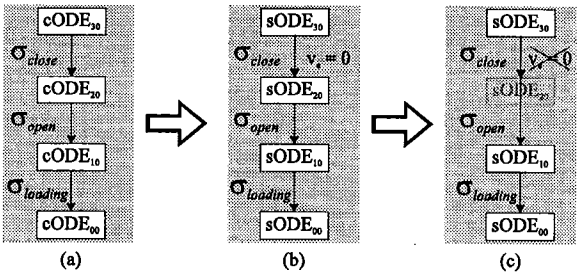

Figure 8: **Hybrid automata specifying the actuator1 elevator subsystem.**

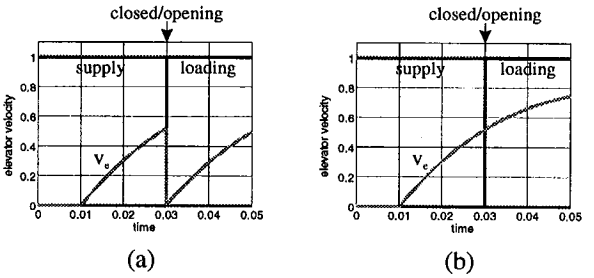

Figure 9: **Effect of a (a) time scale abstraction and (b) parameter abstraction.**

Hybrid Automata for Modeling Complex Systems

Application of time scale and parameter abstractions while producing simpler models for analysis requires the explication of the type of abstraction applied to the model, so that formal semantics can be applied to ensure correct behavior generation. Hybrid automata provide a powerful formalism for specifying hybrid dynamic systems.

Modeling with Hybrid Automata

Fig. 8(a) illustrates the hybrid automata implementation of the switching of an actuator from its *active* to *loading* mode in the elevator subsystem. The behavior models include the fast continuous transients, therefore, they are numerically complex ODEs (cODE). Time scale and parameter abstractions produce numerically simpler ODEs (sODE), but require the specification of discrete transition functions, ϕ, γ, and g. In the first case, transition conditions were based on spool valve position, x. However, in the latter model, the detailed continuous behavior of the system around $x = 0$ is abstracted away, so the corresponding events $\{\sigma_{supply}, \sigma_{close}, \sigma_{open}, \sigma_{load}\}$ have to be generated by explicit discrete control. Our analysis shows that transients into *closed* results in $v_e = 0$. If these transients are abstracted away, a discontinuous jump specified by the hybrid automaton transition sets $v_e = 0$ (Fig. 8(b)).

Time scale abstraction applied to the actuator model produces correct behaviors (Fig. 9(a)). Parameter abstraction, however, produces incorrect behavior

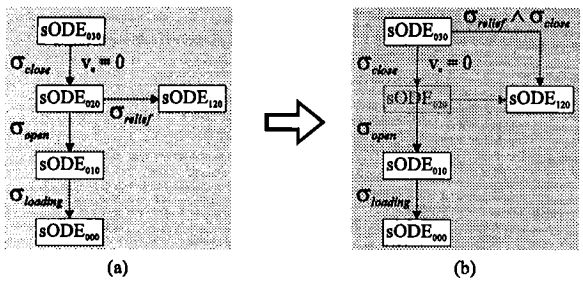

Figure 10: **Mode switching with pressure relief valve.**

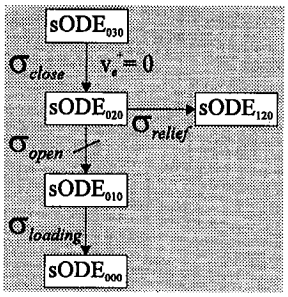

Figure 11: **Compositional hybrid automata.**

(Fig. 9(b)) because the underlying continuous transient that changes $v_e = 0$ in *closed* is aborted by the *opening* mode becoming active, therefore, v_e remains unchanged. In effect, this means that mode α_{20} is never active, i.e., it is *mythical* (Mosterman and Biswas, 1997a), and the change in continuous state never occurred. An intuitive solution would be to remove mode α_{20} from the hybrid automata model, as shown in Fig. 8(c). However, this model reduction requires global applicability conditions about the mode change behavior of the elevator system. Therefore, this approach makes it hard to develop complex system models by composing constituent elements. Additional transitions require the real system to be re-evaluated (in a sense re-modeled) to establish the correct discrete state transition structure for the extended model.

Consider the situation where a pressure relief valve becomes active when the pressure in the cylinder exceeds a threshold value. When the spool valve is closed, a rapid pressure build up occurs induced by the fast change in v_e. If the continuous transients are abstracted into a discontinuous change, this pressure is modeled as a Dirac impulse function whose area is determined by the v_e values immediately before *closed* and initial values in the *closed* mode ($v_e = 0$). If this area exceeds a critical value, the pressure relief valve opens up to prevent excessive pressures in the cylinder. The energy transient undergoes a continuous trajectory, and mode α_{20} should not be removed.

