HYBRID DYNAMIC SYSTEMS: A HYBRID BOND GRAPH MODELING PARADIGM AND ITS APPLICATION IN DIAGNOSIS

By

Pieter J. Mosterman

Dissertation

Submitted to the Faculty of the

Graduate School of Vanderbilt University

in partial fulfillment of the requirements

for the degree of

DOCTOR OF PHILOSOPHY

in

Electrical Engineering

May 1997

Nashville, Tennessee

Approved:	Date:
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____

To my parents, Johannes Mosterman and Leny Mosterman-Wolbers.

ACKNOWLEDGEMENTS

Well, this is it. Time to sit back and reflect on the past five years of my life. Unfortunately, there is no overwhelming sense of achievement. I could have accomplished much more had I started working on this topic from day one. Instead, I ended up writing software that did not get me any closer to a dissertation, though it generated some publications [90, 91], and it is now commercially available [89]. It was not until I took matters into my own hand and decided to work with people dedicated to educate me and help me earn a degree. Dr. Biswas and Dr. Sztipanovits have made this endeavor academically possible, and I would like to express my sincerest gratitude towards them. The almost daily interaction with Dr. Biswas and his tremendous work ethic have shaped this research into what it is. None of this would have taken place without Dr. Sztipanovits who gave me the opportunity to pursue my interests. His engineering experience put this work in perspective and his critical questions revealed many implicit assumptions. Furthermore, I wish to thank the other members of my committee, Dr. Cook, Dr. Debelak, Dr. Goldfarb, Dr. Karsai, and Dr. Misra for their interest in my work and their comments that improved the quality of this thesis.

Of course, this thesis would have never materialized if it were not for the undying support of my parents, Johannes Mosterman and Leny Mosterman-Wolbers. I hope I do them proud, and I dedicate this work to them.

"Nobody put a gun to your head," once used as a justification for being exploited, cheated, threatened, and patronized, proves a remarkably concise summary of too much of my gradual student experience. I am not sure how things would have turned out if it were not for my brother, Rudolf, who gave me the strength to endure the

abuse. Furthermore, I would like to thank my sisters, Aleida and Andrea, for their everlasting affection. I missed out on too much these five years. Thanks to Anja Feringa for her love and support even though I am tenthousand miles away.

I will sure not forget to mention others who made the experience bearable by their continued interest, even though I am hardly around anymore, longtime friends Gerard B. Boelens, Haan, Ronald Hesselink, Jeep L. Zeepbel, Lange T., Roelfsma, Ste fan Teuben, and Hanneke de Knegt, Eric Kooistra, Lies Kroon, Katy Smid, Diana (Z)Uchtman, Petra Visser, and Jochum Zijlstra.

I also wish to acknowledge my fellow *euroslave* Eric-Jan Manders, who helped me make one of the defining decisions in my life, my roommate Mark Cambron, Dan Hartman, Martine Dawant, Magued Bishay, Ravi Kapadia, and Narasimhan Sriram. Then there is Herman van der Molen and Alex Zijdenbos who helped me out on a number of occasions and probably did not think much of it. I do. Furthermore, I would like to mention the guys at HP I worked with, Lee Barford, Mel (MegaMel) Downs, and Marsh Faber (no, he is not from Hollum, Ameland), or just hung out with, Al Moye. Thanks also to Karl Oelgeschlager and Andy Covell of Falcon Software for picking up my work and believing in it!

Finally, I would like to acknowledge PNC Japan, Hewlett-Packard Co., Falcon Software, Inc., the Alfred P. Sloan Foundation, the National Science Foundation, Vanderbilt University, and Kampeerbedrijf Mosterman for their financial support. Without it my thirst for knowledge would have ended in a drought.

Thanks to y'all!

Pieter J. Mosterman
Nashville, Tennessee

ELECTRICAL ENGINEERING

HYBRID DYNAMIC SYSTEMS: A HYBRID BOND GRAPH MODELING PARADIGM AND ITS APPLICATION IN DIAGNOSIS

PIETER J. MOSTERMAN

Dissertation under the direction of Professor G. Biswas & Professor J. Sztipanovits

Physical system behavior follows the general principles of *conservation of energy* and *continuity of power*, but may exhibit nonlinearities that result from small, parasitic, effects, or occur on a time scale much smaller than the time scale of interest. At a macroscopic level, the detailed continuous behavior may appear to be discontinuous, thus the system is efficiently described by a mixed continuous/discrete, *hybrid*, model. In continuous modes the energy distribution describes the system state. Discrete configuration changes in the model may cause discontinuities in the energy distribution governed by the principle of *conservation of state*, and may trigger further configuration changes till a new *real* mode is achieved where no further changes occur. The intermediate, *mythical*, modes between two real modes have no physical representation. The principle of *invariance of state* applies to derive the energy distribution in a mode as a function of the energy distribution in the preceding real mode. When a loop of consecutive instantaneous mode changes occurs time stops progressing. This conflicts with known physical system behavior, therefore, the principle of *divergence of time* forms an important model verification mechanism. The principle of *temporal evolution of state* requires the energy state to be continuous in left-closed time

intervals to ensure proper causal attribution.

From another viewpoint, abrupt faults in process components can be modeled as discontinuities that take system behavior away from its nominal, steady state, operation. To quickly isolate the true faults, well constrained hybrid models avoid the inherent intractability problems in diagnostic analyses by integrating and facilitating the (1) generation of behavioral constraints from physical laws, (2) expression of system dynamics as energy transfer between constituent elements, and (3) modeling of steady state behavior as a special case of dynamic behavior. The analysis of transients is paramount to accurate and precise fault isolation. However, this is a difficult problem which can be further complicated by operator intervention, and intermittent and cascading faults, therefore, quick capture and analysis of transients is the key to successful diagnosis.

This thesis develops a formal hybrid modeling theory based on physical principles, a model verification method, and a physically correct behavior generation algorithm. Next, it describes a methodology for monitoring, prediction, and diagnosis of dynamic systems from transient behavior, based on the developed hybrid bond graph modeling paradigm. Simulation results from diagnosing a high-order, nonlinear, model of a liquid sodium cooling system in a nuclear reactor demonstrates the success of the approach.

Approved_____ Date_____

Approved_____ Date_____

TABLE OF CONTENTS

Page

LIST OF FIGURES

LIST OF TABLES

CHAPTER I

INTRODUCTION

The increase in complexity with the introduction of advanced technology into large-scale engineering systems necessitates the use of computer based tools to assist in the design, manufacturing, control, monitoring, and diagnosis of these systems. The success of all these tools is critically dependent on the ability to develop accurate models for simulating, predicting, and verifying system behavior. State of the art engineering design methods rely almost completely on computer based modeling and simulation to avoid the high cost of designing and testing mock-ups and physical prototypes [99].[1] As a typical example of such systems, consider the liquid sodium cooling system in Fig. 1 which constitutes the secondary cooling system in a nuclear reactor [81]. The secondary 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 by pulse width modulation. Actuated valves throughout the loop guard against catastrophic failures. When critical situations occur, the main loop can be closed by a normally opened valve and, if pressure in the piping exceeds a predetermined threshold value, the alarm loop (which consists of the air-cooler) can be activated by opening a normally closed valve. The behavior of the system is

[1] Assembly of the first Boeing 777 showed an unprecedented first-time fit and alignment when a 37680 kg, 63.8 m long wing and fuselage section were found to be out of alignment by only 0.69 mm!

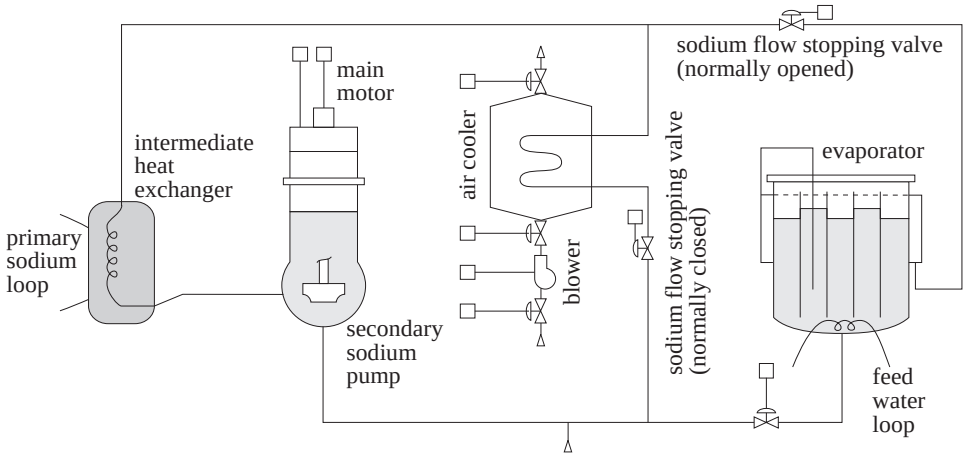


Figure 1: Continuous and discrete process control.

inherently continuous, but the time scale for the opening and closing of the valves is small enough compared to other behavior changes so that they can be modeled to switch instantaneously between on and off states, as a result of which the system seems to operate in distinct modes of operation. Each mode corresponds to distinct on and off states of individual valves and switches.

In each mode, system behavior evolves continuously and at points in time, when valves open and close, discrete mode switches occur. Normally, the valve in the main loop is open and the valve in the alarm loop is closed (Fig. 1). In this mode the amount of liquid stored in the evaporator vessel, and the flow velocities (momentum) in the pipes change in a continuous manner over time. These two sets of variables constitute the state vector of the system. When the valve in the main loop is closed (discrete change), interaction between the discrete changes and continuous dynamic behavior causes a pressure build-up in the piping because the flow momentum is abruptly forced to 0. In this mode, the system can be described with one state variable, i.e., stored liquid since the flow momentum is 0. When the built-up pressure exceeds a

critical value, to avoid catastrophe, the alarm loop is activated and the model moves into yet another configuration where once again stored liquid and flow momentum constitute the state vector. The challenge in modeling these multi-mode systems is to come up with systematic and consistent specifications that govern the interaction between continuous behaviors associated with the individual operational modes, and discrete model parts that specify transfer of the state vector during switching.

Examples of other systems that are best described by multi-mode configurations are auto engine controllers which run quite different control programs as a function of the engine *rpm* [5]. Similarly, the Airbus A-320 fly-by-wire system has a number of modes: *take off*, *cruise*, *approach*, and *go-around* [119]. Models of these systems typically support encapsulation and are derived from the concepts of abstraction, partitioning, and hierarchical modeling [9, 46, 133].

In general, multi-mode system models often arise from a local piecewise linearization of complex nonlinear component descriptions to reduce overall complexity, but system behavior then appears to make discontinuous changes when mode switching occurs. Selection of the appropriate linear component that constitutes the active mode of operation is normally achieved by a meta-level control model that operates on top of the data flow model of a real-time system [73, 132]. Comprehensive models of dynamic physical systems, therefore, require a dedicated signal flow model that selects active model parts.

With the emerging complexity of embedded control systems [133], i.e., physical systems that are controlled by digital computers (Fig. 2), interest in methods to model interaction between the signal and power domain is becoming increasingly important. Process control operations, are now implemented as sophisticated Programming Logic

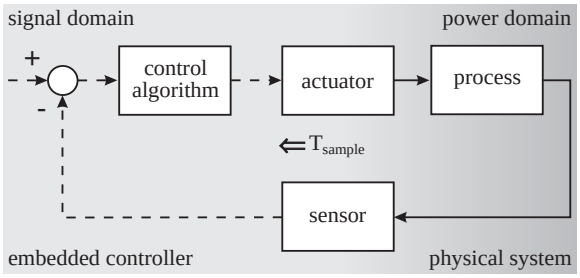


Figure 2: Embedded computer control systems operate in the signals and power domain.

Arrays and/or complex software modules run on embedded computer systems. These digital control mechanisms are discrete in nature and coexist with low-level continuous *proportional*, *integral*, and *derivative* (PID) control [42, 103]. Therefore, complete process control modeling schemes 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.

Hybrid Modeling

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 [133]. This can be achieved by developing a formal description of hybrid systems that combines continuous system characteristics with discrete event models. Continuous system models are well described by differential equations and analytic and numerical simulation methods may be employed in solving these equations [11]. Similarly, a number of approaches with well defined execution semantics, such as Petri nets and

finite state automata, have been applied to discrete system modeling [109, 110]. Systems with mixed continuous/discrete components need model semantics that combine these two approaches, and simulation schemes that can seamlessly combine continuous behavior generation with discrete mode switches. Lygeros, Godbole and Sastry [66] have shown that independent determination and proofs about the continuous behavior and the discrete phenomena in a hybrid model do not constitute proofs of correctness of their combined effects.

To develop hybrid models that generate correct system behavior, interaction between the continuous and discrete modeling formalisms has to be rigorous, unambiguous, and consistent. Consider the electrical circuit in Fig. 3 which resembles behavior of the liquid sodium system in Fig. 1. The inductor models the helical coil in the intermediate heat exchanger and the diode resembles operation of the pressure controlled alarm switch. When the manual switch is closed, the inductor is connected to the source and builds up a flux, p_0, by drawing a current (Fig. 4). The diode is not active in this mode of operation. When the switch is opened, the current drawn by the inductor drops to 0, causing its flux, p_0, to discharge instantaneously. Because of the derivative nature of the constituent relation $V_L = L\frac{dp}{dt}$, the result is an infinite negative (the flux changes from a positive value to 0) voltage across the diode. Because its threshold value, V_{diode}, is exceeded, the diode comes on instantaneously and the mode of operation where the switch was open and the diode inactive is never realized in real time. If it were, the stored energy of the inductor would be released instantaneously in a mode where the model has no real representation, producing an incorrect energy balance in the overall system. Consequently, there would be no flow of current after the diode becomes active. This shows that the flux of the inductor

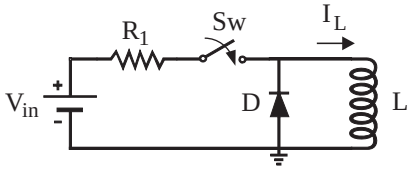


Figure 3: Physical system with discontinuities.

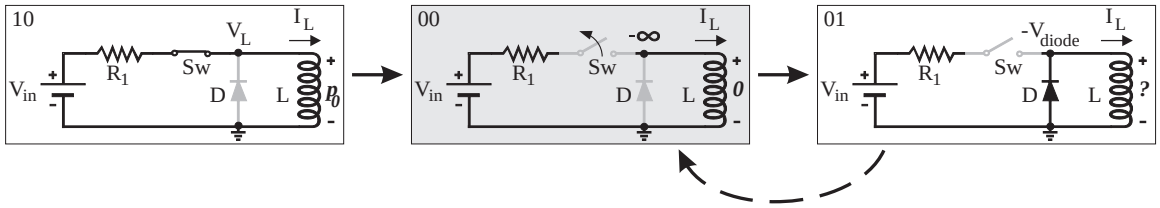


Figure 4: A series of mode switches may occur.

when the diode comes on should be computed based on the flux before switching started, p_0.

Consider a scenario where the diode requires a threshold current $I_{th} > 0$ to remain on. If the inductor has built up a positive flux, the diode comes on when the switch opens. However, if the flux in the inductor is too low to maintain the threshold current, given a certain inductance, the diode goes off instantaneously, but when it is off, the voltage drop exceeds the threshold voltage again. The model goes into a loop of instantaneous changes (see dashed arrow in Fig. 4) during which real time does not progress or diverge, and this conflicts with the notion that in the physical world time does not halt. This example illustrates a number of characteristics specific to hybrid system models:

- When changes in the model configuration, *mode of operation*, occur, the system state vector has to be transferred correctly. This is complicated by discontinuous changes in state variables and even the state vector itself may change.

- When a mode change occurs, a number of consecutive instantaneous changes may follow. This complicates behavior generation because:

    1. The final mode in which instantaneous mode changes cease has to be derived correctly.

    2. The state vector in the final mode has to be computed across a number of intermediate modes.

    3. When a sequence of mode changes starts, it should not end up in a loop of instantaneous change, as this would prevent real time from progressing.

- Discontinuous changes in state variables may require computation of limit values in time. If limit values at switching time, t_s, require future knowledge of system behavior, $\lim_{t \downarrow t_s}$, the model is acausal, and, consequently, ill defined.

One of the primary contributions of this thesis is the development of a theory for hybrid modeling of dynamic physical systems. The modeling scheme has three components: (1) a differential equation model of continuous system behavior, associated with the operational modes of the system, (2) a discrete-event model, based on finite state automata for handling mode transitions, and (3) an algorithm for correctly transferring the system state vector from one operational mode to another through a sequence of transitions. A systematic set of principles are developed to correctly specify semantics and constraints to ensure that the models generate correct behavior. The increased sophistication and complexity of current engineering systems has also increased by several levels of difficulty the task of monitoring system behavior, and keeping systems operational. A second contribution of this thesis investigates

the use of these modeling methodologies in developing better monitoring, prediction, and diagnosis methodologies for hybrid systems.

Model Based Diagnosis

Economic constraints on commercial systems such as automobiles and chemical plants, mandate ever stricter demands on maximum down time. To meet these constraints, diagnosis methodologies can be developed to predict and identify which components are about to fail, and quickly detect failures before they reach catastrophic proportions [23]. Initial work in Fault Detection and Isolation (FDI) relied on *hardware redundancy*. Multiple hardware components, such as actuators, sensors and process components, at different points in a system provided redundancy in function to avoid failure. For example, critical measurement points were equipped with multiple sensors for detecting discrepancies, and schemes such as the majority vote method were applied for reliable detection of signal and parameter deviations, which were then directly mapped into specific fault scenarios. As systems became complex, the hardware requirements for FDI became excessive, both in terms of cost and space. Furthermore, processing of a large number of sensor signals for fault isolation greatly increased processor and memory requirements. Therefore, *functional redundancy* has become the preferred approach to FDI [23].

Functional redundancy schemes measure system variable values at different points in the system, and use relations imposed by the system configuration and functionality to study discrepancies among the measured values [113, 115]. When faults occur in the system, observed deviations in measurement values are analyzed using the system model to generate a set of possible faults. A fault implicates one or more components

of the system and explains all the observed measurements: deviating and normal. These faults are then used to predict future behaviors of the observed variables based on the system model.

Generally, faults can be characterized as [23]:

- Incipient faults; these faults occur slowly over time and are the result of, e.g., wear and tear.
- Intermittent faults; these faults are only present for a very short time but can be disastrous.
- Abrupt faults; these faults are dramatic and persistent, they cause deviations from steady state operations and move the system into new steady state conditions or, after some transient behavior, return to the original steady state.

Abrupt and incipient faults exhibit different behavior that may, and in general will, require different diagnosis strategies. Moreover, a fault manifestation may not persist (the system may be halted before steady state is reached, new faults may occur, or faults may be intermittent), therefore, it is essential to track and analyze system behavior at frequent intervals.[2] The goal is to capture the transient behaviors that occur in response to a fault, because they are often the best clue for identifying and isolating faulty components in dynamic systems before compensating mechanisms start altering the transient characteristics. Modeling, tracking, interpreting, and analyzing dynamic systems and transient behavior is a difficult task. To eliminate the modeling difficulties but to keep the dynamic, discriminative, information, several methods have been proposed that perform diagnosis based on deviations from a static model

[2]It is a frustrating experience to take your car in because it is malfunctioning, only to find that the problem does not seem to occur in the presence of a mechanic.

[69, 102]. However, these methods result in underconstrained process models. This especially causes problems for larger systems where the number and size of fault candidate sets explode and the diagnosis problem becomes intractable. To prevent this exponential blow-up, highly constrained models can be used based on physical laws of *continuity of power* and *conservation of energy*, and these models are inherently continuous.

The primary focus of this research is on abrupt faults. When a faulty situation is detected, the fault isolation and prediction methodologies rely on a model of the system to reason about its dynamic behavior. Behavior in each mode is continuous but abrupt faults introduce discontinuities at the point where faults occur. Moreover, when faults occur, the system may undergo structural changes, e.g., when a valve is closed during normal operation in the cooling system in Fig. 1, the system is split in two independent systems. The model needs to have the capability to incorporate such discrete structural changes which may affect the causal relations among parameters and variables in a system. In the specific case of the cooling system, when a valve closes abruptly it forces flow of liquid to zero that was previously free in its behavior. This causal change predicts excessive build-up of pressure (voltage in the equivalent diode-inductor circuit in Fig. 3) and demonstrates the need for an alarm loop to prevent damage.

Contributions and Organization of the Thesis

This thesis develops a systematic modeling and analysis framework for hybrid dynamical systems. It develops formal specifications of a hybrid modeling paradigm for dynamic physical systems and provides systematic principles that govern behavior

generation. The second part of the thesis uses the systematic modeling framework to develop a methodology for monitoring, prediction, and diagnosis of abrupt faults in complex, dynamic systems.

Chapter II of the thesis reviews traditional mathematical models of physical systems and bond graphs, a systematic approach for modeling the continuous characteristics of physical systems. Chapter III presents a detailed study of the nature and effects of discontinuities in physical system models. Chapter IV develops a systematic set of principles to handle discontinuous changes in system behavior, and the bond graph formalism is augmented to incorporate discrete modeling concepts whose semantics are in keeping with the principles discussed. To ensure the interaction between the continuous and discrete parts of the resulting hybrid bond graph modeling paradigm is consistent, rigorous, and unambiguous, a multiple energy phase space analysis to verify physical correctness of models is developed in Chapter V. The formal mathematical specification for a general hybrid model is developed in Chapter VI and Chapter VII shows how the hybrid bond graph model described in Chapters IV and V can be exploited to systematically derive the required mathematical specifications of the model components.

In the second half of the thesis, the modeling methodology developed in the first part is used to model and analyze the transient behavior of a system in response to abrupt faults. Chapter VIII develops a comprehensive architecture for process monitoring and diagnosis starting from a hybrid system model and transforming it into a temporal causal graph for prediction and diagnostic analyses. Chapter IX develops and explains the algorithms for measurement selection, possible fault generation, and prediction of future behaviors under fault conditions. The methodology developed in

Chapters VIII and IX is applied to the liquid sodium secondary cooling system in a nuclear reactor described in this chapter, in Chapter X. A summary and discussion of the accomplishments of this thesis and directions for future research are presented in Chapter XI.

CHAPTER II

MODELING DYNAMIC PHYSICAL SYSTEMS

Good modeling schemes for physical systems must include features that capture the set of salient characteristics that help define physical system behavior. A key characteristic is that physical system behavior is continuous in time, and the most general mathematical model for expressing such behaviors relies on differential equations, possibly extended with algebraic constraints. This section reviews physical systems theory, and their mathematical representations, and then presents the bond graph modeling language as a systematic approach to modeling physical systems.

Physical Systems Theory

Systems theory focuses on describing dynamic behavior of objects and mechanisms of interest which are collectively called a *system* [65, 130].[1] In some cases, a system is a collection of phenomena that can be observed. The set of phenomena, their influences and observations made on the system determine its *boundary*. Everything that does not belong to the system is called its *environment* and interactions between them define the *system context* (Fig. 5). Behavior of physical systems is governed by the laws of physics. This thesis focuses on a particular class of physical systems, those that are man-made or engineered. They are referred to as *engineering systems*.

[1] By defining a system like this, it does not preclude other definitions of a system.

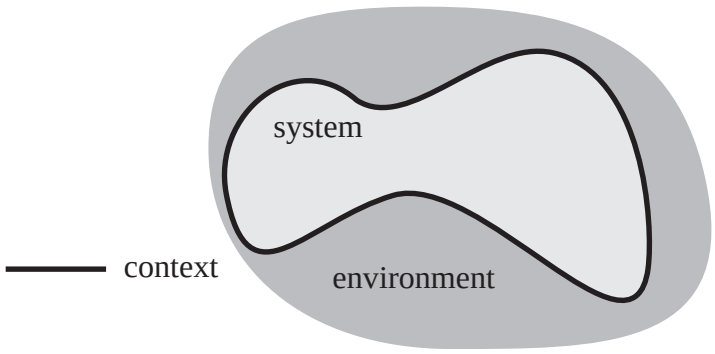


Figure 5: Definition of a system.

Typically, system behavior is derived by applying physical laws to a system description. As the physical or engineering system under scrutiny becomes more complex, additional systematic considerations have to be taken into account for describing and analyzing system behavior. These systematic considerations are often looked upon as a theory for translating real situations into more abstract forms for analysis, called *models*.

> *A model of a concrete system is a description of that system, based on the application of existing theories* [130].

The quality of a model is often based on how well its behaviors of interest match the real phenomena under study. Though a model may be *verified* to be correct in theory, i.e., it violates no physical laws, *validation* of the model with respect to the phenomena of interest is essential before its usefulness can be determined.[2] Models that conform to an underlying modeling theory are called *theoretical models*, otherwise they are *descriptive models*. Descriptive models give a formal description of how

[2] A model of a bicycle can be verified to be consistent with the laws of physics. It still has to be validated whether it describes the behaviors of interest of the system under consideration correctly (which might be a car).

the phenomena of interest in a system behave, and they can only be validated in comparison to the real phenomena. On the other hand, models based on underlying theory allow for additional checking, verification, in terms of conformance to physical laws. For example, a descriptive model of an electrical wire can be: *The wire causes a voltage drop which increases linearly with the length of the wire.* A corresponding more general theoretical representation is Ohm's law which defines the concept of *resistance* and introduces a theory for voltage-current dependency based on this concept.