The hybrid automata model for this mode change behavior is shown in Fig. 10(a). To prevent the pressure build-up, the pressure relief valve prevents $v_e = 0$. Therefore, when the pressure relief valve comes on (indicated by a 1 in the left most index of the sODE subscripts) $v_e = 0$ does not apply. However, this information is not available in the hybrid automata when the straightforward extension with σ_{relief} and corresponding sODE_{120} mode is applied (Fig. 10(a)). An exhaustive analysis of the real system is required to reduce the hybrid automata to the one in Fig. 10(b) but now the transition to the relief mode is invoked when σ_{relief} **and** σ_{close} is generated. This results in a non intuitive and complex state transition structure that has little relation to the actual transition behavior of the real system. Detailed pre-analysis of the discrete transition behavior of the complete system is required before hand to generate the correct model, and this complicates the development of compositional modeling techniques in the hybrid automata framework.

Structure Preserving Hybrid Automata

To enable compositional modeling additional transition semantics in the form of discontinuous changes from the *a priori* state vector, x, to the *a posteriori* values, x^+, have been developed (Mosterman *et al.*, 1998b). In combination with recognition of parameter and time scale abstraction events this has proved to be a powerful mechanism for modeling complex systems (Mosterman and Biswas, 1997a). Applications to the secondary sodium cooling system of a fast breeder reactor appear in (Mosterman *et al.*, 1998b). The hybrid automata formalism makes state vector assignments to x^+. Transients enabled by events caused by parameter abstraction are traversed instantaneously and the *a priori* state vector is unchanged. Time scale abstraction generated events cause an update of the *a priori* state to the current *a posteriori* values, $x = x^+$. State transitions with time scale abstraction events are marked by a sloped stroke line.

For the pressure relief valve, the event σ_{relief} is a function of the change in v_e between α_{030} and α_{020}. The v_e value in α_{020} is assigned to the *a posteriori* value v_e^+, and the event generation can be specified as $v_e^+ - v_e > v_{th} \rightarrow \sigma_{relief}$. The illustration of the hybrid automata in Fig. 11 clearly shows that the state transition structure and the corresponding discontinuous jumps in state vector values are preserved while still generating correct behavior. The description is complete, and does not require modifications when new transitions are added to the overall system.

Conclusions

Hybrid automata combine discrete transitions with continuous behavior evolution to provide a powerful formalism for modeling hybrid systems. Discrete transitions cause changes in the system behavior model, but discontinuous changes in the state vector values may also occur. These changes are specified as transitions. We have incorporated the two abstraction types

(i) time scale abstraction and (ii) parameter abstraction and the associated semantics that govern discontinuous changes in behavior specification into our hybrid automaton framework. An important feature of our work is that these abstractions relate back to physical parameters of the physical system that cause fast continuous transients. We have also shown how the use of *a priori* and *a posteriori* switching values help specify the formal semantics for the two types of abstractions. Analysis of the hydraulic cylinder with the pressure relief valve demonstrates the usefulness of this method for compositional modeling of complex systems.

References

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

Guckenheimer, J. and S. Johnson (1995). Planar hybrid systems. In: *Lecture Notes in Computer Science* (P. Antsaklis, W. Kohn, A. Nerode and S. Sastry, Eds.). Vol. 999. Springer-Verlag. pp. 202-225.

Merritt, H.E. (1967). *Hydraulic Control Systems.* John Wiley and Sons. New York.

Mosterman, P.J. and G. Biswas (1997*a*). Formal Specifications for Hybrid Dynamical Systems. In: *IJCAI-97.* Nagoya, Japan. pp. 568–573.

Mosterman, P.J. and G. Biswas (1997*b*). Principles for Modeling, Verification, and Simulation of Hybrid Dynamic Systems. In: *Fifth Intl. Conf. on Hybrid Systems.* Notre Dame, Indiana. pp. 21–27.

Mosterman, P.J., F. Zhao and G. Biswas (1998*a*). An Ontology for Transitions in Dynamic Physical Systems. In: *AAAI98*, Madison, WI, pp. 219-224.

Mosterman, P.J., G. Biswas and J. Sztipanovits (1998*b*). A hybrid modeling and verification paradigm for embedded control systems. *Control Engineering Practice.*

Seebeck, J. (1998). *Modellierung der Redundanzverwaltung von Flugzeugen am Beispiel des ATD durch Petrinetze und Umsetzung der Schaltlogik in C-Code zur Simulationssteuerung.* Diplomarbeit. Arbeitsbereich Flugzeugsystemtechnik. Technische Universität Hamburg-Harburg.