Sir Isaac Newton was the first to record formal theories in the domain of *mechanics.* Soon formal theories in other domains, such as fluid mechanics, electricity, and magnetism were developed. Further, observations that any physical domain can interact with another by means of energy exchange influenced the beginnings of physical systems theories in thermodynamics, which concentrates on thermal processes which are ubiquitous in physical systems. The emergence of *network analysis* as powerful tool made it beneficial to describe the different domains that constituted physical systems in terms of an electrical equivalent, i.e., an interconnected topology of energy sources, dissipators (resistors), energy storage elements, such as capacitors and inductors, and transformer and gyrator elements that convert energy from one form to another. More recently, *control* and *information* theories have been established. Systems theory unifies and generalizes formal theories from various domains into a common mathematical framework, usually in the form of a set of differential equations.

From Reality to Mathematical Models

To efficiently analyze, design, control, and understand physical systems, it is desired to represent their behaviors in a language that captures the salient aspects of the behavior, allows formal analysis, and at the same time allows abstraction so that the computational burden of the analysis is not overwhelming. Mathematical formulations provide a high level of abstraction where physical characteristics become implicit but the systematic and uniform notation allows for formal analysis in a domain independent way.

Abstraction

Physical reality typically embodies numerous phenomena, a lot of which may be secondary to the gross behavior of interest for the problem being addressed. To prevent unnecessary complexity in tasks such as design, analysis, and control it is desirable to only capture those aspects of the system that are of immediate interest to the behaviors in question. This process of reducing complexity by eliminating peripheral phenomena is called *abstraction* [43], a technique that plays an important role in the construction of system models. Often it is not obvious which phenomena and interactions actually govern the behaviors of interest, therefore, the application of abstraction techniques in generating system models involves trial and error and iterative processing. After an initial model is established, it has to be validated in terms of how well it represents the behaviors that it is required to capture. When discrepancies occur, the model has to be refined (e.g., by increasing its order) or adapted (e.g., by modifying parameters) to more accurately reflect the actual system. This iterative process is repeated until system behaviors of interest are replicated

satisfactorily.

Complex system models are often constructed by considering the system to be a *composition* of a set of entities. Each entity is modeled separately, based on the concept of *reticulation* which assumes that certain properties of a system can be isolated and lumped into processes with well defined parameter values, and the system can be defined as a network of interacting processes [14, 104]. To achieve compositional modeling three properties have to be satisfied: (1) *decomposition*, (2) *classification*, and (3) *representation*. Successful application of these concepts to modeling relies on *typing* and *port based interfacing*. Typing enforces correctness of object usage and it allows encapsulation of local information of an object. A port based interface connects entities that can be either constitutive relations or networks of entities themselves [27]. Therefore, the *lumped parameter* assumption in modeling allows the definition of a system as a composition of entities with each entity having its own constitutive relations that can be expressed in a mathematical form. The network or compositional structure defines the system configuration. System configuration can be abstracted away by composition of the mathematical relations of each entity into a system of equations.

A Systematic Approach to Abstraction

A general theory of modeling defines methodologies that support a succession of abstractions for a physical system domain and its component structure. The most abstract representational form is a mathematical set of equations (Fig. 6). There are two basic approaches for deriving the mathematical form [65]: (1) *state space* representations, and (2) *transfer functions*. The mathematical representations generated

by these approaches are equivalent, and the methods are complementary.

The state space approach first translates the ideal physical model into a physical analog[3] that defines the level of abstraction. Alternately, a generic representation of physical mechanisms can be used which moves closer to the mathematical representation by discarding domain specific information [54]. Next, the equations for each of the components of the generic model can be compiled at which point symmetric constituent equations are obtained and causality is lost. Then, connections between equations can be established by substituting variables represented by using a block diagram. So, the block diagram helps combine *local* component equations into a *global* mathematical scheme. In the final step, the block diagram can be translated into a system of equations which is a pure mathematical description of global system behavior that does not allow to trace back through the abstraction stages.

To obtain a mathematical model in terms of a transfer function representation, the overall system is decomposed into functional components such that individual components are made up of highly interacting system parts, and there is little interaction between components. In the cooling system example shown in Fig. 6, these parts are chosen to be the intermediate heat exchanger and the evaporator/motor sub-system. Next, the order of these parts and their parameters in the frequency domain are estimated. The connected parts can then be combined into one transfer function in the frequency domain, $H(s)$, which can be transformed into a mathematical model as a difference or differential equation. Alternatively, partial fraction expansion can be applied to the overall transfer function to obtain a summation of

[3]The mechanics of the Philips CD player, for example, were simulated by translating their model into the electrical analog and using SPICE for simulation.

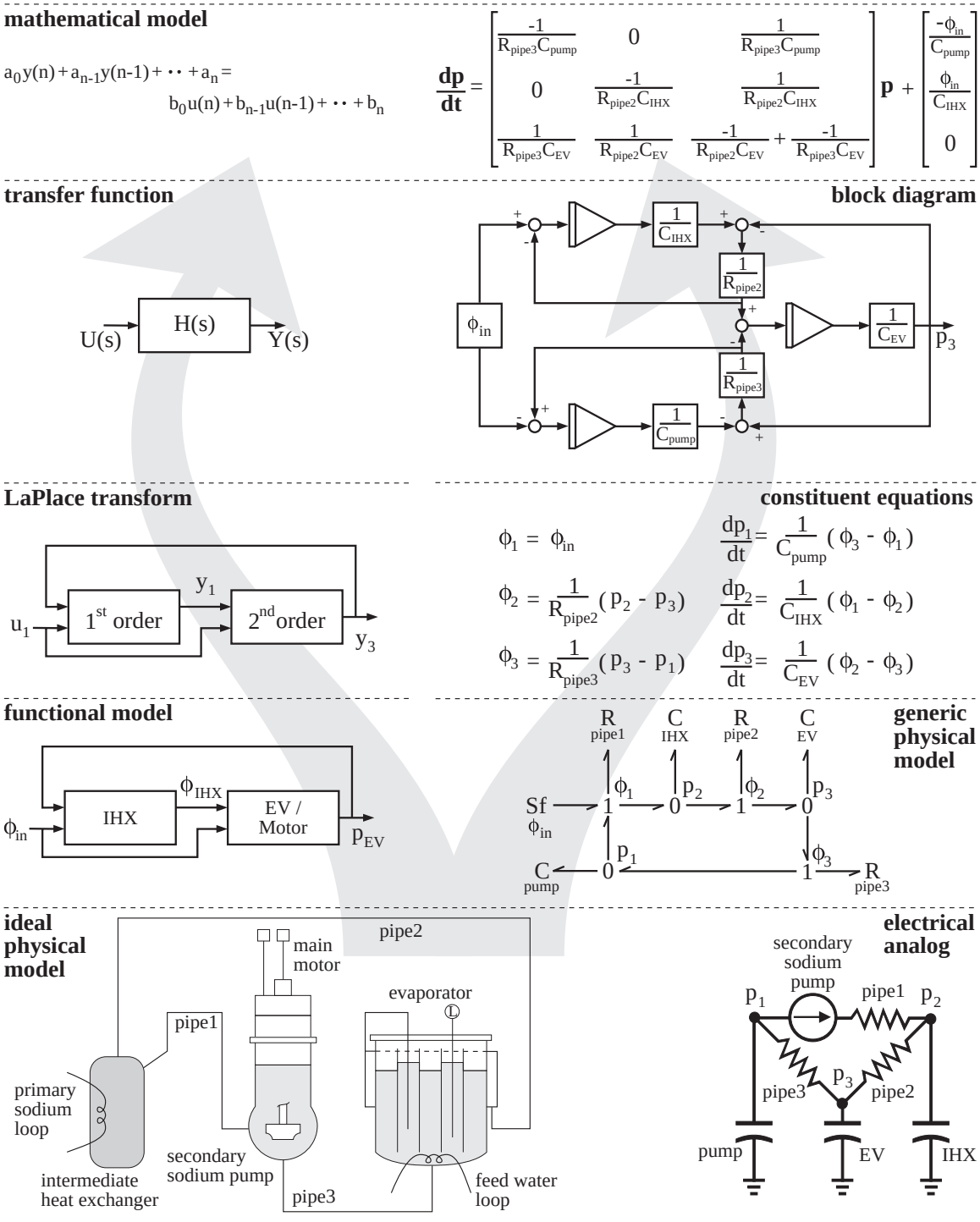


Figure 6: The stages of the modeling process with increasing abstraction.

lower order constituent responses. These lower order transfer functions can then be transformed to the time domain to establish a mathematical model as a summation of impulse responses.

Bond Graph Modeling

In the late fifties, Paynter was able to synthesize the similarities between physical system characteristics in the electrical, mechanical, and hydraulic domains, specifically in terms of power and energy transfer. He exploited these similarities to develop a generic modeling language called *bond graphs* [104]. A number of other researchers have added components to the language, particularly Breedveld [14], who developed sound principles for bond graph analysis based on the laws of thermodynamics [7, 20, 35].

Energy Based Modeling

Bond graphs, based on modeling of the energy content and transfer in physical systems, adopt the lumped parameter approach to modeling and describe a physical system at any given time as energy distributions over connected physical elements. This energy distribution reflects the history of the system, and defines its *state*. Future behavior is determined by its current state description, and subsequent *input* to the system. Changes in state of a physical system are attributed to energy exchange among its components, which can be expressed in terms of the time derivative or flow of energy, i.e., *power*. Irrespective of domain (e.g., mechanical, fluid, pneumatic, and electrical) power is the product of two conjugate variables: the *intensive* variable or

Table 1: Energy Variables in Different Physical Domains

Domain	**Effort** $e(t)$	**Flow** $f(t)$	**Momentum** $p = \int e \cdot dt$	**Displacement** $q = \int f \cdot dt$
Mechanical				
Linear	force, F	velocity, v	momentum, p	distance, x
Rotation	torque, τ	angular velocity, ω	angular momentum, h	angle, θ
Electrical	voltage, V	current, i	flux, λ	charge, q
Hydraulic	pressure, P	volume flow, $\dot{V}$	flow momentum, L	volume, V
Thermal	temperature, T	entropy flowrate, $\dot{S}$	–	entropy, S

effort, e, and the *extensive* variable or flow, f (Table 1).[4] Therefore, effort and flow are called *power* or *signal variables*. Intensive variables are specified at points in a system (e.g., pressure, temperature), and may vary from point to point. Extensive variables on the other hand, are defined over an extent (e.g., volume, charge), and are typically additive in nature.[5] For example, if one considers two blocks of the same material at the same temperature, and brings them together to form one system, the volume of the overall system is the sum of the individual volumes. On the other hand, the temperature of the combined system remains the same.

The Model Context

The first law of thermodynamics states that

> *internal energy is conserved in processes taking place in an isolated system* [30].

For a system to adhere to the first law it has to be isolated. However, a completely isolated system is of little practical use, and the conservation of energy principle is

[4] More precise, the flow variable is the time derivative of the extensive variable, q, as formulated by the free energy equation $dE = \sum e dq$.

[5] Though variables of an additive nature are extensive, not all extensive variables are necessarily additive in nature. This is particularly true in the case of fields.

applied to a system by explicitly specifying its energy interaction with the environment. This represents the system *context* and is modeled by *sources* and *sinks* of effort and flow, S_e and S_f, respectively. The change of energy in a system can be attributed to losses by dissipation through resistive elements. This energy loss needs to be modeled explicitly as a source of entropy in the case where it constitutes free energy. In an isothermal environment, this flow of energy to the thermal domain is not modeled explicitly. Although conservation of energy is the most fundamental law of physics, it is the hardest to enforce [35, 105], since only the significant interactions are captured by the system model and this may not capture all the energy interactions that occur. An additional assumption in macrophysics is the restriction of *power continuity*, which follows from the assumption of conservation of energy. It is observed that energy cannot be annihilated at one point in a system and produced at the same rate at another point. It has to traverse the intermediate space [14, 104]. Therefore, any physical system not only conserves energy, but by nature is continuous in its signal or power variables, effort and flow.

Primitive Elements in Bond Graph Models

Energy can be represented as stored effort and stored flow. The energy corresponding to stored effort is called *generalized momentum*, p, and the energy corresponding to stored flow is *generalized displacement*, q. Consequently, p and q are called *energy variables*, and constituent elements that store generalized momentum and displacement in the bond graph framework are called inductors, I, and capacitors, C, respectively (Table 2). These ideal energy storage element relations are shown by

the *tetrahedron of state* [107] in Fig. 7, and represent the reversible processes in nature. Because of their *integrating* nature, the actual energy stored in these elements is a function of the initial value of the energy they contain. Each initial value, therefore, introduces a *degree of freedom* in the system and an additional dimension to the state vector. Note that this does not necessarily add to the *order* of a system. For example, two capacitors in series with a resistor is still a first order system. If the integral form cannot be used, the stored energy value is completely determined by the other components in the system model, and the element does not introduce an additional degree of freedom.[6] Typically, this occurs when

- a source or sink is modeled to enforce a specific amount of stored energy on a storage element (source-storage dependency), or
- storage elements are directly connected to each other without intervening dissipators (storage-storage dependency).

These two situations can be directly attributed to choices made when designing the system model. In the first case, component mechanisms that are assumed to have insignificant effects with respect to the modeling task or scope are neglected and dependency of storage elements only introduces additional loading effects. In the second case, the dependent storage element most likely represent the same effect. The lumped parameter assumption can be extended to replace the dependent storage elements by their combined equivalent.

Irreversible processes are represented by the dissipative element, R. The S_e, S_f, C, I, and R elements exchange energy via ports. To connect more than two basic

[6] In bond graph terminology, this storage element is then said to operate in *derivative causality*.

Table 2: Generic elements in Different Domains

Domain	**Resistance** $e(t) = R \cdot f(t)$	**Capacitance** $e(t) = \frac{1}{C}\int f(t)dt$	**Inductance** $f(t) = \frac{1}{I}\int e(t)dt$
Mechanical	dashpot	spring	mass
Electrical	resistor	capacitor	inductor
Hydraulic	pipe, valve	tank	narrow pipe
Thermal	thermal resistance	thermal capacity	–

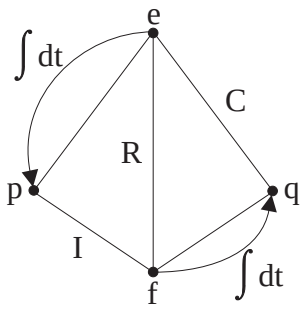


Figure 7: The tetrahedron of state.

elements together, a *junction* structure is required. Junctions typically allow an arbitrary number of components to be connected together. They preserve continuity of power by adhering to the generalized forms of Kirchoff's current and voltage laws, which define the two forms of junctions, 0- and 1-junctions, respectively. The 0- and 1- junctions are illustrated in Fig. 8. Junction relations are instantaneous, i.e., they do not introduce temporal effects. Two special types of junctions, or signal transformers; the transformer, TF, and the gyrator, GY, complete the nine basic elements in the bond graph language. The transformer establishes a ratio between input and output efforts and flows as shown in Fig. 9. It can be used as an *impedance transformer* within a physical domain, and as a *class transformer* between domains [122]. The gyrator operates similarly, by establishing a relation between input effort and output flow, or between input flow and output effort.

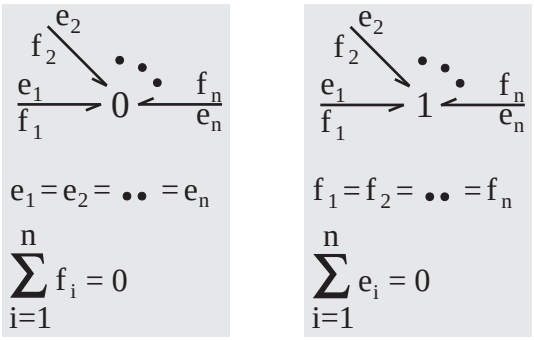


Figure 8: Continuity of power across junctions is ensured by their constituent equations.

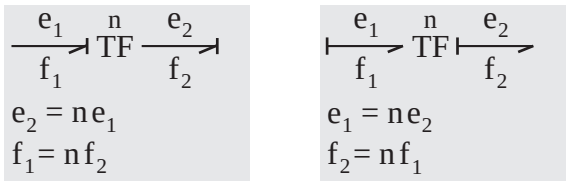


Figure 9: Relations of the transformer for both of the possible causality assignments.

Systematic Modeling

Overall, there exists a systematic procedure for building bond graph models of physical systems [107]. This procedure is illustrated by two examples.

Bi-Tank System

To derive the bond graph model for a simple bi-tank system illustrated in Fig. 10, common pressure points are identified first. In this system, the two important pressures are the ones at the bottom of the tanks, e_2 and e_7. A 0-junction defines each of these variables and connects to the storage elements that represent the tank capacities, C_1 and C_2. The two 0-junctions exchange fluid via a common flow connection, the 1-junction, and the dissipative element R is connected to this junction to represent the pressure drop across the connecting pipe. The flow variable f_5, represents

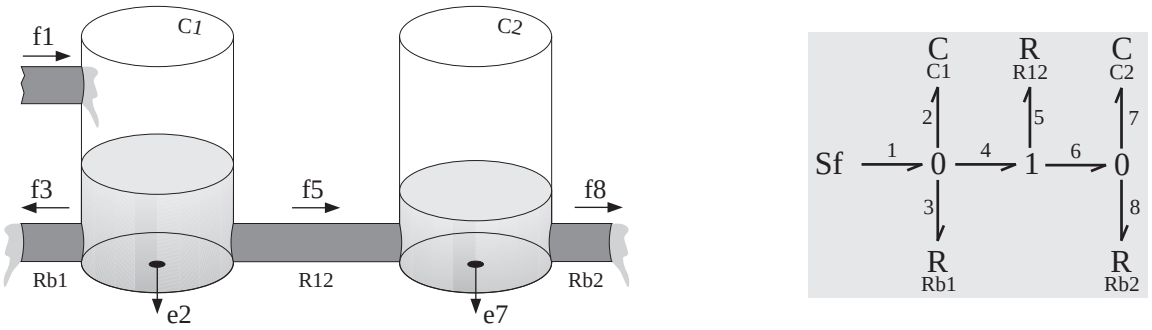


Figure 10: The bi-tank system and its bond graph model.

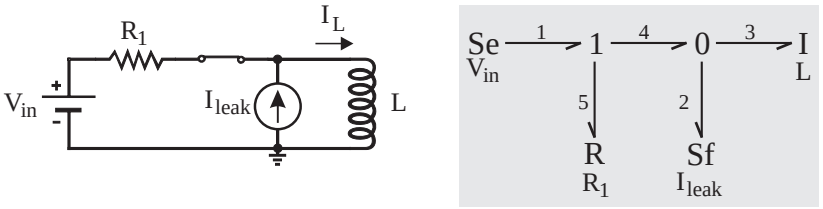


Figure 11: An electrical circuit.

the corresponding volume flow. The Bernoulli outflow resistances are modeled by two dissipators as well, where their outflow depends on the pressure at the bottom of the tanks with respect to a reference value (typically a sump-pressure). The inflow into the left tank is independent of its pressure at the bottom, and, therefore, it is represented as an ideal flow source.

An Electrical Circuit

A second example derives the bond graph for the electrical circuit in Fig. 11. First, 0-junctions are associated with all, common voltage, nodes. In the particular circuit there is one node (apart from ground) with three branches. The current source, I_{leak}, and inductor, L, can be directly connected to the 0-junction. The remaining branch consists of a series or common flow connection, 1-junction, between R_1 and V_{in}.

Causality

A causal structure can be imposed on a bond graph model, using local constraint relations among the components associated with a junction.[7] A systematic algorithm for causality assignment is the Sequential Causality Assignment Procedure (SCAP) [107]. This algorithm categorizes local constraints as (1) *enforced* causality, by sources S_e and S_f, (2) *preferred* causality, for energy storage elements, C and I, and (3) *indifferent* causality for resistances R. The constraints are applied in the order above, with an effort source always imposing an effort causality on a junction and a flow source always imposing a flow causality on a junction. Opposite assignment of causality to a source indicates a physical incorrectness of the model, e.g., a shorted voltage source. The preferred causality for energy storage elements is for them to operate as integrators as opposed to differentiators. The integral relation establishes natural dependence among the effort or flow variables associated with a bond.[8] For example, the implication of the integral form for a C element is that it prefers to deliver effort on a junction, i.e.,

$$e = \frac{1}{C} \int f dt \tag{1}$$

An I element prefers to enforce flow on a junction, i.e.,

$$f = \frac{1}{I} \int e dt \tag{2}$$

Resistive elements have no preference of causality, they conform with how they are driven.

[7] In pathological cases, global constraints have to be used as well. This typically happens in cases when closed power loops occur.

[8] These relations cannot be recovered from the differential relation for lack of knowledge of the integration constant [25].

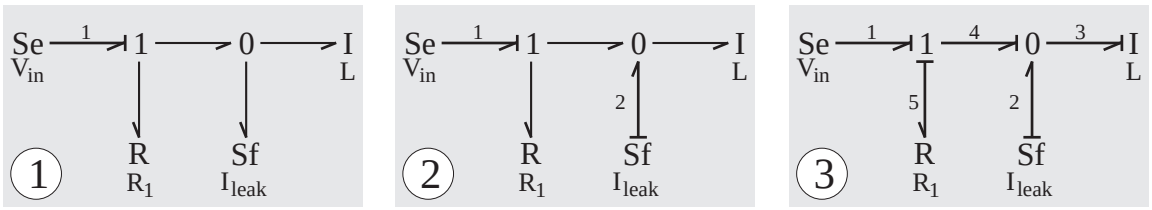


Figure 12: Algorithmic assignment of causality.

Consider the electrical circuit and its bond graph in Fig. 11. Fig. 12 depicts the causality assignment procedure. The first step selects the voltage source, S_e and assigns it *effort causality* which means it enforces effort on the adjacent 1-junction, depicted by the perpendicular stroke at the end of its connecting *power bond* (1) in Fig. 12. Because the effort constraint on a 1-junction requires the sum of all efforts to equal zero, the effort values of the two remaining bonds are still undetermined. Therefore, the assigned causality does not propagate any further. In the second step, the flow source S_f is selected and enforces flow on its adjacent 0-junction depicted by the perpendicular stroke at the beginning of bond (2). Again, causality on all other bonds of the junction is still undetermined. In the third step, L is selected and its preferred causality enforces flow on the 0-junction through bond (3). By default the remaining bond (4) has to impose effort causality on the 0-junction, and, therefore, a flow on the 1-junction. This determines the flow value at all other bonds connected to the 1-junction (common flow), so resistance R_1, through bond (5) has to impose effort on the 1-junction. Therefore, R_1 is in effort causality. In terms of the electrical circuit, this assignment reveals that the current through R_1 is determined by I_{leak} and L. As long as the corresponding voltage drop across R_1 differs from V_{in} the inductor charges.

Qualitative Reasoning with Bond Graphs

Bond graphs provide a ripe structure for pure qualitative reasoning about physical system behavior. First of all, the differential equations directly derived from the model can be used as qualitative constraints in a QSIM-like constraint-centred simulation and analysis [10, 67, 134]. In this regard, bond graphs provide the advantages of a methodology well grounded in physical laws. This contrasts with the approach used in QSIM [57] that focuses more on pure mathematical constraints and QPT [40] that models component and process descriptions individually. The inability to build in conservation of energy and continuity of power relations explicitly into the model, and the lack of well defined primitives can often lead to the development of rather *ad hoc* models. In a sense, QPT draws many parallels with bond graphs, but its lack of a small set of constituent processes that define the theory universally makes it hard to verify physical correctness. Imposing power continuity and energy conservation explicitly eliminate large numbers of spurious behaviors and imprecise parameter estimation [116]. Second, bond graphs establish global causality assignment based on local constraints derived from constituent relations of its primitive components (see previous section), enabling the use of global constraints to refine system behavior. Third, much like human experts, the bond graph framework allows several qualitative aspects of system behavior be derived from physical structure and topology as opposed to mathematical equations. For a linear system these translate to concepts of *controllability*, *observability*, *state variables*, *order*, and *number of distinct eigenvalues* (degrees of freedom). These qualitative characteristics have proven to be of great value to system engineers and their derivation in a bond graph modeling framework is algorithmic [37, 114]. An effective technique to apply these qualitative notions in

complex systems is *condensation* [127]. Condensation allows one to inspect qualitative behavior of sub-systems of zero, first and second order, so that their possible behaviors can be labeled as static, exponential, or oscillating, respectively. This enables qualitative analysis at a level intermediate between the individual interactions in the system and the complete global structure.

Causal analysis is of paramount importance to the successful use of models in conceptual design, tutoring, and diagnosis. In previous work, Iwasaki and Simon [52] have used a basic set of constituent elements as the basic building block for defining causal relations. Furthermore, their work illustrates that causality is determined jointly by individual mechanisms and interactions with adjoining mechanisms which is governed by propagation of effects through connecting junction structures. The SCAP algorithm for deriving causal relations in bond graphs is based on exactly the same set of principles, where the model topology and the energy interactions among components defines causal assignments [16]. This distinguishes the bond graph causality assignment scheme from the Iwasaki and Simon method, which attempts to derive the causal constraints from acausal constraint equations, often resulting in *ad hoc* assignments.

Applications

Bond graph modeling has been successfully applied in the areas of analysis [49, 61, 118], general design tasks [37, 106] and design of mechatronic systems [28, 112], diagnosis [10, 22], and teaching [32, 58]. An extensive bibliography [38] includes applications to mechanical systems, thermal and thermodynamic systems, biological and physiological systems, chemical systems, fluidic systems, electrical systems, economic

and social systems, magnetic systems, acoustic systems, agricultural systems, solar systems and nuclear systems [38].

Summary

It is important to note that the choice of modeling method is very dependent on the type of system being modeled and the task for which the model is being constructed. For example, informational systems[9] are not well suited for energy exchange based modeling [73], because of their discrete nature. Petri nets, finite state automata [56], object modeling, Timed CSP [26], and discrete event systems are more suitable. In other cases, implicit modeling techniques [72] may be advantageous. Continuous systems are best modeled by differential equations, supplemented by algebraic constraints, if necessary. Presently tools that incorporate multiple modeling techniques are being developed to present the user with a generalized environment and allow for interaction between the methodologies [120, 133].

Bond graphs represent a component in terms of its basic physical concepts such as energy dissipation or energy storage. On the one end, it supports model structure analysis based on component aspects of a dynamic physical system (rather than specific parameter values). On the other end of the spectrum, it provides a comprehensive and systematic approach to generating the describing differential equations. Moreover, because of its compositional characteristics, it supports partitioning and hierarchical modeling of increasingly complex systems as well as modifying a particular sub-system to a more detailed model [34, 54].

[9] Informational systems are those that handle data in its general form. So, no *a priori* modeling constraints on data exist.

CHAPTER III

A THEORY OF DISCONTINUITIES IN PHYSICAL SYSTEM MODELS

This chapter introduces the types of modeling abstractions and the discontinuous behavior they cause in system models. The resultant mixed behavior patterns, continuous with intervening discrete jumps, are systematically modeled by combining bond graphs and finite state automata in the modeling scheme.

The Nature of Discontinuities

Physical systems by nature are continuous, and discontinuities are artifacts of simplifications and assumptions introduced into the system model. In general, discontinuities in behavior generation can be attributed to two abstraction phenomena:

- *time scale abstraction*, and
- phenomena or *parameter abstraction*.

The time scale for the actual nonlinear behavior of the system may be much faster than the time scale at which system behavior needs to be analyzed. If system behavior were explicitly modeled at this small time scale, appropriately positioned small energy storage and dissipative effects have to be included in the system model. The ensuing time constants may obscure or complicate the generation of the more gross (or abstract) phenomena that are of interest, therefore, time scale abstraction techniques may need to be introduced to focus on the more useful behaviors. Furthermore, small time constants cause steep gradients and fast oscillations in system behavior, which results in numerical stiffness problems when conventional simulation methods

are used, or lead to the use of less accurate implicit integration methods to avoid stiffness problems [11, 15]. For example, consider a ball bouncing on a floor. When the ball on its downward trajectory hits the floor, elastic compression of the floor, or the ball, or both, enables storage of the ball's kinetic energy as internal compression energy which eventually builds up a negative force that imparts a negative velocity to the ball causing it to fly back upward. The velocity of the ball changes continuously, but starting from the point of impact for a short time interval, the velocity change has a very steep slope. This is shown in Fig. 13 for a system where the ball is considered to be an ideal rigid body and the floor is modeled to have a relatively large stiffness value. The effects of compression are indicated by the ball and floor displacement becoming negative for a very short period of time. When modeled in less detail, the velocity of the ball is instantaneously negated because the compression behavior is ignored. If the interaction between the ball and floor is modeled as a perfect *elastic* collision, the overall behavior implies that the ball continues to bounce but with decreasing amplitude because of air resistance. The simpler but correct behavior was achieved by abstracting away the stiffness effects causing an instantaneous change in velocity. Therefore, the abstract model combines continuous behavior with *abrupt* or discontinuous behavior changes. Since the elasticity of the ball or floor is not abstracted away but condensed into an instantaneous effect, this is an example of time scale abstraction.

A second cause for discontinuities in models can be attributed to component parameter abstractions. The effects of particular component characteristics, often parasitic terms, are simplified or ignored. However, this may reduce the degrees of behavioral freedom in the system by making energy storage elements (i.e., capacitors

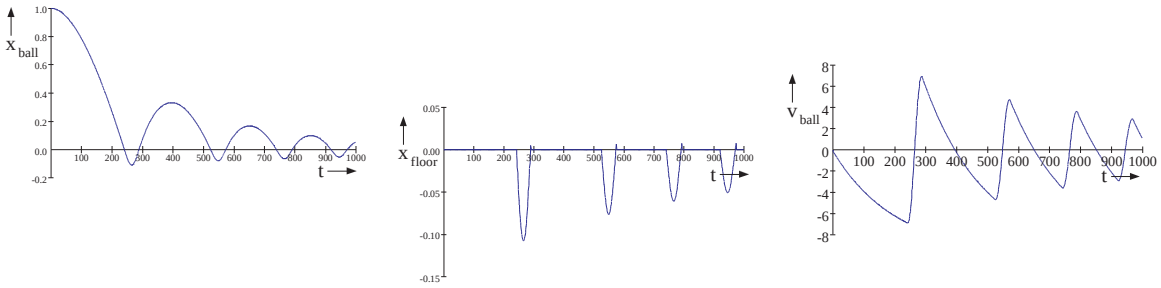


Figure 13: A perfect elastic collision between a rigid body and a very stiff floor.

and inductors in the bond graph framework) dependent. As a result, discontinuities can again become part of system behavior. For example, if the ball and floor elasticity in the bouncing ball example are abstracted away, the system shows a perfect *non-elastic* collision and the ball comes to an immediate stop at the point of impact. Because of the rigid body assumption, the kinetic energy of the ball dissipates instantaneously the moment its velocity is forced to be 0. In reality, small elastic coefficients with deformation effects allow the ball to maintain a velocity that is not directly coupled to the floor, and the behavior exhibits a steep gradient but remains continuous. This implies that the elasticity and deformation effects introduce additional degrees of freedom in the system model that result in continuous system behavior. On the other hand, if these effects are very small, and they have a negligible impact on the macroscopic behavior, they can be abstracted away in the system model. In that case, at the point of impact, the ball with non-zero velocity is directly coupled to the floor which has 0 velocity, and a discontinuity occurs. Note that this eliminates the ball momentum as a state variable.

Effects of the Lumped Parameter Assumption

To study the lumped parameter assumption in detail, consider the free expansion experiment conducted by Gay-Lussac and Joule, shown in Fig. 14 [35]. A chamber is made up of two connected bulbs with an idealized open-closed valve between them. Initially, only the left bulb contains a gas and the valve is closed. When the valve connecting the two volumes is opened, the gas in the left bulb expands freely and starts diffusing into the right bulb. Even if the connecting orifice is non-resistive, this diffusion introduces non-homogeneous turbulence effects that are active for a period of time. This is not an issue from a thermodynamics perspective, where the goal is to establish energy balance after the distribution has become homogeneous (e.g., determining the temperature of the water in the compartment surrounding the two volumes). Therefore, the lumped parameter assumption holds. Based on the underlying modeling assumption, the transitional effects are negligible to the time scale of interest. However, the lumped parameter assumption does not hold during discontinuous changes. In case of the free expansion experiment, if the valve were closed quickly enough after opening to operate as a sequence of two instantaneous changes, the gas could not have diffused yet. So, the homogeneous distribution of gas over the two volumes is never actually established. In fact, an immediate closing of the opening renders the intermediate opening mode *mythical*. Therefore, it leads to no redistribution of energy which conforms with the observation that in real time there never was a connection.

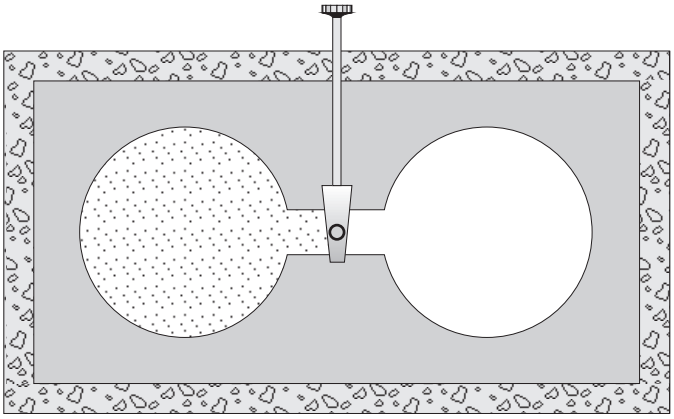


Figure 14: Free expansion of a gas by diffusion after an instantaneous change.

The Effects of Discontinuities

Discontinuities in physical system models have a number of effects that do not occur in the analysis of continuous system behavior:

- Energy storage elements may become dependent, thus changing the dimension of the state vector which may result in an apparent violation of conservation of energy.

- A discontinuous change may trigger a chain of discontinuous changes.

Conservation of Energy

The lumped parameter assumption requires dependent storage elements to be treated as one, but that may result in a discontinuous change of stored energy when a configuration change occurs, and may cause a Dirac pulse[1] generated at the instant the change occurs. The pulse represents an amount of energy that dissipates discontinuously as heat, *discontinuous dissipation*, much like the loss of energy due

[1]This is a pulse of finite area but infinitesimal width that occurs at a point in time.

to resistive dissipation which is modeled as a source of entropy. If the environment is assumed to be *isothermal*, this source is not modeled explicitly, otherwise it can be represented as a Dirac pulse. Note that this instantaneous loss of energy would actually occur over a short time interval if small dissipative elements capturing the energy redistribution effects in the connection between the storage elements were returned in the model.

Conservation of State

After a new mode of operation is inferred, its state vector has to be derived based on the state of the system in the previous mode. This is referred to as the *initial value problem* and if the system state is represented by energy stored by independent elements, which have an integrating relation, there are no discontinuous changes. Therefore, discontinuous state changes only occur if storage elements become dependent and their value is determined by other system elements. In this case, two situations characterize the initial value problem:

- Mode changes cause two or more storage elements to become dependent, and this causes the size of the state vector to change. As discussed, individual energy variable values change, but their total remains the same and conservation of state determines the new state values.

- Mode transitions cause dependency between sources and energy storage elements. In this case, the switching causes a source (i.e., the environment) to instantaneously transfer energy into or out of the system. The new values of energy stored by the elements involved is set to the source enforced values.

As a result, the system transfers from one mode to another, but the initial energy distribution in the new mode may be different from the distribution in the last real mode.

Invariance of State

The diode-inductor example in Chapter I illustrates that a discontinuous change may generate additional discontinuous changes that occur instantaneously. In general, discontinuous changes occur when signal values cross a threshold value during continuous evolution of system behavior [77, 78, 79]. The effect of this is that energy connections are activated or deactivated. This may cause adjoining signals to change discontinuously and cross switching thresholds themselves. The result is the activation or deactivation of one or more energy connections one after another, leading to a sequence of discontinuous changes. The modeling assumption is that discontinuous changes are instantaneous, therefore, real time does not progress during a sequence of discontinuities. Real time continues to evolve only after a model configuration is reached where no more switches occur. Sequences of instantaneous changes make it difficult to infer the new mode of continuous operation. In addition, the task of correctly advancing the system state across a series of model configuration changes, producing the correct mapping of the system from the last continuous mode onto the new one, where behavior again evolves in real time is nontrivial.

This is solved by observing that any system configuration that occurs during a sequence of switches has no real existence, and these system configurations are transitional, or *mythical*. A consequence of this is that the system cannot exchange energy with its environment during this period, in other words, it is *isolated*. Therefore,

there is no redistribution of stored energy within the system during this sequence. The principle that energy is not redistributed during discontinuous, instantaneous changes, but only after a new mode is reached is termed the principle of *invariance of state*.

Illustration

This section illustrates the previous concepts and notions by two examples.

Conservation of State and Energy

Consider an electrical circuit with two capacitors in parallel connected by an ideal switch (Fig. 15). When the switch is open, the two energy storage elements can charge and discharge independently, but they become dependent when the connecting switch is closed requiring the two elements to achieve a common potential. The total charge on the two capacitors before the switch was closed has to be preserved so that the physical principle of conservation of state (charge), is not violated. Assuming that the initial charge on C_1 is q_1, and there is no initial charge on C_2, the common potential after the switch is closed is $V^+ = \frac{q_1}{C_1+C_2}$. The amount of energy before closing the switch is $\frac{q_1^2}{2C_1}$. After closing the switch the charge on C_1 is $q_1^+ = \frac{C_1}{C_1+C_2}q_1$ and the charge on C_2 is $q_2^+ = \frac{C_2}{C_1+C_2}q_1$, therefore, the amount of energy in the system, $\frac{q_1^{+\,2}}{2C_1} + \frac{q_2^{+\,2}}{2C_2}$, is $\frac{q_1^2}{2(C_1+C_2)}$. This implies that closing the switch causes a loss of energy equal to

$$\frac{q_1^2}{2C_1} - \frac{q_1^2}{2(C_1+C_2)} = q_1^2 \frac{C_2}{2C_1(C_1+C_2)}. \tag{3}$$

Imposition of the conservation of charge principle appears to result in an instantaneous loss of energy in the system, i.e., the conservation of energy principle is violated.

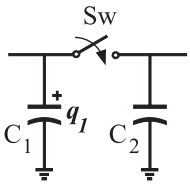


Figure 15: Two capacitors that become dependent by an ideal connection.

This loss occurs because of an arc across the switch when it closes, and could be explicitly modeled by a dissipative effect. However, since it is small and of a parasitic nature, it is abstracted away. In any event, in case the environment is not isothermal the loss has to be explicitly modeled

Mythical Modes and Invariance of State

To illustrate a sequence of discontinuous changes, consider the effect of a diode that operates in one of two possible modes in Fig. 16:

- As an effort source; it enforces 0 volts, independent of the current.
- As a flow source; it imposes a negative leakage current, independent of the voltage.

Initially, the voltage drop across the diode is 0 and it operates in its effort source mode (Fig. 17). When the switch is closed, this effort source enforces 0V on both of the capacitors which requires C_1 to discharge instantaneously. This results in a current flow that approaches negative infinity and based on the switching specification of the diode ($I_D \leq I_{leak}$) the model configuration changes immediately to one where the diode operates as a current source. Since no more discontinuous changes occur, the capacitors become dependent and redistribution of charge occurs as in the

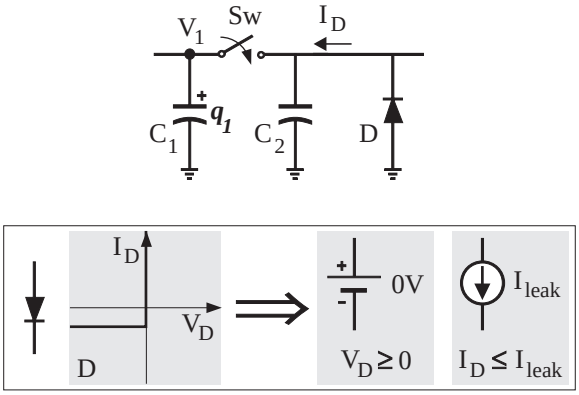


Figure 16: Two capacitors that become dependent and switch modes of operation of the diode.

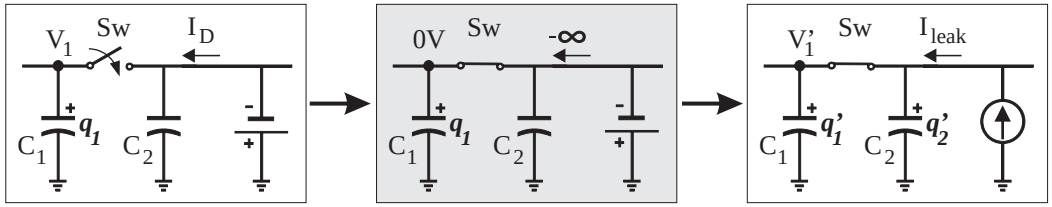


Figure 17: A series of discontinuous changes may contain mythical modes.

two capacitor system described above. Imposing the modeling assumption that discontinuous changes occur instantaneously, the model configuration where the diode operates as an effort source and the switch is closed is departed instantaneously and never achieved in reality. The infinite current is never actually established, it is only used to infer the new mode of continuous operation. If it were considered a real mode of operation, C_1 would discharge instantaneously during the intermediate mode of operation. Therefore, when the diode switches to its current source mode of operation no energy would be left to maintain the leakage current.

Limitations

The occurrence of a number of discontinuous changes may make energy storage elements alternate between mutual dependence and independence during a sequence of changes. From a physical perspective this situation has to be analyzed carefully. To illustrate a situation where this may lead to a conflicting system model, consider the electrical circuit in Fig. 18, where the relay turns off when the voltage drop across C_1 equals the voltage drop across C_2. If the relay was on initially, the moment the switch is closed this condition holds and the relay opens. Because of the instantaneous nature of discontinuities, the model configuration where there is a connection between both the capacitors is departed immediately. As discussed in terms of the free expansion experiment, even though dissipative effects of the connection are not explicitly modeled, energy redistribution still takes time. In real time, the basic model configuration does not change, i.e., the capacitors are disconnected all the time. However, the relay is open and its switching condition implies that the charge on both of the capacitors has been redistributed to reflect their equal voltage drop, which it has not. This indicates that analysis of continuous behavior with instantaneous junction switching does not generate consistent behavior in this scenario, and either resistive or inductive effects of the connection have to be included, or the modeling methodology has to be modified.

To summarize, switching conditions based on energy stored by elements that are alternating dependent and independent within one sequence of discontinuous change are inconsistent with lumped parameter assumptions, and, therefore, prohibited. In this case, either model refinement or another modeling approach has to be chosen. Notice that gradients based on these energy variables *can* be used as demonstrated

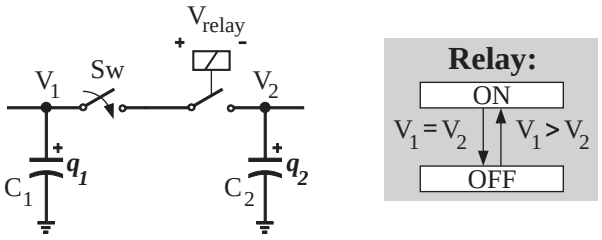


Figure 18: Instantaneous dependency changes between energy storage elements cause problems.

by the capacitor-diode example. Though there may not be an actual flow of current, a gradient exists the moment the switch closes, and this causes the diode to change to its mode of operation where it enforces a leakage current.

A Discrete Formalism

An idealized discrete switching element can impose a discontinuous binary, *on/off*, relation on energy transfer paths in the system to model configuration changes. Fig. 19 shows two connected tanks that get disconnected when a latch closes. If the latch closes, the flow through the connection becomes 0. Therefore, there is no energy transfer across it and the net result is that the two tanks become independent subsystems, and this illustrates a seamless implementation of the *mode-switching* process. The physical *on/off* state for a switch is affected by continuous variables which cause the switching when variable values cross prespecified thresholds (e.g., $p_1 > p_2$ in the above example) and control logic that governs the *on/off* relations is defined by combinational or sequential automata. The need for sequential control logic is demonstrated in Fig. 20. Initially the latch in the connecting pipe is in its upright position. When a threshold pressure difference is exceeded, the latch opens to the left or to the right. Once it has opened in one direction, it cannot open in the opposite

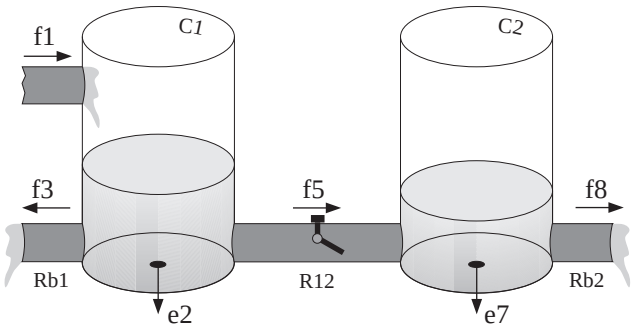


Figure 19: Mode switching of a bi-tank system by a latch.

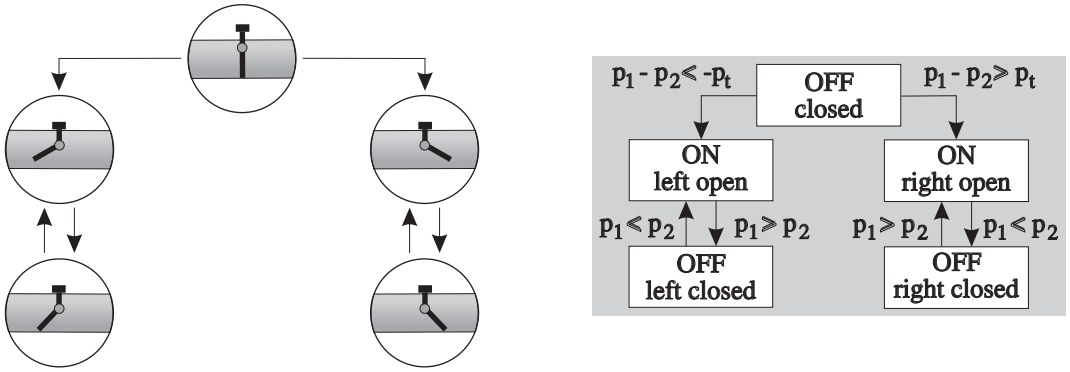


Figure 20: A physical system may contain discontinuities that have memory.

direction anymore, therefore, future *on/off* states are a function of the past state of the latch.

An important observation applies to the *state vector* of a system, which contains a necessary and sufficient number of variables to completely describe the system state. In continuous physical systems the set of variables that describe the energy distribution in the system capture its entire history, and, therefore, serve as the state vector for the system. However, when models contain discontinuities that are controlled by sequential logic, the *energy state* vector cannot completely specify system history. Future behavior becomes dependent on the internal states of the sequential automata, and a hybrid state vector is required to capture the necessary *logic states* as well. As

an illustration, consider the latched bi-tank system. If the internal state of the latch in Fig. 20 is not known, future behavior cannot be determined uniquely based on energy variable values alone. The same energy distribution can result in two different states for the latch, depending on its history, and system behavior can evolve along two different trajectories. To disambiguate this situation, the system state vector needs an additional logic component, which specifies the model configuration at different points in time.

Temporal Evolution in Behavior Transitions

The semantics of behavior generation for hybrid models need to combine real modes with instantaneous discrete behavior changes where real time does not advance. Note that a real mode of system behavior can encompass an *interval* where continuous evolution is specified, or a *point* in real time where a new energy state value may be specified by an algebraic relation. All discrete changes have to occur at well defined points in time. Consider the example of the perfect elastic collision of the bouncing ball shown in Fig. 21. Model configurations where the ball is moving freely (up or down) represent continuous modes of operation where system behavior evolves over time. The system model is abstracted so that the collision process is perfect and elastic, and holds only at a point in time at which the ball momentum is reversed. At this point in real time, no continuous evolution can be specified, if the ball and floor were in contact for any period longer than a point in time, the ball's momentum would transfer to the floor, and it would come to rest. On the other hand, if this real mode did not exist, i.e., the ball and floor never touched, the ball could not exchange momentum with the floor, which implies its velocity would never reverse.

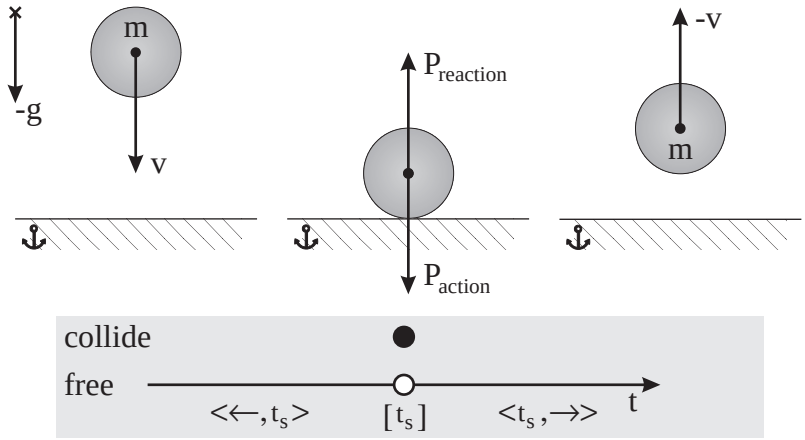


Figure 21: Real modes can have a point or interval presence in time and have to ensure time continuity.

The configuration where the ball is in contact with the floor is abstracted to a point, which is then followed by an interval of time where the ball travels upwards exhibiting continuous behavior.

In summary, this method for representing the bouncing ball behavior as a discontinuous change from a real mode (moving downward) to a second real mode (point of contact with the floor modeling the collision) and then a discontinuous change to a third real mode (moving upward) is much cleaner than a discontinuous model which represents the reversal in ball velocity as an initial value problem (e.g., [19, 57]). In the latter situation, the point in time at which the collision occurs is considered to be the start point of the second time interval with the ball moving freely upward, and the model specifies the initial velocity of the ball at the start point. This model implies the ball is always moving freely, up or down, reversing its velocity at a particular position with no explicit physical phenomenon, such as a collision, to account for the change. The net result is a model that violates the principle of invariance of state, because stored momentum changes abruptly without explicit interaction with

the environment.

Summary

This chapter categorically analyzes discontinuities in physical system behavior. These discontinuities can be the result from either time-scale abstractions or parameter abstractions, and result in mode switching behavior of the system. The principle of conservation of state governs transfer of system state between two modes of operation.

Principle 1 (Conservation of State) *Between two modes of operation, the total state in the system (i.e., charge, momentum, etc.) is conserved.*

Transfer of state between modes of operation may result in discontinuous dissipation. In case of an isothermal system this dissipation is not shown, otherwise it has to be modeled explicitly as a Dirac source of entropy, to ensure that the principle of conservation of energy is not violated.

Principle 2 (Conservation of Energy) *If the environment is not assumed isothermal discontinuous dissipation as a result of changes in the operational mode has to be represented by a Dirac source of entropy*

Another effect of discontinuities is that the system model may move through a series of mythical mode changes before a new real mode is achieved.

Definition 1 (Mythical Mode) *A mythical mode is a model configuration that has no representation in real time. Therefore, its state vector is not part of its operational domain.*

Because the state vector is beyond the operational domain of the model configuration, a mythical mode immediately generates a new configuration. Mythical modes do not have a physical meaning nor representation, and, therefore, these sequences of instantaneous change are modeling artifacts and could be removed in a model pre-processing step. Since a particular model configuration may be mythical or real, depending on the corresponding energy state, it is cost-effective to formulate model semantics that handle mythical modes correctly. These semantics are based on the principle of invariance of state, which states that mythical modes do not affect the mapping of the energy state between real modes.

Principle 3 (Invariance of State) *The energy state vector of a system is not affected by mythical modes.*

It is important to note that mythical modes do not affect the system state. This is a more precise statement than attempting to define mythical modes as states where no continuous behavior is defined. This is clearly illustrated in the bouncing ball example, where the point of impact represents a mode where only algebraic constraints on priori and posteriori values hold, and no continuous evolution in the form of differential equations can be defined. Though this mode is not continuous, it is still real.

CHAPTER IV

HYBRID BOND GRAPHS

Bond graphs use a small set of domain independent physical mechanisms to provide a powerful modeling formalism for complex physical systems [53, 104, 107]. These mechanisms and a junction structure inherently enforce energy conservation and power continuity constraints on system models to provide an elegant basis for analyzing the continuous behavior of physical systems. However, as shown in Chapters I and III, efficient modeling requires support for discrete, model configuration, changes. This chapter introduces a primitive switching element and control structure into the traditional bond graph formalism to allow for discrete changes in model configuration, and, therefore, discontinuous changes in system behavior.

Introduction

Typically, a system can be looked upon as a composition of n sub-systems, each of which can operate in k possible modes of operation. Overall, the system can operate in n^k behavioral modes. However, only a small number of these modes are actually achieved during normal operation of the system. In case of well known systems that operate in a limited number of modes, a global control structure can be pre-defined for determining system modes and the model associated with each of them [9, 18]. However, for lesser known systems, or for systems that will be operated in unknown ways, pre-enumeration becomes infeasible and brute force techniques cannot be applied because of combinatorial explosion, therefore, *compositional modeling* methods

are often applied to provide a solution [36, 95, 96]. These approaches generate models dynamically by composing model fragments.

Bond graphs provide a good framework for compositional modeling approaches [10] where the model for a mode of operation is generated by establishing or disconnecting energy connections between sets of bond graph elements at junctions. Such junctions are controlled locally using signal values tapped from the bond graph model. A controlled junction can be in one of two states, *on* and *off* which correspond, respectively, to the presence or absence of energy connections associated with this junction. Consequently, a system is modeled by first establishing a bond graph model of all possible components and their energy connections or interaction. Next, junctions are identified that turn *on* and *off* based on a local control mechanism. Once all control mechanisms are defined, valid models for each of the modes a system goes into can be generated dynamically from this overall energy model while components of the bond graph can be mapped back to components and mechanisms of the physical system.

Discontinuities in Bond Graphs

To extend bond graph modeling to hybrid physical systems without losing the inherent principles that govern physical system behavior and captured in bond graph models, several issues need to be taken into account.

- Interaction between the discontinuous part and the continuous part has to be consistent and the formalism specifying the discontinuous behavior has to be verifiably correct.

- In a discontinuous environment, continuity of power cannot be guaranteed,[1] however, conservation of energy during a sequence of discontinuous changes can and must be enforced. This means that the basic building block for ideal discontinuous configuration changes, modeled as switches, do not dissipate energy. When dissipative effects have to be modeled, resistive components have to be included in the system model.

- When discontinuous changes occur, causality may change. The formalism has to provide for a consistent, algorithmic, causality assignment scheme that holds in general.

- If a number of instantaneous discontinuous changes occur, the formalism has to ensure that this sequence ends in a valid mode of system operation with a correct state vector. The last mode in this sequence has a real manifestation, therefore, exchange of energy with the environment may occur causing discontinuous dissipation.

To model configuration changes in bond graphs, recently Broenink and Wijbrans have introduced *switching bonds* [18] and Strömberg, Top, and Söderman have applied an ideal *switch* [116, 117, 126]. Switching bonds are based on structural analysis of real-time systems, which can be modeled in terms of two components: a *data flow* and a *control flow* component [29, 46, 131, 132]. Lent conjectures that all systems can be described in this manner [128]. For switching bonds, the data flow part is represented by the bond graph formalism and the control flow part is represented by a *control box* that contains switching logic in the form of a global finite state automaton. Interaction

[1] Consider power supplied by a source that generates a true step.

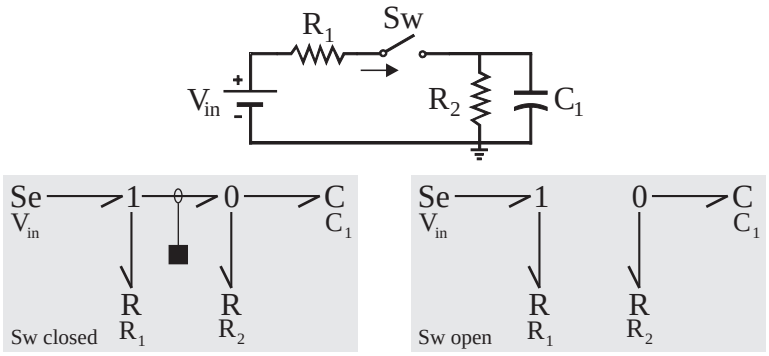


Figure 22: Switching bonds handle boundary conditions incorrectly.

occurs through switching bonds, i.e., an energy connection (or bond) is either present or not present. The interaction is depicted by a black box that is connected to a switching energy bond in the bond graph. Though the concept of separate data and control models has a lot of merit, interaction between the two through switching bonds causes problems. First of all, it may cause hanging junctions. More seriously, changing boundary conditions due to switching are incorrectly handled. For example, in Fig. 22 if the switch opens, the corresponding bond in the bond graph disappears. Though this disconnects the parallel part, the bond graph still shows a series connection (1-junction) that in reality is not present.

To eliminate these problems, Strömberg, Top, and Söderman introduced the *switch*, a new bond graph element, that enforced 0 effort or 0 flow on a junction to turn it on or off (Fig. 23). Notice that this is a degenerate form of the regular source elements, and its primitive *effort*/*flow* characteristic is commonly used in power electronics [55]. Because of the 0 effort or 0 flow, power (*effort*×*flow*) supplied by the switch is always 0, which conforms with the requirement that switching consumes no energy. However, the switch has some disadvantages:

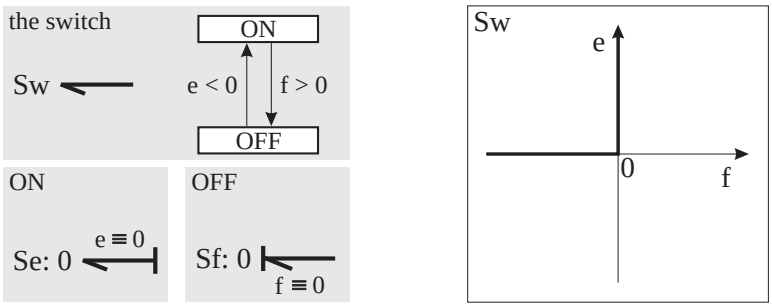


Figure 23: Operation of the switching element.

- Since switches are not energy elements, it is unnatural to represent them as bond graph elements.

- Switches are *transitional* elements that are only used to infer new modes of operation, consequently they represent *control aspects* rather than *physical concepts*.

- The use of switches obscures hierarchical structures. Typically when a system switches modes, a set of switches go from on to off or *vice versa*. Relations among switches are hard to identify and the use of bond graph elements clutters the model.

- The functionality of a switch is completely determined by the type of junction it is connected to. The same switch connected to a dual junction is *off* when *on* and *vice versa*.

Based on the aspects of switching bonds and the switch as bond graph element, Mosterman and Biswas have proposed the *controlled junction* [76]. The controlled junction recognizes the presence of data and control flow structures and the need for a separate formalism, along with bond graphs, to represent the control structure of a

physical system. The interaction between the control structure and the bond graph occurs by switching controlled junctions (rather than bonds), where the specification of controlled junctions adheres to the ideal specifications of the switch. Using finite state automata [1, 56] for the control part provides for a powerful hybrid formalism that is based on proven modeling methods. Interaction between the formalisms is consistent and rigorous which renders it verifiably correct.

The Modeling Language

Formally the hybrid bond graph approach combines continuous-time and discrete-time formalisms, which are modeled by traditional bond graphs [53, 104] and finite state automata [1], respectively. These two formalisms interact through controlled junctions which capture discontinuous variable changes. In the bond graph, controlled junctions have associated subscripts, e.g., 1_1, 0_2, to differentiate them from traditional junctions and also to provide a reference to their corresponding finite state automaton. The hybrid bond graph model of the latched bi-tank system (Fig. 19) is shown in Fig. 24. A controlled junction behaves like an idealized switch. A 0-junction that is *off* enforces 0 effort whereas a 1-junction that is *off* enforces 0 flow. When the junctions are *on*, they operate as regular junctions.[2] A change of state of a controlled junction may affect adjoining junctions and thus the causal relations in the graph. These new causal relations can be algorithmically derived using SCAP or its modified form MSCAP [129] in the new configuration.

In case of configuration changes, it becomes critical to establish correct loading on all adjoining bonds. Each power bond connected to a deactivated controlled junction

[2] A similar *timed junction* has been applied for illustrative purposes in [123].

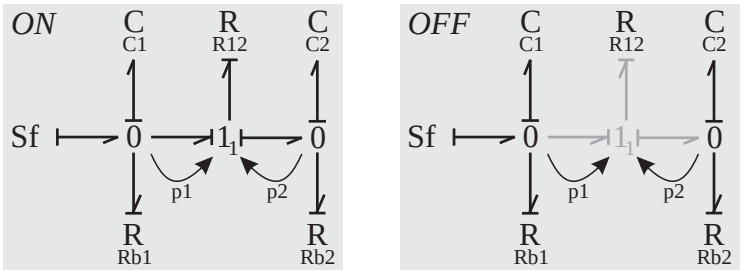


Figure 24: The controlled junction establishes mode-switching behavior.

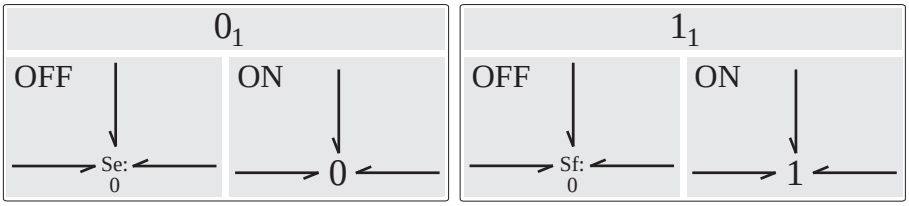


Figure 25: Operation of the controlled junction.

is loaded by a source element of 0 value to correctly handle boundary conditions of the disconnected model fragments (Fig. 25). In most bond graph models of physical systems, 0- and 1-junctions appear in alternating sequence, therefore, when a junction is deactivated, its adjoining power bonds can be removed from the bond graph to establish the new model configuration. Also, the effort causality as enforced by a deactivated 0-junction typically does not completely determine causality on all bonds of a neighboring 1-junction. Neither does flow causality of a deactivated 1-junction propagate directly across adjoining 0-junctions.

The subscripts of each controlled junction (e.g., 0_1 in Fig. 26) identifies its associated finite state automaton that determines whether it is in the *on* or *off* state. In other words, the finite state automaton defines a junction's control specification (CSPEC). The input of each CSPEC consists of

– power variables from the bond graph, and

– external control signals.

This input is depicted in the hybrid bond graph as arrows into the controlled junction. Mathematical operations may be applied to the power signals before they are input to the CSPEC. These operations can be modeled by a block diagram (as is used in the bond graph modeling tool CAMAS [16]) to manipulate signal values on active bonds. The output of the CSPEC sets the associated controlled junction to *on* or *off*. Internally the CSPEC can have any number of states, its control logic can be combinational or sequential. Conditions for a valid CSPEC are:

– Each internal state must map onto an *on* or *off* state of the controlled junction.

– Transition conditions on the edges have to evaluate to boolean values in each mode of operation.

– The CSPEC conditions have to result in at least one real mode of operation for all reachable energy distributions.

– The CSPEC conditions consist of values immediately before (priori) and after (posteriori) switching.

The final condition is an important characteristic of the CSPECs. The priori values are unchanged, invariant, during a sequence of instantaneous changes. Such a sequence is driven by the posteriori values, as these may differ for each newly inferred mode of operation. Therefore, priori values determine which modes of operation are required to have a physical manifestation (typically the result from time scale abstraction), whereas posteriori values determine sequences of modes that can be traversed to

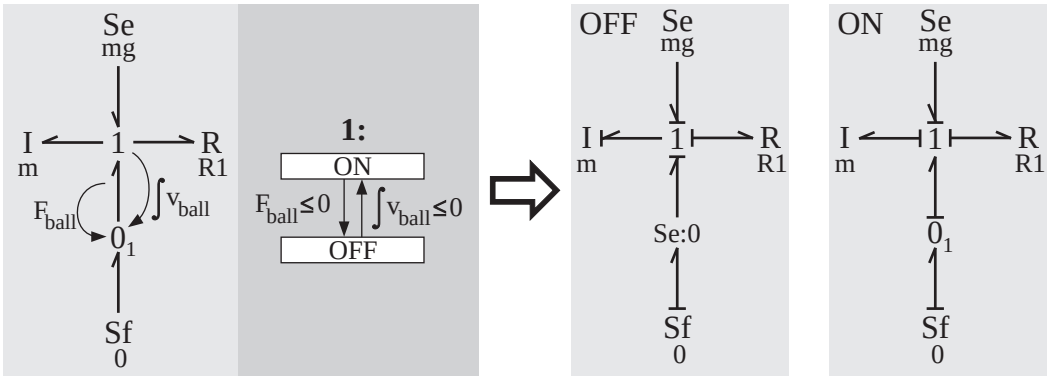


Figure 26: Hybrid bond graph model of an ideal non-elastic collision.

reach these modes (typically the result from parameter abstraction). Discontinuous changes in signal values are characterized by different priori and posteriori values.

The set of local control mechanisms associated with controlled junctions constitute the *control model* of the system. The control model performs no energy transfer, therefore, it is distinct from the bond graph model that deals with the dynamic behavior of the physical system variables. Control models describe the *transitional*, i.e., mode-switching behavior of the system. A *mode* of a system is determined by the combination of the *on/off states* of all the controlled junctions in its hybrid bond graph model. Note that the system modes and transitions are dynamically generated, and do not have to be pre-enumerated.

A Perfect Non-Elastic Collision

The use of controlled junctions for a perfect non-elastic collision between a ball and floor is illustrated in Fig. 26. The ball inertia is modeled as m, the air resistance as R_1, and gravity as an effort source, mg. In the *off* state, there is no connection between the ball and the floor and the bonds connecting the ball to the resulting effort

source with 0 value can be disregarded. The flow source with 0 value is disconnected as well. The CSPEC part of the model shows that at the point the ball touches the floor ($\int v_{ball} \leq 0$), the controlled junction turns *on* and the flow source with 0 value becomes connected to the mass. This implies that the ball velocity = 0, but the connection between the ball and floor source also causes a change of causality, forcing the inertial element to operate in derivative causality. The ball stays connected to the floor as long as the force it exerts is > 0. When this force becomes negative, the junction turns *off* again, and the ball inertia becomes independent. Its momentum is reversed, and the ball flies up. This example illustrates a seamless integration of multi-mode behaviors in one model based on a local switching mechanism. Other examples of hybrid bond graph models are discussed in [77, 78, 79, 85].

Mode Switching

As shown in Chapters I and III, a discontinuous change may propagate through a system causing other discontinuous changes to happen instantaneously. Model abstraction makes it hard to explain such a complex chain of events in incremental causal terms. Nishida and Doshita [98] address these problems by moving the system through a sequence of *mythical instants*. These mythical instants are arrived at and departed instantaneously, therefore, they are considered *transitional*. Sequences of mythical changes make the task of algorithmically inferring the eventual real mode that is attained and its correct state vector, a real challenge. Since the transitional modes are mythical and never achieved in reality, they do not affect the energy balance of the system. Therefore, the signal values in each mode in a sequence of instantaneous changes are calculated from the *original* energy distribution, i.e., the energy

distribution in the last *real* mode before the sequence of mythical changes occurs. This issue is of paramount importance in correctly inferring new modes of operation and the initial state vector in these modes.

The principle of invariance of state (Chapter III), i.e., repeatedly using the energy distribution of the previous real mode during mythical mode switches, governs the computation of the initial values of the state variables in the new real mode. This process is illustrated in Fig. 27. The real modes are depicted by a white background and the mythical modes are shown on a dark background. In the final stage, after a real mode α_m is reached, the energy distribution of the previous real mode, as represented by their energy state variables p and q, is mapped onto the new mode. Physically, and in real time, the system has moved from mode α_k into mode α_m instantaneously at time t_s. The Mythical Mode Algorithm (MMA) is formally presented as Algorithm 1. In this algorithm, (E, F) represent the set of effort and flow variables, and (P, Q) represent the set of energy variables (generalized momentum and generalized displacement) in the system at switching time t_s. The set $(P_{\alpha_k}, Q_{\alpha_k})$ and corresponding $(E_{\alpha_k}, F_{\alpha_k})$ are *priori* switching values, whereas (P^+, Q^+) and (E^+, F^+) are *posteriori* switching values.

Algorithm 1 Mythical Mode Algorithm

Calculate the energy values $(Q_{\alpha_k}, P_{\alpha_k})$ and signal values $(E_{\alpha_k}, F_{\alpha_k})$ for bond graph model α_k at time t_s.
Use *CSPEC* to infer a possible new mode given $(E_{\alpha_k}, F_{\alpha_k})$ and $(E^+, F^+) = (E_{\alpha_k}, F_{\alpha_k})$.
while one or more controlled junctions switch state **do**
  Derive the new bond graph, α_{k+i}
  Propagate causality.
  Calculate the energy values (P^+, Q^+) and signal values (E^+, F^+) for the new mode, α_{k+i}, based on the initial values $(Q_{\alpha_k}, P_{\alpha_k})$.
  Use *CSPEC* again to infer a possible new mode based on $(E_{\alpha_k}, F_{\alpha_k})$ and (E^+, F^+).
end while
Establish the mode, α_m, as the new real system configuration at the point of discontinuity.
Update $(Q_{\alpha_k}, P_{\alpha_k})$ to the energy distribution for α_m, $(Q_{\alpha_m}, P_{\alpha_m})$.

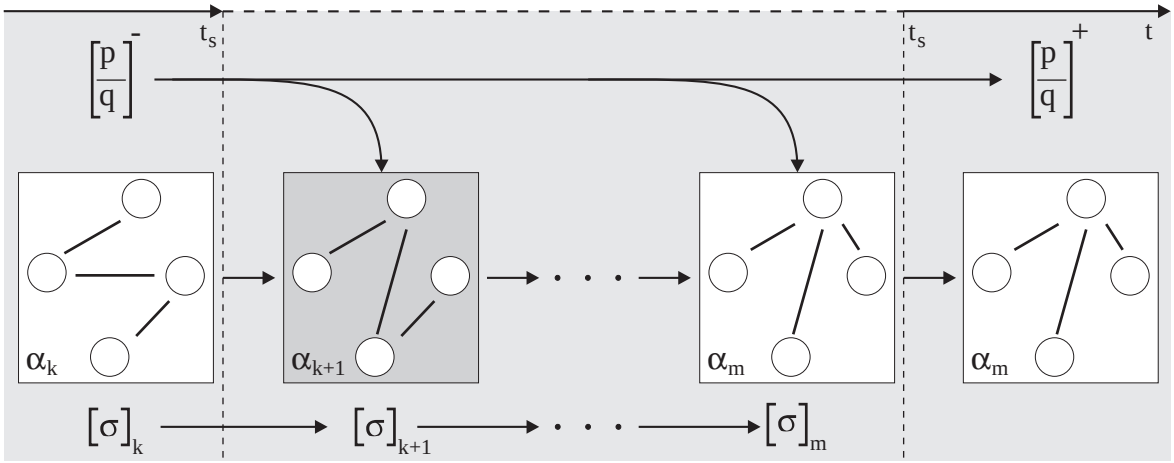


Figure 27: During mythical changes the discrete state vector changes to reflect configuration changes. The energy state vector is only changed once a real mode is reached.

Transfer of Energy State

To derive the new state vector in a new mode, it is observed that in a hybrid bond graph discontinuous state changes only occur if storage elements become dependent (Chapter III). One of two cases may occur [79]:

1. one or more storage element may become dependent on a source, or

2. two or more storage elements may become dependent on each other and δ-sources become active, and the state vector between the two configurations is different.

In the first case, the posterior energy stored in the dependent elements, p_i^+, is determined by the value of the source, u,

$$p_i^+ = r_{S,i} C_i u, \tag{4}$$

where $r_{S,i}$ is a gain factor associated with the route from the source to the dependent element, i, and C_i is the dependent element. In the second case, Dirac pulses, δ, are

induced by explicitly modeled sources or dependent storage elements that enforce a discontinuous change of the independent, integrated, state variable, p_0^+. The area of such a pulse combined with the gain factor from its origin to the independent storage element specifies a change of p_0^+. The cumulative change is given by the general formula:

$$p_0^+ = p_0 + \sum_{storage,i} a_{\delta,i} r_{i,0} + \sum_{sources,j} a_{\delta,j} r_{j,0}. \tag{5}$$

The area $a_{\delta,j}$ is the explicitly modeled interaction with the environment, and the area $a_{\delta,i}$ can be calculated as

$$a_{\delta,i} = p_i^+ - p_i, \tag{6}$$

which is the loss of generalized charge or momentum in the dependent storage elements. The new signals generated by dependent states, $\frac{p_i^+}{C_i}$, are forced to values determined by the new signal from the independent storage element, $\frac{p_0^+}{C_0}$, and the gain factor. This is described by

$$p_i^+ = r_{0,i} \frac{C_i}{C_0} p_0^+ \tag{7}$$

which, along with Eq. (5) and Eq. (6), can be applied to determine the new value of the independent state variable, p_0^+.

In the special case that no explicitly modeled δ-sources become active, conservation of state holds because the amount of generalized charge and momentum added to the independent storage element equals the loss by each of the dependent storage elements combined. Therefore, the total amount of charge and momentum in both modes remains the same. For n dependent storage elements, element 0 is assigned to be in integral causality and the new value of its stored energy, p_0^+, is determined by

$$p_0^+ = p_0 + \sum_{i=1}^{n-1} (p_i^+ - p_i) r_{i,0} \tag{8}$$

This can be expressed in terms of the value of the independent storage element, p_0^+, by substituting Eq. (7)

$$p_0^+ = p_0 + \sum_{i=1}^{n-1}(r_{0,i}\frac{C_i}{C_0}p_0^+ - p_i)r_{i,0} \tag{9}$$

or [85],

$$p_0^+(1 - \sum_{i=1}^{n-1} r_{i,0}r_{0,i}\frac{C_i}{C_0}) = p_0 - \sum_{i=1}^{n-1} r_{i,0}p_i \tag{10}$$

where $r_{i,0}$ is the gain factor associated with the route from storage element i to element 0, and C_i is the parameter value of storage element i. Note that this may result in loss of energy to the environment [87].

Implementation

The behavior generation algorithm has three key modules:

1. *ESPEC*, the energy model of the system, specified by a bond graph,
2. *CSPEC*, the information model of the system specified by finite state automata, and
3. *MMA*, the mythical mode algorithm that controls interaction between *ESPEC* and *CSPEC*.

The *MMA* was implemented under Microsoft Windows using Visual Basic 3.0 Professional Edition [24]. The continuous model is incorporated as a system of explicit difference equations which are derived from the bond graph model manually. Note that the derivation process for system equations is already fully automated in systems like CAMAS [16]. Integration is implemented as a forward Euler approximation and

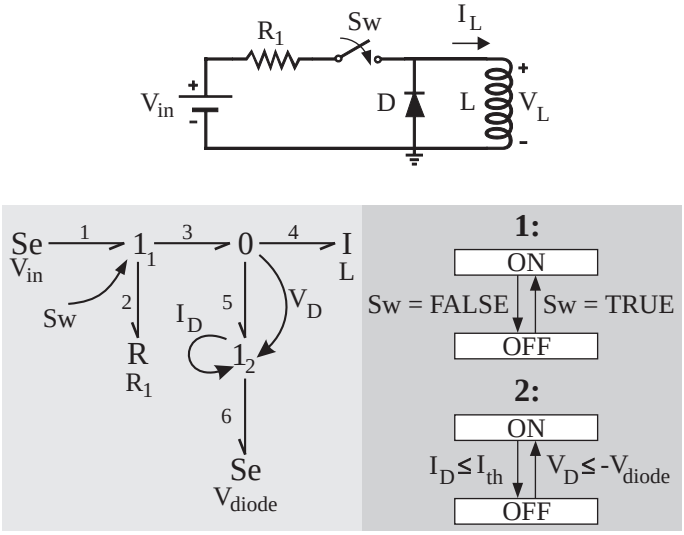


Figure 28: A diode-inductor circuit which causes a mythical mode.

careful selection of the time step produced good results in spite of this simplifying approximation. Each CSPEC is implemented as an IF-THEN-ELSE statement.

The Diode-Inductor Circuit

An example implementation of the diode-inductor circuit (Fig. 28) discussed previously in Chapter I is illustrated. First, 0-junctions are associated with all common voltage nodes. In the particular circuit there is one node (apart from ground) with three branches. The inductor, L, can be directly connected to the 0-junction. The remaining branch that consists of a series or common flow connection, 1-junction, connects R_1 and V_{in}. Because of the switch, this is a controlled junction, 1_1, that turns *on* and *off* based on an external control signal as specified by CSPEC **1**. Finally, the diode branch can turn *on* and *off* as well, based on signal values in the circuit specified by CSPEC **2**. When *on*, the diode enforces a constant voltage, V_{diode}, and this is represented by a voltage source.

This circuit was also described by Lorenz [63], but the *MMA* methodology is more formal and systematic. The CSPEC definitions for the model are local, therefore, no global control structure needs to be known beforehand. Given an initial state, all system modes that are reachable from this state will be generated dynamically by simulation. The four possible modes of the system are:

$$\gamma : \begin{cases} \alpha_{00} \rightarrow \text{switch open} \wedge \text{diode not conducting} \\ \alpha_{01} \rightarrow \text{switch open} \wedge \text{diode conducting} \\ \alpha_{10} \rightarrow \text{switch closed} \wedge \text{diode not conducting} \\ \alpha_{11} \rightarrow \text{switch closed} \wedge \text{diode conducting} \end{cases} \tag{11}$$

CSPEC and *MMA* are applied to effect mode switching in the system. Initially the switch is open, the inductor has no stored energy, the diode is not conducting, and the system is in mode α_{00} (Fig. 29). At time step 10 the switch is closed, the system moves into the mode α_{10}, and all effort and flow values are recalculated for the new configuration. No further mode transitions occur, and the inductor charges as shown. At time step 100, the switch is reopened and the *MMA* recomputes all efforts and flows for the new mode α_{00}. The inductor becomes dependent on a 0 value flow source which forces its flux to 0 (Eq. (4)). Because it had built up a flux (i.e., energy p_0), disconnecting it induces a large negative voltage ($-\infty$ in the limit). This causes the diode to come on instantaneously, so mode α_{00} becomes mythical, and the system switches to mode α_{01}, where the effort and flow values are recomputed based on the initial flux of the inductor. Again the new values do not cause another mode change so *ESPEC* is active and Fig. 29 shows that the inductor discharges through the diode. Note that the signal values computed for the mythical mode are only used for switching (with no loss of energy), therefore, the infinite negative voltage

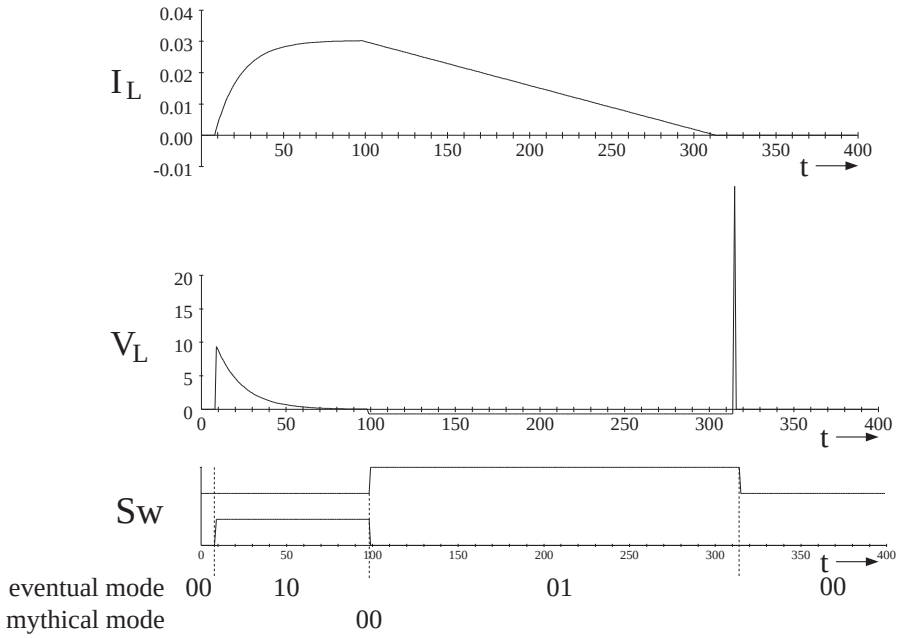


Figure 29: Simulation result of the diode-inductor circuit with parameter values $V_{in} = 10V, R_1 = 330\Omega, L = 5mH$.

is never reached. If it were, the stored energy of the inductor would be released instantaneously, producing an incorrect energy balance in the overall system.

With time, the flow (current) through the inductor decreases to zero. At time stamp 315 the current value is ≤ 0. Therefore, the current through the diode is ≥ 0 (opposite sign) and this causes a final transition: The system again switches to mode α_{00}, and since there is no stored energy in the system, this becomes the final state. The spike observed in simulation is an artifact caused by the time step used for simulation. In this simulation, between the two time steps the current went from a small positive to a small negative value before the transition took place. Thus when the system transitioned to the α_{00} mode the small current in the inductor went to 0 instantaneously, which resulted in the spike shown.

Divergence of Time

Consider a scenario where the diode requires a threshold current $I_{th} > 0$ to maintain its *on* state. If the inductor has built up a positive flux, the diode comes on when the switch opens. However, if the flux in the inductor is too low to maintain the threshold current, the diode goes off instantaneously, but in its *off* state the voltage drop exceeds the threshold voltage again. The model goes into a loop of instantaneous changes (see Fig. 4) and real time halts.

In general, if switching specifications are such that at any point in a discontinuous sequence of switches the system comes back to an already generated discontinuous configuration, a loop of discontinuous changes ensues. This implies that system behavior stops progressing or diverging in real time, which is obviously in conflict with physical reality, and *divergence of time* constitutes an important condition for verifying a system model for physical consistency.

Principle 4 (Divergence of Time) *Model configuration changes have to terminate in a real mode of operation.*

Chapter V presents a verification methodology based on divergence of time by applying a multiple energy phase space analysis in detail. Discussion on the divergence of time principle also appears in other work [47, 79].

Summary

Hybrid bond graphs incorporate a combination of continuous energy modeling and multiple discontinuous behavior modes to define a comprehensive modeling methodology for physical systems. The systematic and uniform mode-switching method adds

to the formal compositional modeling properties already associated with traditional bond graphs. The consistent, rigorous and complete modeling language developed combines:

- the bond graph scheme to model the energy-related aspects of system behavior, and
- finite-state automata to model signal-flows that cause configuration changes in the bond graph model to produce discontinuous behavior.

Interaction between the two components of the model are restricted to signals that act on so-called *controlled junctions*. These signals are an integral part of the bond graph language.

The strict definition of the interaction between the energy-flow and signal-flow components of the modeling methodology is of paramount importance in generating valid physical models. The approach presented supports modeling discontinuities caused by (1) abrupt switching, such as in idealized valves and diodes, (2) mode switching caused by parameter value changes, such as the change from laminar to turbulent flow in a pipe when the Reynolds number goes above a threshold value, and (3) configuration switching caused by changes in sub-system models.

The advantages of the described method are:

1. focusing on the energy model instead of the external control models (as is done in QSIM and CC) allows for dynamic model composition,
2. the inherent integrity checks enforce physically correct models during continuous operation, and

3. the physically consistent interaction between the energy and logic model components allows for verification of mode switching behavior.

Dynamic model composition is extremely important for complex systems that include a large number of discontinuous components. As discussed earlier, such systems can exhibit an exponential number of modes, therefore, pre-enumerating modes is not a feasible approach to building system models. Moreover, a large number of these modes are not physically achievable, but that cannot always be determined before hand. The inherent integrity checks ensure the physical correctness of a model and aid the modeler in building a correct model. Furthermore, the conservation and continuity constraints help reduce the number of spurious behaviors that are generated. Overall, the four principles outlined in Chapters III and IV along with the hybrid bond graph methodology provides a powerful scheme for modeling complex physical systems at various levels of abstraction, while ensuring the physical correctness of the models generated.

CHAPTER V

ENERGY PHASE SPACE ANALYSIS FOR MODEL VERIFICATION

This chapter first shows how priori switching values are used to model algebraic constraints. Next, it develops a multiple energy phase space analysis technique for analyzing CSPECs, detecting physical inconsistencies, and then using priori switching values to correct them. This approach is based on invariance of state (Chapter III) which provides for a switching invariant that can be exploited in analysis of switching behavior [79].

Combined Perfect Elastic and Non-Elastic Collision

As an example, first consider a perfect elastic collision (Fig. 30). When the ball hits the floor with a velocity v, it creates an impulse which results in a reaction impulse by the floor. In case of a perfect elastic collision, this causes the ball to instantaneously reverse its velocity and start traveling upwards. In the corresponding hybrid bond graph model, this is modeled as a modulated flow source that becomes active when the controlled junction 1 (0_1) comes on. The CSPEC indicates that this happens when the position of the ball reaches the floor. Collision behavior is best described algebraically, i.e., $v^+ = -v$, as opposed to continuous dynamic operations defined by a set of differential equations. Therefore, the configuration only holds at a point in time, instead of an interval. This is achieved by using the momentum of the ball prior to switching as the condition to turn the flow source off ($p_{ball} > 0$). Upon impact, it is first inferred whether this is the new real model configuration and if so, switching

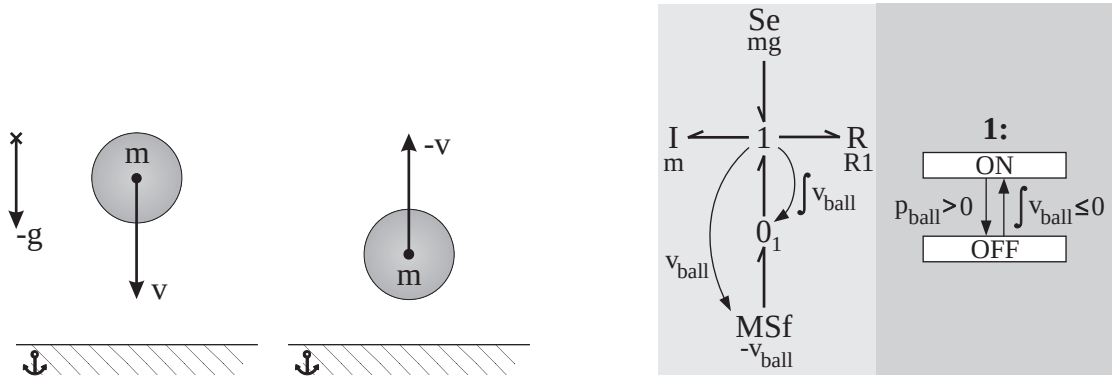


Figure 30: A perfect elastic collision.

ends and the priori switching values are updated. In this case, the new priori values immediately invoke a configuration switch, resulting in a model where behavior is governed by a set of differential equations. The convention throughout this thesis is that these priori values are expressed as energy variables (generalized momentum p, and generalized charge q) and the posteriori values are described by power variables (flow f, and effort, e), here v and F. The configuration switch infers that the ball moves upward as a result of reaction force created by the algebraic constraints, and the controlled junction turns off and the ball is represented as a separate model fragment again. Notice that the amount of energy returned by the floor, i.e., the coefficient of restitution [13], is a modeling decision since the floor is not part of the system but of its context.

Now, consider a combination of a perfect elastic and a perfect non-elastic collision model. This results in a ball bouncing with decreasing amplitude because of air resistance, and after a certain amount of time, when the ball hits the floor with momentum less than a pre-set threshold, the perfect non-elastic collision[1] mode is

[1]This models the physical notion that the restitution coefficient depends on the impact velocities [13].

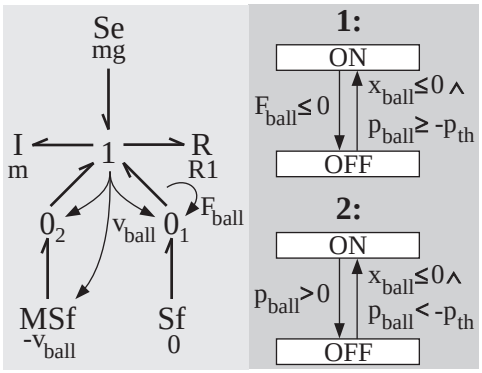


Figure 31: A combined perfect elastic and non-elastic collision.

activated, and the ball comes to rest on the floor (Fig. 31).

To analyze the physical consistency of this model, all switching conditions in the CSPECs have to be expressed in terms of stored energy prior to switching, which is represented as a two dimensional space with axes p_{ball} for the momentum of the ball, and x_{ball} for the position of the ball. All CSPEC transitions are already specified in terms of priori switching values, except for the force F_{ball} which, from the bond graph, is computed as $F_{ball} = -F_g + F_m + F_{R1}$, where g represents gravity. F_{ball} is the force of the ball on the floor and when its value is negative, the ball disconnects from the floor. To derive the conditions under which the controlled junction 1 turns *off* in a sequence of switches, F_{ball} has to be expressed in terms of the priori values for x_{ball} and p_{ball}. The storage element dependency that arises when the flow source becomes active, causes F_m to have a derivative relation, i.e.,

$$F_m = \frac{dp}{dt}. \tag{12}$$

From the CSPEC for controlled junction 1, the condition for switching is $F_{ball} \leq 0$. When this junction is in its *off* state, $p_{ball} = mv_{ball}$. When it switches *on*, the velocity and momentum of the ball become 0 instantaneously. Based on the constituent

relation in Eq. (12), this causes a Dirac pulse on F_m which approaches $\pm\infty$ depending on whether the stored momentum was negative or positive. If the momentum was 0, F_m equals 0. Let the function $sign$ be defined as

$$sign(x) = \begin{cases} -1 & \text{if } x < 0 \\ 0 & \text{if } x = 0 \\ 1 & \text{if } x > 0 \end{cases} \tag{13}$$

The condition for switching becomes $F_{ball} = -F_g - sign(p)\delta + F_{R1} \leq 0$. Because of the magnitude of the Dirac pulse, the effect of the gravitational force and the air resistance can be neglected at switching if $p \neq 0$. With the minus sign compensated, the condition for the controlled junction 1 to transition immediately from the *on* to *off* state is $sign(p)\delta \geq 0$. This inequality holds for all values of $p > 0$. If $p = 0$ then $F_m = 0$ and $F_{R1} = 0$, so the switching condition becomes $-F_g \leq 0$. Because of the negative value of F_g (the gravitational force acts downward), this condition is never satisfied and no further switching occurs. Consequently, the area for which $p > 0$ (modes 01 and 11) is grayed out in the phase space in Fig. 32.

The two switches instantaneously affect variables that are used in transition conditions of their CSPECs so they have to be analyzed for consistency of their combined effect. To this end, phase spaces are established for each of the four modes of the combined elastic and non-elastic collision and labeled 00, 01, 10, and 11, where the left digit indicates whether the controlled junction 2 is *on* (1) or *off* (0), and the right digit indicates the same for controlled junction 1. The energy phase spaces for each of these modes of operation are shown in Fig. 32. The areas that are instantaneously departed are grayed out and the conjunction of the four energy phase spaces (Fig. 33) shows that there is an energy distribution which does not correspond to a real mode

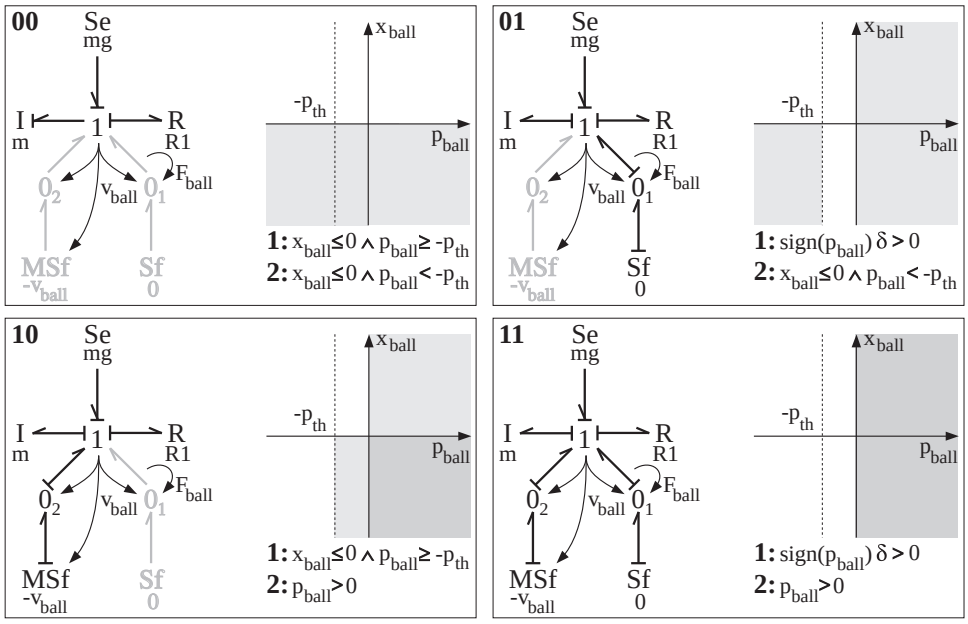


Figure 32: Energy phase spaces for each of the modes of operation of a combined perfect elastic and non-elastic collision.

of operation. Since the dimensions of the energy phase space are invariant across switches, this energy distribution cannot reach a real mode of operation during a sequence of switches, thus violating the divergence of time condition.

In this part of the phase space, when the ball hits the floor, it has positive momentum. For the bouncing ball, this mode is unreachable, and, therefore, the model is consistent. The system always moves towards a negative momentum and it instantaneously reverses when the displacement becomes zero (Fig. 33). So, analytically, the displacement never becomes negative. However, due to numerical disturbances, or initial conditions, the model may arrive in the physically inconsistent area of operation, especially in case the floor is another moving body. Therefore, in such situations or when simulating the system, the CSPEC conditions are not sufficiently constrained to avoid meaningless physical behaviors.

From the physical system it is clear that additional constraints can be imposed

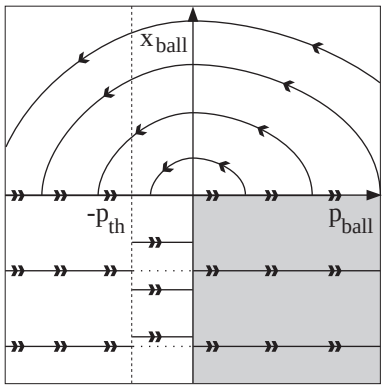


Figure 33: Conjunction of the multiple energy phase spaces for each mode of operation of a combined perfect elastic and non-elastic collision.

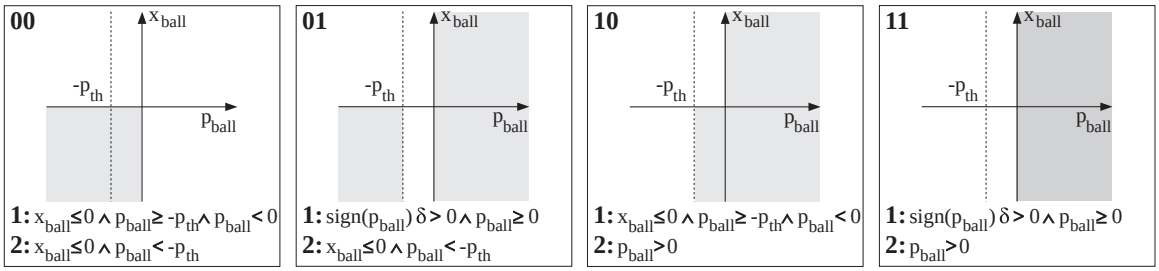


Figure 34: Multiple energy phase space analysis of the modified model shows there is a real mode of operation for each energy distribution.

based on the momentum of the ball. Since the switching conditions of CSPEC **1** are not mutually exclusive, the conditions $p_{ball} < 0$ and $p_{ball} \geq 0$ can be added to the *off/on* and *on/off* transitions, respectively. This results in the energy phase spaces shown in Fig. 34, and the conjunction of the energy phase spaces now has a real mode of operation for each energy distribution. Because of the combinatorial switching logic, this real mode of operation is reachable in one switching step. A simulation of the physically consistent system is shown in Fig. 35. The air resistance (R_1), causes the amplitude of the ball to decrease until the momentum of the ball falls below the threshold value p_{th} and the ball comes to rest on the floor.

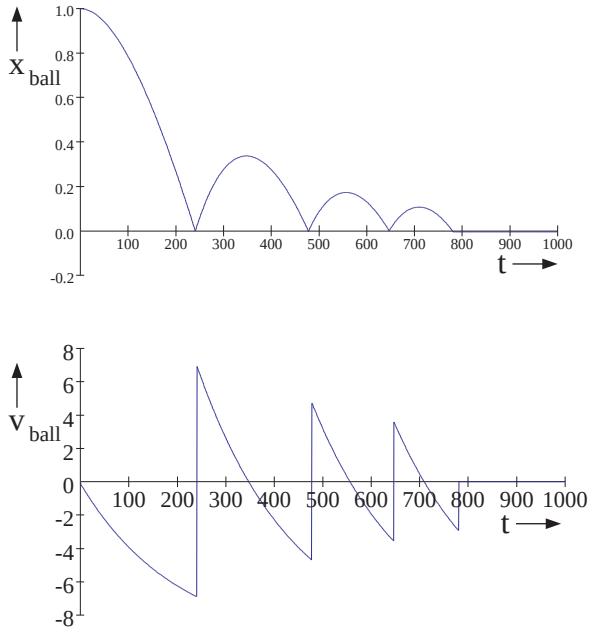


Figure 35: A bouncing ball which comes to rest when its impact velocity is below a specific threshold value.

Managing Complexity

The multiple energy phase space analysis technique requires combined analysis of all of the individual phase spaces for each permutation of the switches, and this is exponential in the number of switches. To prevent this combinatorial explosion, the physical characteristics of energy storage elements can be utilized to partition the system. If k sub-systems can be identified with l interacting local switches, the energy phase space complexity becomes $k2^l$, which is much smaller than $2^{k \cdot l}$. Typically, good system decompositions should result in a small[2] number of switches per sub-system, therefore, the analysis becomes more manageable. If decomposition is unable to bring down the computational complexity to reasonable levels, the model stiffness can be reduced by introducing additional storage elements (i.e., reintroducing some parasitic

[2]Otherwise, the system is very stiff and dynamic effects can be collapsed into fewer storage elements, e.g., this is the case for automatic transmissions in automobiles [68].

effects).

To determine the set of locally interacting switches, the input and output signals of each switch are established. All switches (i.e., controlled junctions) whose associated signals are used as CSPEC input to other switches are considered dependent and part of the same sub-system, thus creating their own *local* mode. Signal value changes at a switch may eventually affect signal values over the entire bond graph, but energy storage elements, because of their integrating effects, prevent *instantaneous* changes across them. Therefore, there is a time lag before input signal changes can cause output signal changes for these elements and their downstream elements. Therefore, switching effects only propagate along power bonds till an independent energy storage element is reached.

The delay characteristic of energy storage elements is only valid for elements that are independent in all switch configurations. When elements become dependent, i.e., their behaviors are defined by derivative causality, signals propagate across them instantaneously. Dependency between storage elements in the bond graph is checked by identifying causal boundaries. In the bond graph, each 0-junction with an incident S_e, R or C can be considered a *sink of flow causality*, and each 1-junction with an incident S_f, R or I is a *sink of effort causality* (Fig. 36). These constructs prevent flow and effort causality from propagating. This property can be exploited to partition the bond graph into instantaneous effort and flow areas with their corresponding C and I elements. If each effort and flow area contains only one of each type of storage element, this element is always independent. Therefore, it can be used as a border of instantaneous propagation, i.e., no instantaneous propagation is transmitted through such elements.

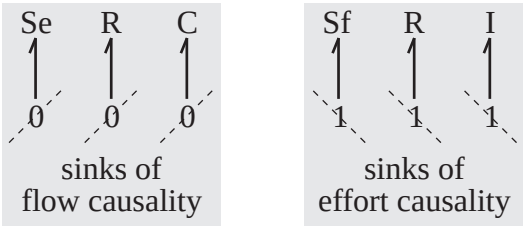


Figure 36: Sinks of effort and flow causality partition the instantaneously affected parts of a bond graph.

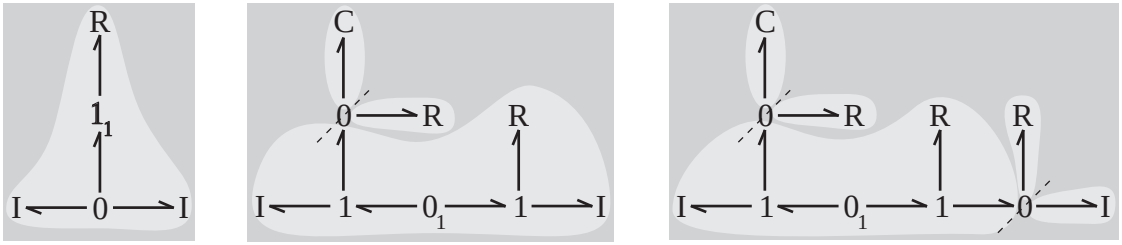


Figure 37: Multiple I elements in one flow area may become dependent.

Fig. 37 shows several examples of the application of the partitioning method. In the left-hand graph, even though the controlled junction is *on* and the two I elements are currently independent, they can become dependent, and, therefore, defined to be part of one flow area for creating sub-systems. Similarly, in the middle graph, the I elements are independent when the controlled junction is *off*, however, the partitioning analysis again determines that they may become dependent when the junction comes *on*, therefore, they are assigned to the same flow area. In the hybrid bond graph model on the right in Fig. 37, the I elements are separated by a sink of flow causality, and, therefore, they are always independent.

Once flow areas and partitions are established, the dimensions of the energy phase space for each of the local modes are derived from the input signals to the CSPECs. All variables of energy storage elements that affect a particular signal which is used as

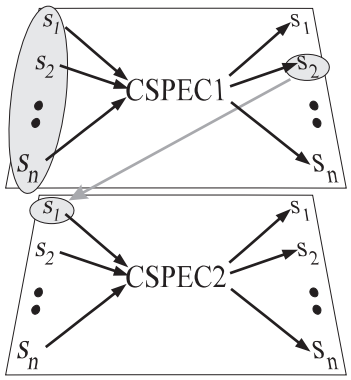


Figure 38: The two steps to establish interacting switches, local modes, and the dimensions of their energy phase space.

a CSPEC transition condition are to be included as a dimension. In addition, input signals (i.e., variables enforced by sources) are included as phase space dimensions.

Finding locally connected switches and the dimensions of the energy phase space is a two step process (Fig. 38):

- For each switch, find all other switches that have input signals affected by its output.
- For each set of interacting switches, compile the energy variables that affect their input signals.

The Internal Combustion Engine

To illustrate how instantaneous changes propagate through parts of a system, a model of a spark-ignition internal combustion engine is considered. Special attention is directed toward the valve- and ignition-timing sub-system.

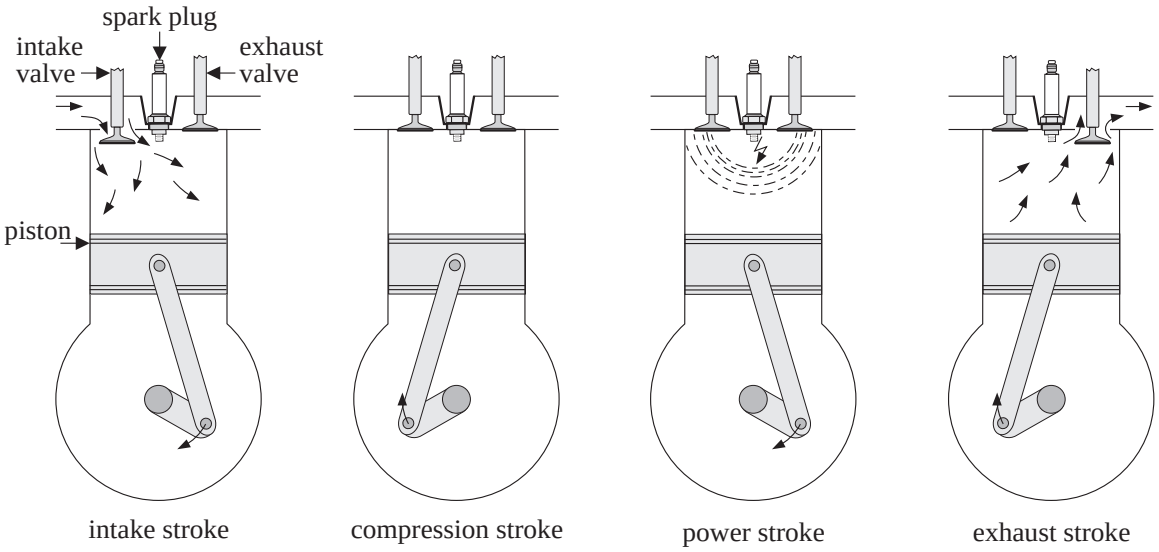


Figure 39: The four strokes of a spark-ignition internal combustion engine.

The Working of a Four-Stroke Engine

The four stroke internal combustion engine operates in four distinct modes (Fig. 39) [100]. During its *intake* stroke, the intake valve is opened and as the piston moves down, the air/fuel mixture is drawn into the cylinder. After the piston reaches *bottom dead center* (BDC, i.e., its lowest point), the intake valve closes and the mixture is compressed as the piston moves up. Ideally at *top dead center* (TDC), ignition occurs by inducing a high voltage on the spark plug. During ignition, the chemical energy contained by the mixture is released in the form of heat. This causes a dramatic rise of temperature and pressure which is transformed into work during the *power* stroke, during which the piston moves down. At BDC the exhaust valve opens and during the *exhaust* stroke the exhaust fumes that remain after ignition are expunged from the cylinder. At the end of this stroke, the piston is at TDC and ready to start a new intake stroke. Note that in reality valve- and ignition-timing deviates from the ideal situation to optimize engine performance under real conditions.

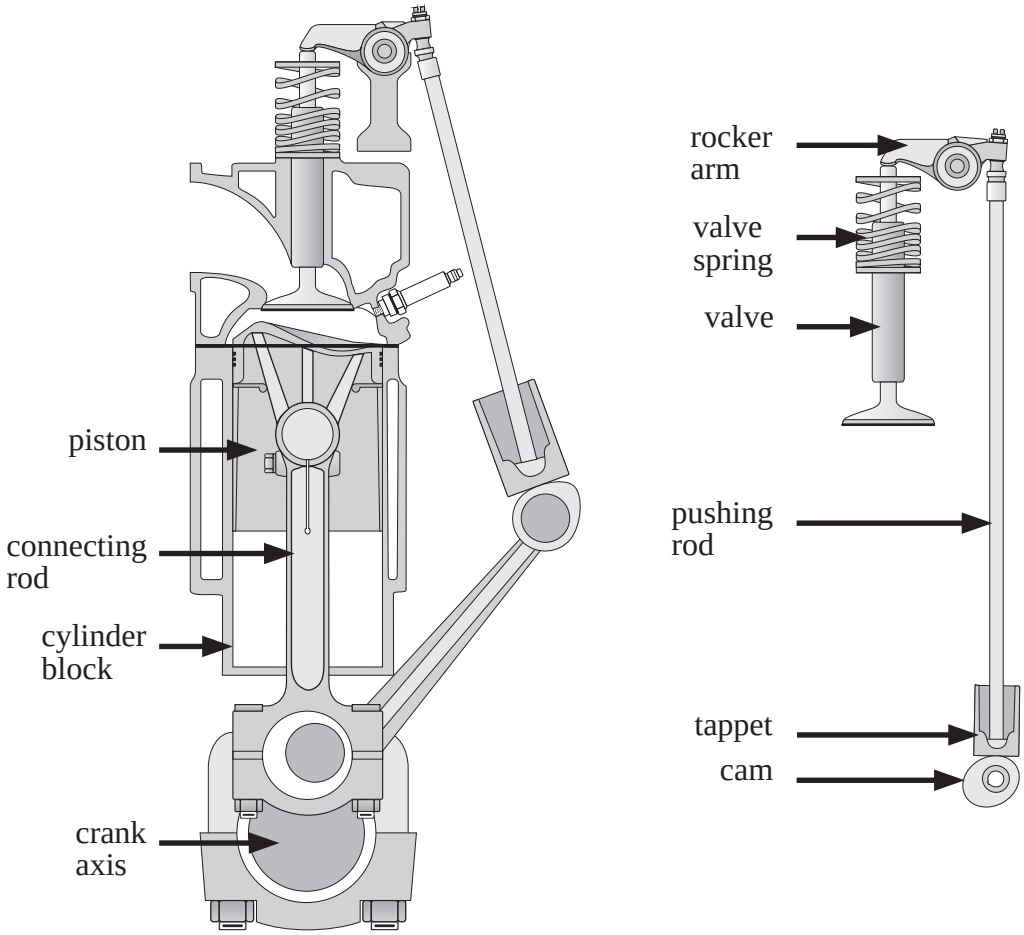


Figure 40: A spark-ignition internal combustion engine.

The Valve-Timing Sub-System

The valve- and ignition-timing mechanism is controlled by a *cam-axis*. For the four-stroke engine each valve opens once every two revolutions of the crank-axis, and the cam-axis rotates at half the speed of the crank-axis (Fig. 40). Timing is established by a number of *cams*, a mechanism to open and close valves, on the cam-axis. At desired times, a cam opens a valve by displacing the valve tappet. This tappet connects to the valve's rocker arm by a pushing rod. A valve spring ensures that the valve closes as the cam displacement returns to its original position (its base).

The cam-follower mechanism shown separately, is modeled as a combined perfect

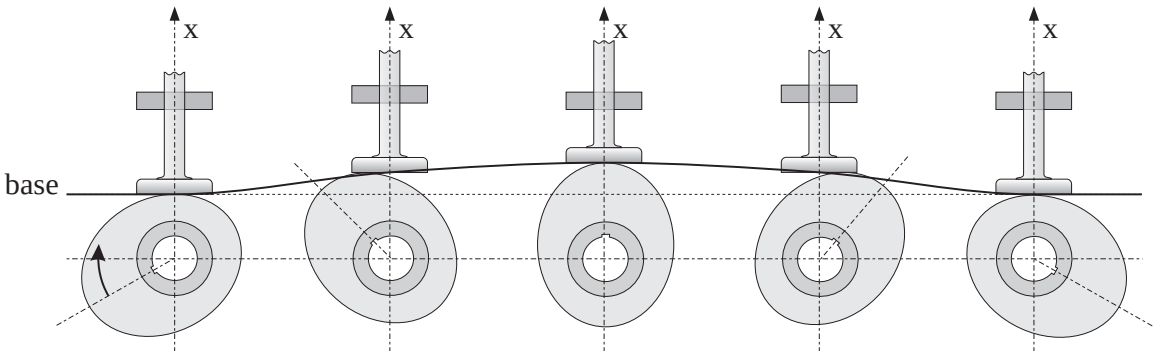


Figure 41: Cam mechanism which opens a valve during one revolution.

elastic and non-elastic collision between the tappet of the pushing rod and the rotating cam. As the cam rotates, the rod is lifted by an amount that is proportional to the rotational angle of the cam (Fig. 41) [45]. If this rotation exerts sufficient force, the inertia of the valve mechanism may cause the tappet to lift off the cam only to bounce back at a later time. If the momentum of the valve mechanism is higher than a threshold value, this collision is considered perfect elastic, otherwise it is perfect non-elastic.

The hybrid bond graph model of this system (Fig. 42) is very similar to the combined perfect elastic and non-elastic model of the bouncing ball. However, in this situation, the value of the displacement, x_{th}, that causes contact between the cam and the rod-tappet is variable and the momentum at collision becomes $p_{col} = m(v_{cam} - v_{rod})$. Furthermore, the value of the modulated flow source differs, as derived by

$$v_{cam}^{+} - v_{rod}^{+} = -\epsilon(v_{cam} - v_{rod}). \tag{14}$$

Given $\epsilon = 1$ and $v_{cam}^{+} = v_{cam}$ (the assumption is that this velocity is not affected by the collision), this yields

$$v_{rod}^{+} = 2v_{cam} - v_{rod}. \tag{15}$$

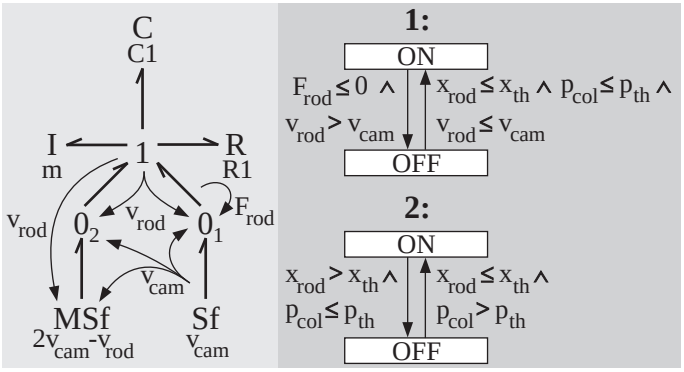


Figure 42: Hybrid bond graph model of cam-follower system.

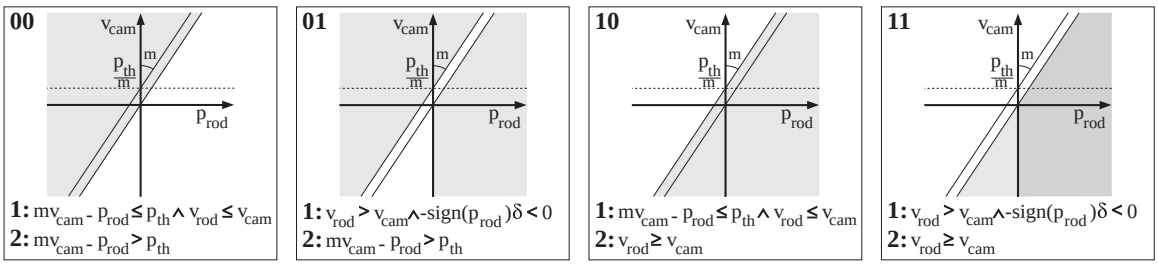


Figure 43: Energy phase space analysis of the cam-follower system.

The energy phase space analysis of the hybrid model in Fig. 42 is shown in Fig. 43 to be physically consistent, and simulation of one cam revolution is shown in Fig. 44. Notice that one of the phase space dimensions represents a source variable. Because sequences of switches occur instantaneously, external variables cannot change during such a sequence. Therefore, external variables are invariant across a sequence of switches as well, and consequently they can be used in the energy phase space analysis. The dimension of the displacement variable x_{rod} is omitted for clarity reasons.

System Partitioning

A bond graph model of the internal combustion engine is shown in Fig. 45. On the top-right, the *intake* and *exhaust valve* mechanisms are shown. The elastic collision

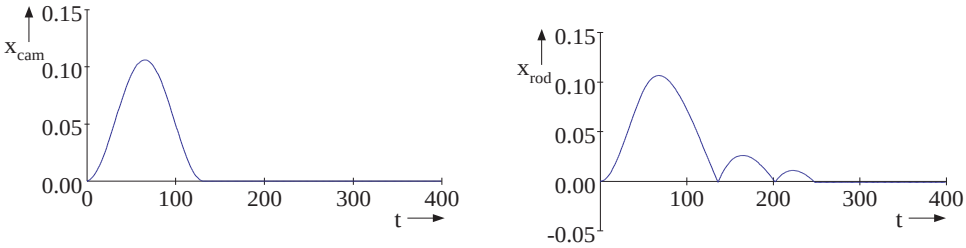


Figure 44: Simulation of the cam-follower system.

effect is modeled as a modulated flow source and the non-elastic collision enforces the velocity of the *cam-axis*. So, the opening and closing of the valves is controlled by the cam-axis, which also controls the ignition by driving a *rotor* and a *breaker*. The rotor mechanism selects the cylinder that requires ignition and the breaker induces the high voltage required by the spark plug. A *gear train* rotates the cam axis at half the speed of the crank-axis. The *cylinder*, on the left, is a control volume that exerts a force on the piston which is translated into a torque on the crank-axis by the *slider-crank mechanism*. The flywheel, which has a high mass, smoothens torque fluctuations. The connected flow source sets the motor to operate at a desired velocity. The cylinder loses heat to the *environment* by conduction through the cylinder wall. Also, heat is lost by convection [121, 122, 124, 125] when the exhaust fumes are blown out of the cylinder during the exhaust stroke. This energy convection is depicted as a flow source modulated by the flow of matter out of the cylinder (Q_{ex}). Correspondingly, there is convection of heat on the intake stroke, which also carries chemical energy. Both of these effects are modeled as flow sources modulated by the flow of matter into the cylinder (Q_{in} and U_c). The *intake* and *exhaust* of the cylinder are controlled by the intake and exhaust valve mechanisms and the ignition is controlled by the breaker-rotor system.

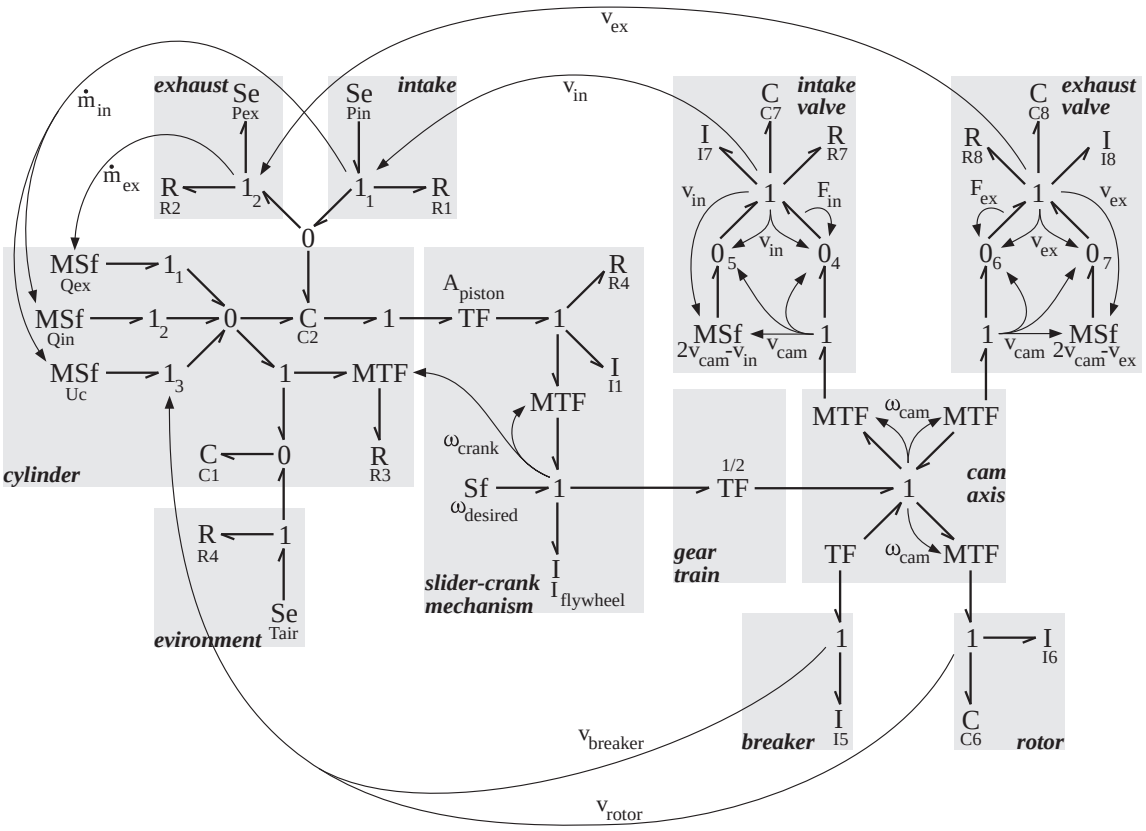


Figure 45: Energy part of a hybrid bond graph model of a four stroke spark-ignition internal combustion engine.

To find interacting discontinuities, the controlled junctions of the intake valve mechanism can be considered. First, notice that there are no sinks of flow causality in the crank-cam system, other than the crank velocity flow source and the modulated flow source that models the perfect elastic collision. Therefore, all inertias are either mutually independent, or they are dependent of the crank velocity flow source. So, when controlled junction 4 switches *on*, it propagates a flow (crank velocity) upward, and consequently an effort (force, torque) downward. This effort instantaneously propagates through the cam-axis and gear train to the crank-axis where it feeds into the system environment (where it may simply be dissipated by brakes). Because the motor is kept at constant *rpm*, this source is a sink of effort causality and instantaneous changes of other variables do not occur. In the opposite direction, switching of controlled junction 4 makes the valve inertia dependent, and, therefore, its velocity (v_{in}) changes instantaneously, which is used as input to controlled junction 1 by an instantaneous signal connection. The effects of this junction are fed into the cylinder control volume which is of an integrating nature so instantaneous propagation halts. Also, the abrupt change in v_{in} caused by switching controlled junction 4 affects controlled junction 5. So, these three junctions (1, 4, and 5) are interacting and their combination of *on*/*off* states form a local mode. The energy phase space analysis now has to include all permutations of this local mode ($2^3 = 8$). Analogously, the number of switch permutations for the exhaust valve is 8, which leaves just one more local mode in terms of controlled junction 3, which regulates ignition. The total number of switch permutations that have to be considered are, therefore, 17, which is far less than the number of permutations of the global mode of this model ($2^7 = 128$).

System partitions are determined by the degree of stiffness introduced in the

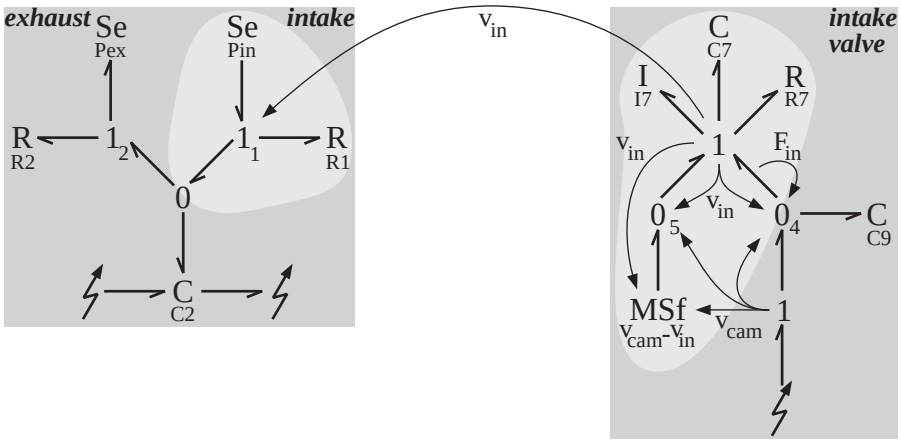


Figure 46: Instantaneous flow areas for the intake valve mechanism when rod flexibility is modeled.

model. If a system is modeled to be very stiff, instantaneous changes propagate through the entire model. Suppose the motor is not kept at constant *rpm* by an external source. Now, any of the inertias that are part of the valve- and ignition-timing sub-system become dependent of the flywheel inertia. These dependent storage elements then reduce to one substitute, I_s, with stored momentum p_s. This causes the velocity of the entire timing sub-system to change instantaneously and all controlled junctions affect each other.

Alternatively, if the valve rod is modeled to be flexible, the valve inertia does not become dependent (on a source or other inertia elements) and it always enforces valve velocity. In this situation, the one I in the valve model fragment only becomes dependent on the elastic collision flow source, and, therefore, all controlled junctions that take the velocity as their input have to be analyzed jointly. This again results in the local mode consisting of controlled junctions 1, 4, and 5.

As demonstrated by the example, physical systems provide for natural boundaries of instantaneous changes. This phenomenon is exploited by the multiple energy

phase space analysis method by partitioning the system into areas of instantaneous propagation, resulting in local modes. This reduces the complexity of the analysis from 2^n in case of n controlled junctions to $k2^l$ in case of k local modes each with l controlled junctions.

CHAPTER VI

HYBRID DYNAMIC SYSTEMS

This chapter develops the mathematical formalisms for hybrid dynamic system models. Model semantics, like before, are based on the principles that govern physical system behavior. The components for an implementation model for hybrid dynamic systems are presented and applied to the liquid sodium cooling system described in Chapter I.

Introduction

A hybrid system combines discrete switching patterns with continuous behavior, and, therefore, operates on a domain with discrete, $\alpha \in \aleph$, and continuous, $t \in \Re$, dimensions (Fig. 47). Behavior in this space is specified by piecewise continuous intervals, $x_\alpha(t)$, a function of both α and t. A hybrid *dynamic* system models dynamic physical system behavior. This behavior has to evolve over time, have and established direction of flow, and necessarily covers the complete interval on the time-line for which it is specified, therefore, the piecewise continuous intervals in temporal behavior evolution are adjacent to each other with *no gaps* (Fig. 47). Behavior in the piecewise continuous intervals is represented by well behaved, continuous functions f, called *fields*, which may be linear or nonlinear, and often are defined by a set of nonlinear ordinary differential equations [44]. An instance of temporal behavior in a field is called a *flow*, $\mathcal{F}$. Switching from one flow to another occurs at well-defined points in time when system variable values reach or exceed prespecified threshold values.

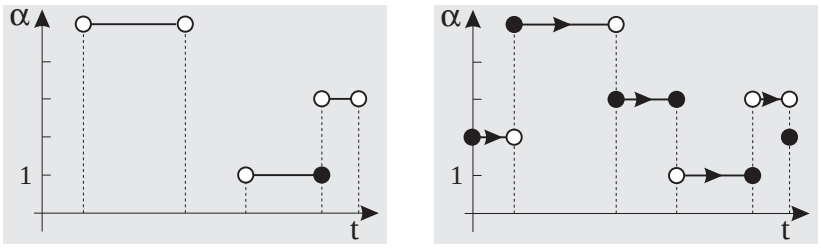


Figure 47: A hybrid system (left) and a hybrid dynamic system (right).

This defines an *interval-point* paradigm where flows are piecewise continuous and any discontinuous changes that occur have to be *simple* [108], i.e., limit values exist at points of discrete switching. An example of such a discontinuity is shown in Fig. 48, where there are two flows $\mathcal{F}_{\alpha_1}$ and $\mathcal{F}_{\alpha_3}$ that are C^2 (i.e., they are two times differentiable) on their respective domains, V_{α_1} and V_{α_3}. A point of discontinuity, $\mathcal{P}_{\alpha_2}$, on V_{α_2} occurs at t_s. Behavior at this point may be determined by an algebraic relation instead of a field.

In summary, a hybrid dynamic system consist of three distinct subdomains (Fig. 48):

- A *continuous* domain, T, with time, t, as a special continuous variable.
- A *piecewise continuous* domain, V, that specifies variable flow, $x(t)$, uniquely on the time-line.
- A *discrete* domain, I, that captures the operative piecewise continuous domain, V_α.

Definitions

Let I be a discrete indexing set and $\mathcal{F}_\alpha$, $\alpha \in I$, be a continuous, C^2, flow on a possibly open subset V_α of $\Re^n$, called a *chart* (Fig 49).[1] The sub-domain of V_α where

[1]Definitions used in traditional hybrid system modeling [44] are followed but modified in parts to fit the physical system modeling paradigm.

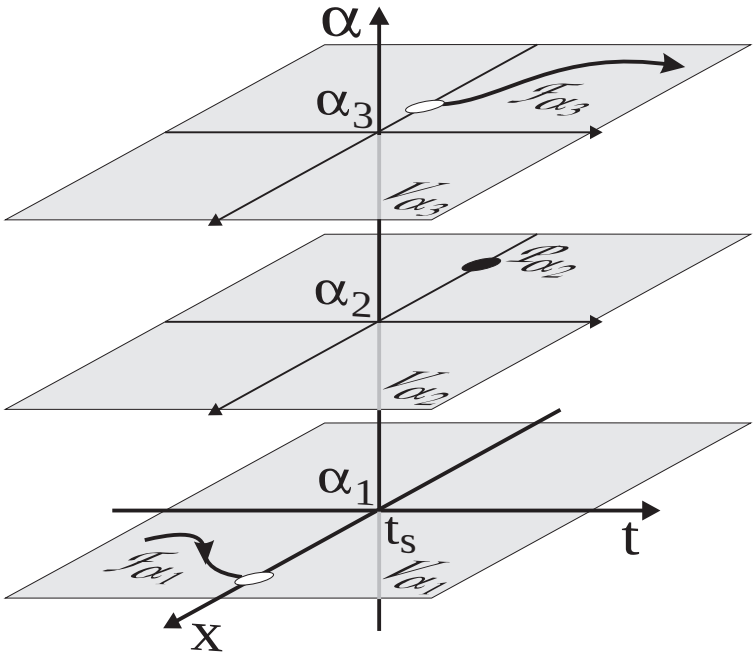


Figure 48: A hybrid dynamic system is piecewise continuous and operates on three sub-domains.

a continuous flow in time occurs is called a *patch*, $U_\alpha \subset V_\alpha$. The flows constitute the piecewise continuous part of the hybrid system. An explicitly defined isolated point that does not embody continuous behavior is called a *pinnacle*, $\mathcal{P}_\alpha$. The discrete switching function γ_α^β is defined as a *threshold function* on V_α. If $\gamma_\alpha^\beta \leq 0$ then the system transitions from chart V_α to V_β, defined by the mapping $g_\alpha^\beta : V_\alpha \rightarrow V_\beta$. The piecewise continuous level curves $\gamma_\alpha^\beta = 0$ are denoted as S_α^β, and define patch boundaries. If a flow $\mathcal{F}_\alpha$ includes the level curve, S_α^β, it contains the *boundary point*, $\mathcal{B}_\alpha$ (Fig 49). In summary, a hybrid dynamic system is defined by the 4-tuple[2]

$$H =< V_\alpha, \mathcal{F}_\alpha, \gamma_\alpha^\beta, g_\alpha^\beta > . \tag{16}$$

Points within the system are specified by $x_\alpha(t)$, a location in chart α at time t. *Trajectories* in the system start at an initial point $x_{\alpha_1}(t)$ and if $\gamma_{\alpha_1}^{\alpha_2} > 0$, $\forall \alpha_2$, the point

[2]Guckenheimer and Johnson refer to the respective parts as $< V_\alpha, \mathcal{F}_\alpha, h_\alpha^\beta, \mathcal{T}_\alpha^\beta >$ [44].

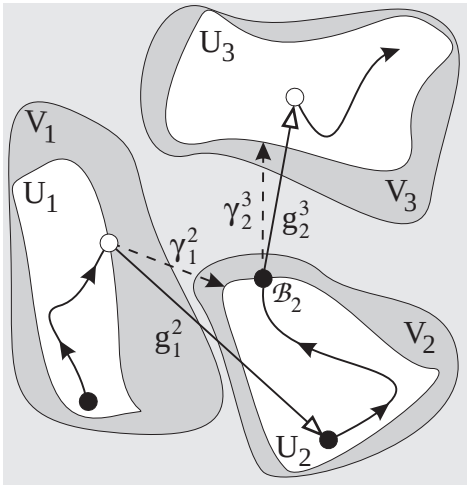


Figure 49: A planar hybrid system.

flows in V_{α_1} as specified by $\mathcal{F}_{\alpha_1}$ until the minimal time t_s at which $\gamma_{\alpha_1}^{\alpha_2}(x_{\alpha_1}(t)) = 0$ for some α_2. Taking $x_{\alpha_1}(t_s^-) = \lim_{t \uparrow t_s} \mathcal{F}_{\alpha_1}(t)$ the transformation $g_{\alpha_1}^{\alpha_2}$ takes the trajectory from $x_{\alpha_1}(t_s^-) \in V_{\alpha_1}$ to $x_{\alpha_2}(t_s) \in V_{\alpha_2}$. The point $x_{\alpha_2}(t_s) = (g_{\alpha_1}^{\alpha_2}(x_{\alpha_1}(t_s^-))$ is regarded as a new initial point and in case the new point is a pinnacle, it immediately invokes $\gamma_{\alpha_2}^{\alpha_3}$ which transfers the trajectory to $g_{\alpha_2}^{\alpha_3}(x_{\alpha_2}(t_s)) \in V_{\alpha_3}$. Continuity of time requires that the basic trajectory of a hybrid system evolves in a manner that intervals and points have to alternate, and, therefore, this is a point on a flow $\mathcal{F}_{\alpha_3}$. Let $x_{\alpha_3}(t_s^+) = \lim_{t \downarrow t_s} \mathcal{F}_{\alpha_3}(t)$, then $x_{\alpha_3}(t_s^+) = g_{\alpha_2}^{\alpha_3}(x_{\alpha_2}(t_s))$ and is regarded the initial point in V_{α_3}, and behavior evolution in time continues along V_{α_3}.

Approaches to Hybrid Modeling

Henzinger, Alur, and Nicollin, propose an approach which is based on the observation that continuous systems are best modeled by global sets of differential equations [2, 97]. A switching mechanism selects the set of differential equations whose assumptions are satisfied by the state of the system. This switching mechanism is

implemented as a temporal state machine. Because of the continuous-time character of physical systems, this temporal state machine relies on *dense time*, i.e., transitions occur at real-valued points in time instead of discrete (integer) time points [4]. This framework has been employed to prove reachability and divergence of time characteristics in the limited case where the physical systems being modeled only undergo constant rate of change in their variable values [48].[3]

A well known characteristic of hybrid systems is the possibility of a number of discrete changes occurring before a new patch is arrived at, where again a flow defined by a field governs system behavior [2, 44, 77, 98]. This situation occurs if $\gamma_{\alpha_k}^{\alpha_{k+1}}$ transports a trajectory to $V_{\alpha_{k+1}}$, and the initial point is transported by $g_{\alpha_k}^{\alpha_{k+1}}$ to a value that results in $\gamma_{\alpha_{k+1}}^{\alpha_{k+2}} \leq 0$, i.e., $g_{\alpha_k}^{\alpha_{k+1}}(x_{\alpha_k}) \notin U_{\alpha_{k+1}}$, and another domain $V_{\alpha_{k+2}}$ is instantaneously arrived at. These immediate transitions continue till a domain V_{α_m} is arrived at where the initial point is within U_{α_m}. To deal with these sequences of transitions, Alur *et al.* [2, 3], Guckenheimer and Johnson [44], and Deshpande and Varaiya [31], propose model semantics based on temporal sequences of abutting intervals [44]

$$\begin{array}{ccccccccccccc} \overbrace{}^{V_{\alpha_0}} & & & & \overbrace{}^{V_{\alpha_1}} & & & & & & \overbrace{\phantom{x_m^+ \mapsto x_{m+1}}}^{V_{\alpha_m}} & & \\ x_0 & \mapsto & x_1 & \overset{g_{\alpha_0}^{\alpha_1}}{\hookrightarrow} & x_1^+ & \mapsto & x_2 & \overset{g_{\alpha_1}^{\alpha_2}}{\hookrightarrow} \cdots \overset{g_{\alpha_{m-1}}^{\alpha_m}}{\hookrightarrow} & x_m^+ & \mapsto & x_{m+1} & . \\ [t_0 & & t_1] & & [t_1 & & t_2] & & & [t_m & & t_{m+1}] & \end{array} \tag{17}$$

Since these intervals overlap in time, a trajectory may be in several locations at a point in time, t_s. Therefore, these points in time are complemented with an index that specifies their order of transition. During a series of discrete switches, $(t_s, i), (t_s, i+$

[3]These systems are far more restrictive than linear systems in the classical systems sense, and only in approximation present in physical reality. Moreover, it requires precise control to achieve, as is elegantly shown by a water clock believed to be built by Ktesibios in Alexandria in the third century B.C.

$1) \ldots (t_s, n)$ the trajectory moves between these ordered points in time, repeatedly applying g_α^β, and depending on the ordering, different initial points of a new flow may be derived. Iwasaki *et al.* [51] introduce the concept of *hypertime* to represent the instantaneous switching time stamp as an infinitely short interval of time. During switching, hypertime elapses, but that corresponds only to infinitesimal actual time changes. The sequence of switches can be analyzed in hypertime to yield similar results.

As described in Chapters III-V, this thesis defines a hybrid dynamic system with model semantics that are more specific than the approaches described above. Instead of allowing several flow values at a point in time on a trajectory, it requires that a flow has a unique representation and lets this unique value be independent of the path during a sequence of instantaneous switches. This requires the implementation of an interval-point paradigm to establish complete coverage of the time-line within the domain of operation of the system. This thesis shows that this more restrictive semantics allow for easier, systematic, model building that results in models whose behaviors do not violate physical system principles. Furthermore, there is no C^2 requirement for behavior at pinnacles, only the point needs to be well-defined in time. This is of great use when time-scale abstractions result in algebraic constraints and system behavior is condensed into a posterior value at a point in time. Finally, the model semantics can be used to deterministically model physical systems and allows for a systematic model verification method [80].

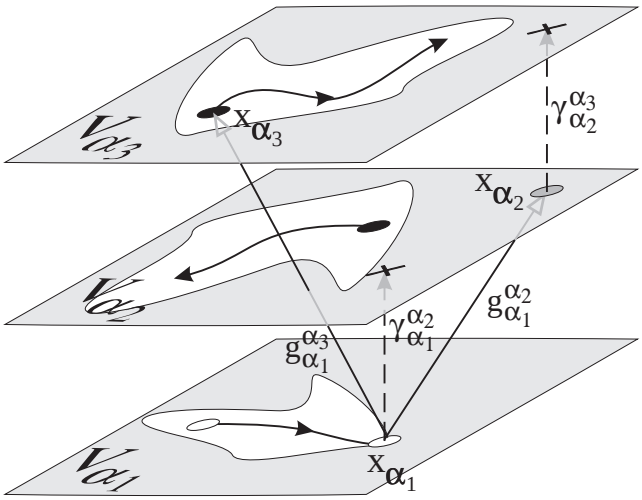


Figure 50: A trajectory is redirected when the transported point is out of the domain of the new patch.

Physical System Model Semantics

To ensure correct transitions between modes in physical system models, the principle of invariance of state applies. During a sequence of transitions, specified by γ^{β}_{α}, a new point x_{α_2} derived by $g^{\alpha_2}_{\alpha_1}$ causes an immediate transition because it is not in U_{α_2}, it is not considered to have an actual representation on the time-line, and does not affect the mapping of x_{α} as shown in Fig. 50. Therefore, when $\gamma^{\alpha_3}_{\alpha_2}$ specifies a transition to V_{α_3}, then new x_{α_3} is derived by applying $g^{\alpha_3}_{\alpha_1}$ to the original point x_{α_1}. Like before, x_{α_2} is mythical, i.e., it has no real existence in time for this behavior trajectory. In general, γ^{β}_{α} is a function that depends on values x_{α}, prior to the jump, and values x^{+}_{α} after the jump. The semantics are specified by the recursive relation between γ^{β}_{α} and g^{β}_{α} which takes the form

$$\begin{cases} x^{+}_{\alpha_k} = g^{\alpha_i}_{\alpha_k}(x_{\alpha_k}) \\ \gamma^{\alpha_i+1}_{\alpha_i}(x_{\alpha_k}, x^{+}_{\alpha_k}) \leq 0 \end{cases} \tag{18}$$

illustrated in Fig. 51. Note the α_k subscript in $g^{\alpha_i}_{\alpha_k}$.

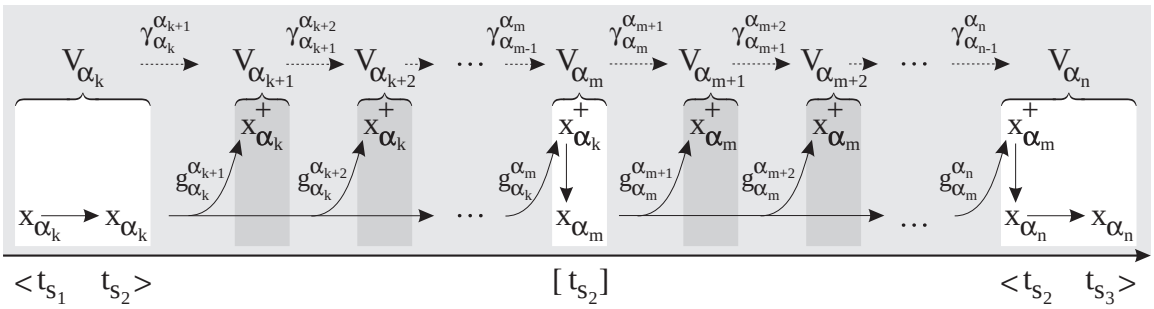


Figure 51: System state is derived from the original state vector.

The interval-point paradigm implies that two types of jumps are possible:

1. $V_{\alpha_k} \multimap\!\!\bullet\, V_{\alpha_m}$: A discontinuity at t_s moves x_{α_k} from an interval, flow $\mathcal{F}_{\alpha_k} \setminus \mathcal{B}_{\alpha_k}$, to a point, pinnacle $\mathcal{P}_{\alpha_{k+1}}$ or flow $\mathcal{F}_{\alpha_{k+1}}$. In this case, $x_{\alpha_k}(t_s^-) = \lim_{t \uparrow t_s} \mathcal{F}_{\alpha_k}(t)$ is transported to $x^+_{\alpha_k}$, computed by Eq. (18) as $g^{\alpha_{k+1}}_{\alpha_k}(x_{\alpha_k})$. The recursive switching terminates when $\gamma^{\alpha_m}_{\alpha_{m-1}}(x_{\alpha_i}, x^+_{\alpha_i}) > 0$ and the function value $x_{\alpha_m}(t_s)$ is taken as $x^+_{\alpha_k} = g^{\alpha_m}_{\alpha_k}(x_{\alpha_k})$.

2. $V_{\alpha_k} \bullet\!\!\multimap\, V_{\alpha_m}$: A discontinuity at t_s moves x_{α_k} from a point, pinnacle $\mathcal{P}_{\alpha_k}$ or boundary point $\mathcal{B}_{\alpha_k}$, to a flow, $\mathcal{F}_{\alpha_{k+1}}$. In this case, $x_{\alpha_k} = x_{\alpha_k}(t_s)$ is transported to $x^+_{\alpha_k}$ computed by Eq. (18) as $x^+_{\alpha_k} = g^{\alpha_{k+1}}_{\alpha_k}(x_{\alpha_k})$. The recursive switching terminates when $\gamma^{\alpha_m}_{\alpha_{m-1}}(x_{\alpha_i}, x^+_{\alpha_i}) > 0$ and the function value $x_{\alpha_m} = \lim_{t \downarrow t_s} \mathcal{F}_{\alpha_m}(t)$ is taken as $x^+_{\alpha_k} = g^{\alpha_m}_{\alpha_k}(x_{\alpha_k})$.

A $\bullet\!\!\multimap$ transition results in the activation of a flow, and the system evolves continuously before a new sequence of switches is initiated.[4]

A trajectory is now described by initially moving along a flow, $\mathcal{F}_{\alpha_1}$, $x^+_{\alpha_1} = g^{\alpha_1}_{\alpha_1}(x_{\alpha_1}) =$

[4]In collision chains, the system may move through a series of pinnacles that represent Newton's collision law, before a new flow is arrived at. Though this behavior can be well captured by the formulated semantics, it violates the interval-point paradigm and may result in poorly defined models since no specific time is know for each of the separate collisions.

x_{α_1}. At time t_s, $\gamma_{\alpha_1}^{\alpha_2}(x_{\alpha_1}, x_{\alpha_1}^+) = 0$ with $x_{\alpha_1} = lim_{t \uparrow t_s} \mathcal{F}_{\alpha_1}(t)$. Therefore, a discrete $\circ\!\!\rightarrow\!\!\bullet$ change occurs, and the trajectory is transported from $x_{\alpha_1}(t_s^-)$ to the point $g_{\alpha_1}^{\alpha_2}(x_{\alpha_1}(t_s^-))$ which results in a posterior value $x_{\alpha_1}^+ = g_{\alpha_1}^{\alpha_2}(x_{\alpha_1}(t_s^-))$. If $x_{\alpha_1}^+ \notin U_{\alpha_2}$ the trajectory is *redirected* by $\gamma_{\alpha_2}^{\alpha_3}(x_{\alpha_1}, x_{\alpha_1}^+) \leq 0$ (Fig. 50), which immediately transports the trajectory to the point $g_{\alpha_1}^{\alpha_3}(x_{\alpha_1}(t_s^-))$. Again, the trajectory may be redirected based on $x_{\alpha_1}^+ = g_{\alpha_1}^{\alpha_3}(x_{\alpha_1}(t_s^-))$ to V_{α_3}. This recursive process continues until an x_{α_m} is arrived at that is within a patch U_{α_m}, $\gamma_{\alpha_1}^{\alpha_m}(x_{\alpha_1}, x_{\alpha_1}^+) > 0$. After the successful transition is made, the *a priori* value is updated to $x_{\alpha_m}(t_s) = g_{\alpha_1}^{\alpha_m}(x_{\alpha_1}(t_s^-))$. If the new point is a pinnacle, $\gamma_{\alpha_m}^{\alpha_{m+1}}(x_{\alpha_m}, x_{\alpha_m}^+) \leq 0$, which leads to $\bullet\!\!\rightarrow\!\!\circ$ switching, and the trajectory is transported from $x_{\alpha_m}(t_s)$ to $g_{\alpha_m}^{\alpha_{m+1}}(x_{\alpha_m}(t_s))$. Based on the new value $x_{\alpha_m}^+$ another sequence of recursive switches may ensue until $x_{\alpha_m}^+$ is within the domain of a patch U_{α_n}. When switching ends, a new flow, $\mathcal{F}_{\alpha_n}$ in V_{α_n}, is reached and the point $x_{\alpha_n}(t_s^+) = g_{\alpha_m}^{\alpha_n}(x_{\alpha_m}(t_s))$ is taken as the initial point and this process continues as system behavior evolves.

Notes

The point x_α initiates switching and controls point-interval evolution in time, whereas x_α^+ drives the recursive switching and determines intermediate charts, V_β, that are traversed before x_α is updated. Furthermore, switching conditions of the form $\gamma_{\alpha_i}^{\alpha_{i+1}} < 0$ are a special case of $\gamma_{\alpha_i}^{\alpha_{i+1}} \leq 0$ where $\mathcal{B}_{\alpha_i}$ is the endpoint of $\mathcal{F}_{\alpha_i}$.

Conjecture VI.1 *Consider a transition sequence* $V_{\alpha_k} \longrightarrow V_{\alpha_m} \longrightarrow V_{\alpha_n}$.

Lemma VI.1 *Domain* V_{α_m} *contains a flow if* $(x = x_{\alpha_k}(t_s))$

$$\mathcal{F}_{\alpha_m} : g_{\alpha_k}^{\alpha_m}(x) = x \tag{19}$$

Proof: If $\gamma^{\alpha_n}_{\alpha_m}(x, g^{\alpha_m}_{\alpha_k}(x)) > 0$ then V_{α_m} is real. If V_{α_m} is a pinnacle, then $\gamma^{\alpha_n}_{\alpha_m}(g^{\alpha_m}_{\alpha_k}(x), g^{\alpha_m}_{\alpha_k}(x)) \leq 0$ which cannot be satisfied for a real mode if $x = g^{\alpha_m}_{\alpha_k}(x)$. ■ If $g^{\alpha_m}_{\alpha_k}(x) \neq x$ then V_{α_m} contains a flow if

$$\mathcal{F}_{\alpha_m} : \gamma^{\alpha_n}_{\alpha_m}(x, g^{\alpha_m}_{\alpha_k}(x)) > 0 \wedge \gamma^{\alpha_n}_{\alpha_m}(g^{\alpha_m}_{\alpha_k}(x), g^{\alpha_m}_{\alpha_k}(x)) > 0 \tag{20}$$

where the first condition ensures the mode is real, and the second that it is not a pinnacle. It consists of a pinnacle if

$$\mathcal{P}_{\alpha_m} : \gamma^{\alpha_n}_{\alpha_m}(x, g^{\alpha_m}_{\alpha_k}(x)) > 0 \wedge \gamma^{\alpha_n}_{\alpha_m}(g^{\alpha_m}_{\alpha_k}(x), g^{\alpha_m}_{\alpha_k}(x)) \leq 0. \tag{21}$$

If V_{α_m} contains a flow, it may not contain the boundary where its flow exits, ($x = x_{\alpha_m}(t_s)$))

$$\mathcal{F}_{\alpha_m} \setminus \mathcal{B}_{\alpha_m} : \mathcal{F}_{\alpha_m} \wedge \gamma^{\alpha_n}_{\alpha_m}(x, g^{\alpha_n}_{\alpha_m}(x)) \leq 0 \tag{22}$$

which implies that V_{α_n} is either a pinnacle or a flow that contains its initial point,

$$\mathcal{F}_{\alpha_m} \setminus \mathcal{B}_{\alpha_m} \Rightarrow \mathcal{P}_{\alpha_n} \vee (\mathcal{F}_{\alpha_n} \cap \mathcal{B}^{\alpha_n}). \tag{23}$$

A Hybrid Dynamic System Implementation Model

To model hybrid dynamic systems, the mathematical model has to be implemented by a model that supports the idiosyncrasies of dynamic physical systems. This section develops this implementation model for embedded control systems, which combine discrete mode-switching behavior with modes of continuous operation [88]. For example, Fig. 52 shows a cooling sub-system in a nuclear reactor. During normal operation, the main motor maintains a constant flow of liquid sodium, controlled by continuous PID controllers. Valves throughout the loop can be operated to move the

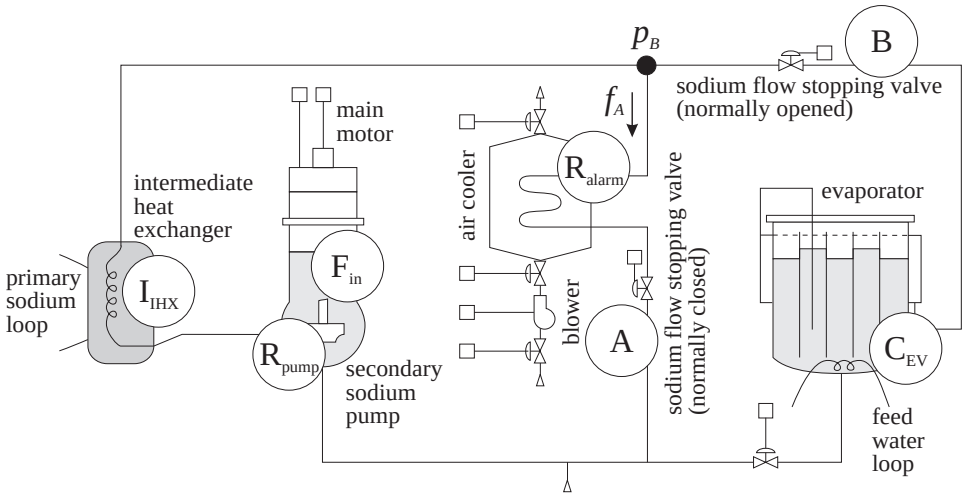


Figure 52: Continuous and discrete process control.

system into different modes of operation. The process and PID controllers are continuous and the opening and closing of valves appear as discontinuous configuration changes. In general, configuration changes in the system can be attributed to three phenomena: (1) when physical system signals, σ_p, cross threshold values; this can be mainly attributed to abstractions incorporated in the physical system and continuous controller model, (2) explicit signals, σ_c, that activate the closed loop controller to make changes, and (3) external, open loop control. The events generated by these phenomena are called, σ_p, σ_c, and σ_x, respectively. Fig. 53 shows the general hybrid architecture of a controlled physical process. The process and its continuous controller embody the continuous characteristics of the system. 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.

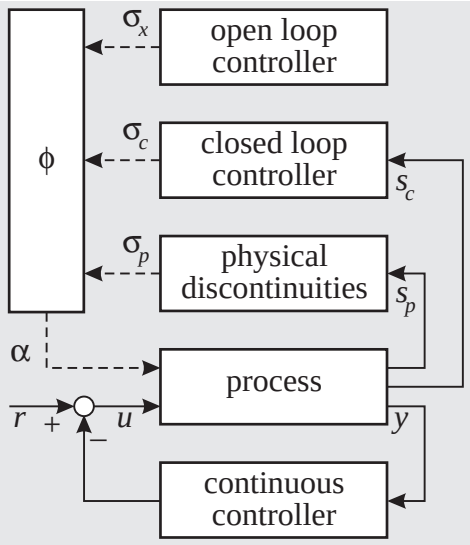


Figure 53: A physical process under hybrid control.

The Continuous Model

Physical system behavior governed by the principles of conservation of energy and continuity of power, is typically described by a state space representation with ordinary differential equations (ODEs),

$$\dot{x}(t) = f(x(t), u(t), t), \tag{24}$$

where $x \in X$ is the vector of the state variables of the system and $u \in U$ the vector of external input. Thus, the continuous system model is made up of:

- $X \in \Re^m$, represents the state vector of the continuous model.
- $U \in \Re^p$, represents the input vector of the continuous model.
- $\dot{x}(t) = f_\alpha(x(t), u(t), t), i \in \aleph, i \in [0, \alpha_{max}], t \in \Re$ defines the continuous behavior of the system in operational mode α for which there is one and only one field, f_α.

The Discrete Model

Discrete events in embedded control systems stem from [59]:

- discontinuous input produced by idealized discrete actuators,
- discontinuous control which switches operational modes based on predefined control algorithms,
- 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 events are of two types: (1) time events and (2) state events [17]. Time events result from digital control, where discrete actuation occurs at a point in time determined by a control algorithm. State events are generated by the process when certain signal values cross specified thresholds and mode transitions are invoked. Since the only spontaneous change in embedded control systems is data [60], these time events are, in principle, state events as well [133]. The discrete changes can be modeled by a state machine, where each state in a set I corresponds to a mode of system operation. The discrete model may consist of a number of independent state machines, in which case an operational mode is determined by the combination of individual states of the independent state machines. Transitions are invoked by events in a set Σ and move the discrete model to a new state based on the transition function, ϕ. This paradigm can be implemented by Petri nets or finite state automata. The discrete model description 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.
- $\phi : I \times \Sigma \rightarrow I$, represents a discrete state transition function that defines the new mode reached after an event occurs. Events are generated by the physical process or the closed loop controller, or they can be external, open loop, control signals, $\Sigma = \Sigma_p \times \Sigma_c \times \Sigma_x$.

Interaction

Interaction between the continuous and discrete part comes in two forms: (1) discrete events generated by the continuous model, and (2) a change of operational mode by the discrete model.

The interaction can be specified by:

- $S \in \Re^n$, the signals used for event generation.
- $g : X \times I \rightarrow X^+$, transfers the continuous state vector to the new operational mode, $\alpha \in I$, which may result in it changing discontinuously. X^+ represents state vector values at the initial point in time when a mode change has occurred.
- $h : X \times U \times I \rightarrow S$, determines signal values S and S^+ computed from X and X^+, respectively.
- $\gamma : S \times S^+ \rightarrow \Sigma_s$, where $\Sigma_s = \Sigma_p \times \Sigma_c$, generates discrete events from the *a priori* and *a posteriori* signal values.

The function γ 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 continuous mode, α, is a field, f_α, specifying continuous behavior evolution. If a mode has no continuous behavior it is completely specified by the algebraic relations in g.

In summary, the complete hybrid system model is defined by the 9-tuple [59]:

$$H =< I, \Sigma, \phi, X, U, f, g, h, \gamma >, \tag{25}$$

with the continuous, discrete, and interface components (Fig. 54). In terms of the mathematical model in Eq. (16), I defines the modes of operation, α. The piecewise continuous domain, V_α, is determined by X and U, and continuous behavior $\mathcal{F}_\alpha$ on each domain is specified by f. The function g_α^β equals g, and γ_α^β is determined by h, γ, Σ, and ϕ. The additional complexity is required because physical configuration switches are typically based on signal values, derived by h from the energy state. Furthermore, the implementation of the discrete model component as a finite state automaton requires and event set, Σ, and transition function, ϕ.

Partial Liquid Sodium Cooling System

Fig. 52 shows a schematic representation of part of the liquid sodium cooling system in a nuclear reactor [81]. The main motor 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. Pump losses are represented by the dissipation parameter, R_{pump}. The 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

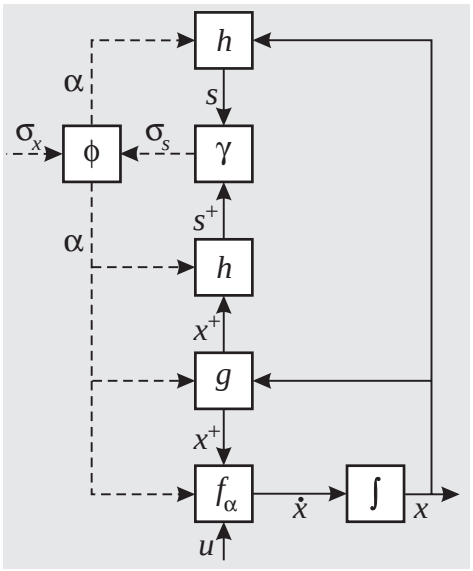


Figure 54: Block diagram of a general hybrid system.

evaporator is used to transport heat from the heat-exchanger to a steam water loop where it drives a turbine to produce electricity. Associated with the evaporator vessel is capacitance C_{EV}. 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_{alarm} by opening valve A until it is safe to stop the flow of fluid completely.

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}

$$x = [x_1 \; x_2]^T. \tag{26}$$

The input to the system is the input flow, F_{in}

$$u = F_{in}. \tag{27}$$

The discrete model is determined by the two valves in the system which results in four operational modes, $\alpha_{00} = \{A_{closed}, B_{closed}\}$, $\alpha_{01} = \{A_{closed}, B_{open}\}$, $\alpha_{10} = \{A_{open}, B_{closed}\}$, and $\alpha_{11} = \{A_{open}, B_{open}\}$, so

$$I = \{\alpha_{00}, \alpha_{01}, \alpha_{10}, \alpha_{11}\} \tag{28}$$

In normal operation, valve A is closed and valve B is open which specifies the initial mode, α_{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 $\sigma_{B\to off}$, the fluid flow becomes 0 instantaneously 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 $\sigma_{A\to on}$ when $p_B > p_{critical}$, which opens valve A. Over time, the pressure falls below $p_{critical}$, and the controller closes the valve A, $\sigma_{A\to off}$. This results in the complete event set

$$\Sigma = \{\sigma_{A\to on}, \sigma_{A\to off}, \sigma_{B\to off}\} \tag{29}$$

The closed loop controller generates events Σ_c, which are specified by γ

$$\gamma : \begin{cases} p_B > p_{critical} \to \sigma_{A\to on} \\ p_B \leq p_{critical} \to \sigma_{A\to off} \end{cases} \tag{30}$$

and the corresponding mode changes are executed by ϕ

$$\phi : \begin{cases} \sigma_{A\to on} \to \alpha_{10} & \text{if } \alpha_{00} \\ \sigma_{B\to off} \to \alpha_{00} & \text{if } \alpha_{01} \\ \sigma_{A\to off} \to \alpha_{00} & \text{if } \alpha_{10} \end{cases} \tag{31}$$

The instantaneous change in flow to 0 when valve B closes represents a reduction in

the size of the continuous state vector, and is captured by g in $x^+ = g \cdot x$,

$$g : \begin{cases} [0\ 1] & \text{if } \alpha_{00} \\ [1\ 1] & \text{if } \alpha_{01} \\ [1\ 1] & \text{if } \alpha_{10} \end{cases} \tag{32}$$

The function h translates the state variables $[x_1\ x_2]^T$ into signal values

$$s = [p_B\ f_A]^T \tag{33}$$

that are used by γ. Note that when both valves are closed, the pressure p_B is determined by a derivative relation, $p_B = F_{in}R_{pump} - I_{IHX}\frac{dx_1}{dt}$, which is approximated by a Dirac pulse, δ, for discontinuous changes in x_1. Let the function $sign$ be defined as

$$sign(x) = \begin{cases} -1 & \text{if } x < 0 \\ 0 & \text{if } x = 0 \\ 1 & \text{if } x > 0 \end{cases} \tag{34}$$

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

$$h : \begin{cases} p_B = sign(x_1^+ - x_1)\delta, f_A = 0 & \text{if } \alpha_{00} \\ p_B = \frac{x_2^+}{C_{EV}}, f_A = 0 & \text{if } \alpha_{01} \\ p_B = R_{alarm}\frac{x_1^+}{I_{IHX}}, f_A = \frac{x_1^+}{I_{IHX}} & \text{if } \alpha_{10} \end{cases} \tag{35}$$

and the specification of the hybrid system is complete.

Model Verification

A key aspect of hybrid system modeling is to define the interaction between the continuous and discrete components in an integrated manner, without violating overall physical system principles. Chapter IV shows how consistency of interaction is

maintained if divergence of time is satisfied. Chapter V employs energy phase space analysis to verify divergence of time based on the principle of invariance of state as developed in Chapter III. This section presents the mathematical equivalent of the energy phase space analysis methodology for a generalized state vector of linear systems. Furthermore, it introduces the principle of temporal evolution of state as an additional verification tool to ensure well defined models.

Generalized Invariance of State

Chapter IV shows that a *special* continuous state vector, p_0, of a physical system model represents the stored energy in physical buffer elements, e.g., springs, capacitors, and inertias, and is invariant across consecutive changes in operational mode [87].

Conjecture VI.2 *The special state vector, p_0, is invariant across mode changes.*

Lemma VI.2 (Generalized Invariance of State) *Any vector that represents the state of a linear physical system is invariant across mode changes.*

Proof: Let x_0 represent a possible state vector in operational mode α_0. Then, for a linear system, there is an algebraic translation T_0 unique to a given operational mode that defines the relation between x_0 and the special state vector p_0 defined above, i.e., $x_0 = T_0(p_0)$. Since p_0 is invariant across mode changes, so is $T_0^{-1}(x_0)$. If x_n is a vector capturing the system state after the mode changes end, then $x_n = T_n(p_n)$ and, since p_0 is invariant across transitions, x_n can be expressed as, $x_n = T_n(g(p_0))$. So, $x_n = T_n(g(T_0^{-1}(x_0)))$ also is invariant because the function g and the mappings

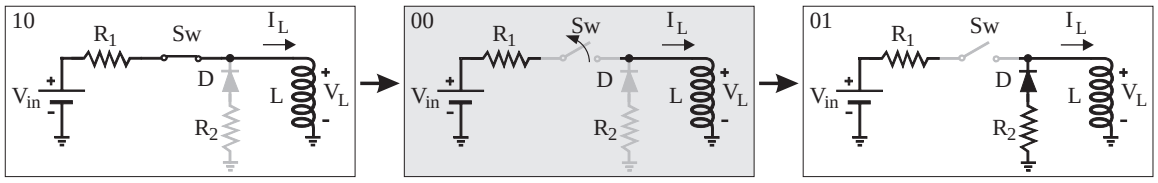


Figure 55: Invariance of state in a dynamic physical system.

T_n and T_0^{-1} are defined by the specific operational mode and is not a function of how this mode was achieved, i.e., they are path invariant. ■

The continuous state vector of the diode-inductor circuit in Fig. 55 (which is also discussed in Chapters I and IV) 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

$$T : \begin{cases} V_L = 0 & \text{if } \alpha_{00} \\ V_L = I_L R_L + V_D & \text{if } \alpha_{01} \\ V_L = -I_L R_L + V_{in} & \text{if } \alpha_{10} \end{cases} \tag{36}$$

The principle of invariance of state in Chapter III states that if the inductor current is chosen to represent system state 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 [79].[5]

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 $T_{\alpha_{10}}^{-1} : I_{L,0} = -\frac{1}{R_1} V_{L,0} + \frac{1}{R_1} V_{in}$ the new continuous system state can be expressed

[5]Note that $g(x) = x$ is a special case of mapping the state vector.

in its value before switching by $T_{\alpha_{01}}(T_{\alpha_{10}}^{-1})$,

$$V_L^+ = \frac{R_2}{R_1} V_{L,0} + \frac{R_2}{R_1} V_{in} + V_D \tag{37}$$

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

Temporal Evolution of State

Changes in the number of degrees of freedom of a system can occur because of dependencies caused by configuration changes. The result is dependencies among state variables and exogenous variables, producing discontinuous changes in values that can only occur at well-defined points in time. Continuity of power requires well-defined functions on the left and right intervals about the point of discontinuity (Fig. 51), therefore, configuration changes cause piecewise continuous behaviors with a countable number of *simple* discontinuities which have a limit value [108]. As demonstrated earlier in this chapter, this defines an interval-point paradigm for active modes of operation. It is shown that the state vector at the point of discontinuity is the limit value of the state in the *new* operational mode.

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

Lemma VI.3 (Temporal Evolution of State) *Continuous state variable values have to be continuous in left-closed intervals.*

Proof: For simple discontinuities, limit values exist, and, therefore, at a time of switching, t_s, $x_k(t_s^-) = \lim_{t \uparrow t_s} x_k(t)$, and $x_n(t_s^+) = \lim_{t \downarrow t_s} x_n(t)$ (Fig. 51). In case of a

jump, $x_k(t_s^-) \neq x_n(t_s^+)$. Because the state vector exists for all points on the real time-line, there is a state vector $x_m(t_s)$ determined at t_s. If the state vector at t_s, $x_m(t_s) \neq x_n(t_s^+)$ then the system continues to evolve in a left open interval, $< t_s, \rightarrow >$, after configuration changes have occurred, starting with $\lim_{t \downarrow t_s} x_n(t)$. However, causality requires that the initial state in the new configuration be a function solely of $x_m(t_s)$ and the new configuration, and, therefore, the state vector has to evolve in left-closed intervals, $[t_s, \rightarrow >$. ■

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. The signals as derived by h may contain Dirac pulses which occur at the points in between intervals of continuous operation.

Divergence of Time

Chapter IV presents the principle of divergence of time [47, 87] as an important verification mechanism for hybrid modeling of embedded control systems. Chapter V shows that this 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 γ to ϕ 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.

Verification of the Cooling System

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

$$\begin{cases} F_{in}R_{pump} - sign(x_1^+ - x_1)\delta > p_{critical} & \text{if } \alpha_{00} \\ R_{alarm}\frac{x_1^+}{I_{IHX}} \leq p_{critical} & \text{if } \alpha_{10} \end{cases} \tag{38}$$

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

$$\begin{cases} F_{in}R_{pump} - sign(-x_1)\delta > p_{critical} & \text{if } \alpha_{00} \\ R_{alarm}\frac{x_1}{I_{IHX}} \leq p_{critical} & \text{if } \alpha_{10} \end{cases} \tag{39}$$

Therefore, closed loop switching events are generated when $F_{in}R_{pump} - sign(-x_1)\delta > p_{critical}$ and $x_1 \leq \frac{I_{IHX}}{R_{alarm}}p_{critical}$. Considering that δ approaches infinity, there is an area $0 < x_1 \leq \frac{I_{IHX}}{R_{alarm}}p_{critical}$ where the system switches between modes α_{00} and α_{10} indefinitely. For this flow momentum, the system is not consistent as it is not

determined which of the operational mode is reached.[6] Note that, if $x_1 = 0$ then $F_{in}R_{pump} > p_{critical}$ causes inconsistency, which is true if $F_{in} \geq \frac{p_{critical}}{R_{pump}}$.

The interaction between the discrete and continuous domain shows that a control algorithm based on pressures 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 has fallen below a safe threshold value, f_{th}, build-up of pressure does not exceed the critical value and the alarm valve can safely be closed. To specify these constraints, $x_1 \leq I_{IHX}f_{th}$ is added to the precondition for $\sigma_{A\rightarrow off}$ and $x_1 > I_{IHX}f_{th}$ to $\sigma_{A\rightarrow on}$. Now, a unique operational mode is specified for the complete hybrid system if $F_{in} < \frac{p_{critical}}{R_{pump}}$. Note that the added condition is of an *energy* nature, since it is based on the flow momentum of the system.

Summary

In general, hybrid dynamic systems can be classified into three categories:

- *Weak* hybrid dynamic systems do not allow discontinuous changes of the energy state, nor do they allow sequences of mode changes, i.e., $g_\alpha^\beta(x) = x$ and $h_\alpha = h_\beta$.
- *Mild* hybrid dynamic systems deal with discontinuities in the energy state variables, but do not allow sequences of changes, i.e., $g_\alpha^\beta(x) \neq x$ and $h_\alpha = h_\beta = \emptyset$.
- *Strong* hybrid dynamic systems generate time and state events that may result in sequences of mode changes as well.

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

The described modeling formalism and semantics cover the spectrum of hybrid dynamic systems. It differs from other approaches by recognizing that there are two types of modes: real modes that affect the state vector and energy variables directly, and mythical modes that do not. If this distinction is not made, and only real modes are allowed, it is the task of the modeler to ensure only real modes are achieved. In the proposed model semantics, any number of mythical modes exist at a point in time, but there is one and only one real mode for each point in time. This distinction allows for a systematic modeling approach and verification methodologies to ensure that the hybrid models generate correct behaviors.

CHAPTER VII

FORMAL SPECIFICATIONS FROM HYBRID BOND GRAPHS

This chapter brings together the material discussed in Chapters III-VI, and uses the classic example of a thin rod colliding with a floor [64] to show how formal specifications developed in Chapter VI can be systematically derived from the hybrid bond graph model presented in Chapter IV [83, 84]. Model verification issues are also explored in greater detail in this chapter.

Specifying the Falling Rod

Consider an idealized rigid thin rod falling toward a perfectly rigid floor at an angle θ shown in Fig. 56. The hybrid bond graph model of the resultant collision is shown in Fig. 57. The model fragments in the bond graph, based on rigid body mechanics, were proposed by Bos [12]. Assuming the movement of the rod is only in the x-y plane, where x is the axis along the floor and y the vertical axis, the rod has three degrees of freedom. Its velocity can be broken down into three components: linear velocities in the x and y directions, v_x and v_y, and an angular velocity, ω. The corresponding storage elements for these components are the inertia or mass elements, m_x, m_y, and J, respectively. The geometric relations between these velocities are illustrated in Fig. 56 and represented in the bond graph model by a modulated transformer. Gravity is modeled by a source of constant force, ma_g, working on the y component of the center of mass.

The forces and velocities at point A connect to the model at the 0_C junctions,

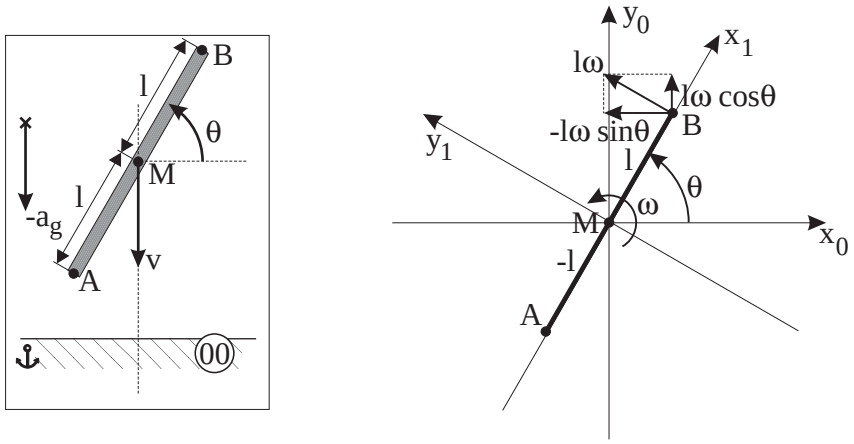


Figure 56: A collision between a thin rod and a floor.

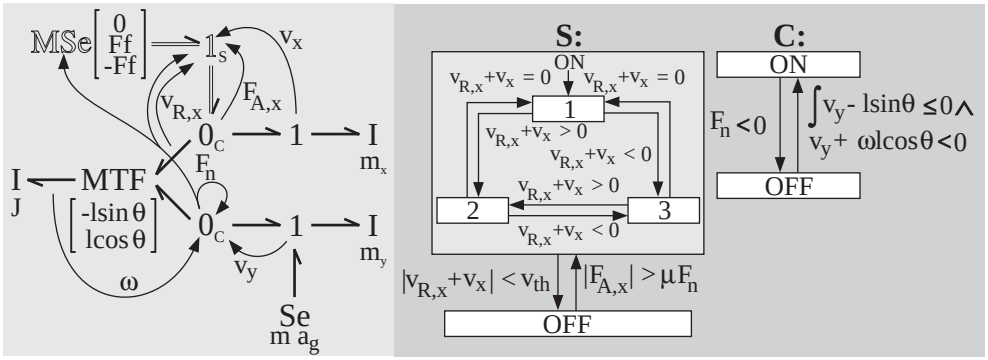


Figure 57: Hybrid bond graph model of a rigid body collision between a thin rod and a floor.

one for the x direction and one for y. When the body is moving freely, forces on this junction are 0, it is *off*. When the body is in contact with the floor, 0_C is *on* and if no other elements are connected, it enforces a 0 velocity at point A. The floor surface has associated Coulomb friction, F_f, whose magnitude depends on the normal force, F_n, exerted by the surface and a friction coefficient, μ, as $F_f = \mu F_n$. On collision, if its force along the surface is large enough, i.e., $|F_{A,x}| > \mu F_n$, the rod starts to slide along the floor and rotate around the point of contact. This sliding motion is invoked by a discrete event σ_{slide} that is generated when $|F_{A,x}| > \mu F_n$ (Fig. 58), which causes the discrete event model, ϕ, to transfer to a discrete mode where the continuous field, f, models a rod sliding on a floor under Coulomb friction. If the velocity of the rod-tip, A, along the floor falls below a threshold value, $|v_{x,A}| \leq v_{th}$, the discrete event σ_{stuck} is generated and ϕ moves the system into an operational mode where the rod is stuck at the point of contact and rotates around it. This represents an example of closed loop physical events. In the hybrid bond graph, the friction force in the x direction is represented by a piecewise continuous modulated source, MS_e. Depending on whether the rod sticks on the floor or slides, friction exerts a force $0, F_f, -F_f$ on A (see Fig. 59). The sign of the friction force depends on the direction in which the rod slides, because friction always opposes motion.

The CSPEC part of the hybrid model is specified as finite state automata, one for each controlled junction. The hierarchical finite state machine that controls 0_S can be in one of several *on* states. Each one activates a region of the piecewise continuous friction function discussed above. The Coulomb friction (Fig. 59) can be represented in a concise form in the hybrid bond graph using the multi-bond notation [14], illustrated in Fig. 60. In its *off* state, the junction enforces 0 flow.

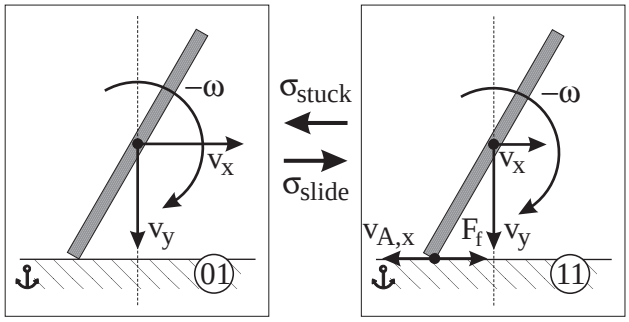


Figure 58: Sliding and rotating rod under Coulomb friction.

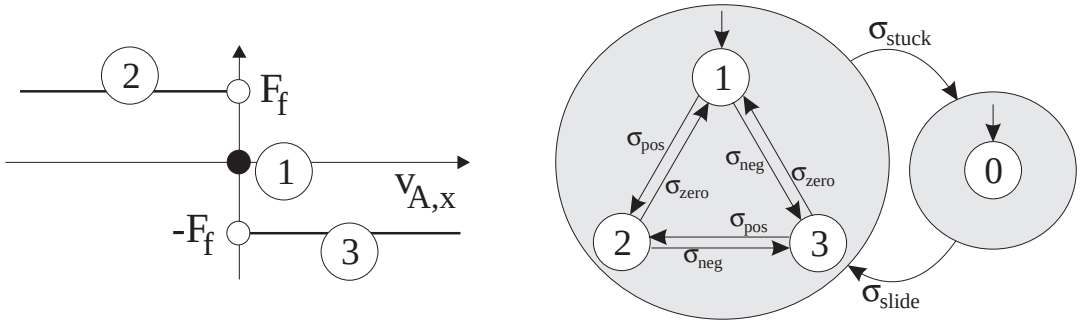


Figure 59: Coulomb friction causes physical events.

This allows all source elements to be continuous functions over their active areas and makes mode switches that are internal to sources explicit in the hybrid bond graph framework, and, therefore, part of the mode-switching algorithm. This guarantees consistency in behavior generation, since all discrete phenomena are handled by one mechanism and all other influences are continuous.

Fig. 61 illustrates the different modes of operation for the rod system. Initially, the rod is falling freely under gravity and this corresponds to mode α_{00} of the system.

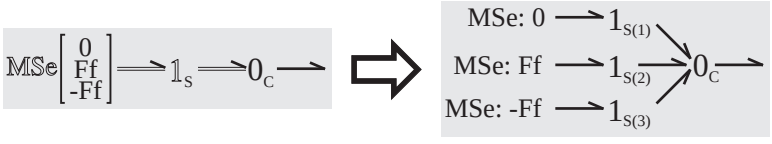


Figure 60: A multi-bond controlled junction to model Coulomb friction.

In the hybrid bond graph, the controlled junctions 0_C and 1_S are *off* and replacing the junctions with their 0 value sources results in the bond graph (mode α_{00}) shown in Fig. 62. To ensure correct loading, not all 0 value sources can be removed (grayed out) from the bond graph. The position of the rod-tip closest to the floor, y_A, is determined by the sum of the position of the center-point, $y_M = \int v_y$, and the distance of the rod-tip from the center point, $-lsin\theta$. When $y_A = \int v_y - lsin\theta$ becomes 0, this implies the rod has collided with the floor and 0_C comes *on*. The model transitions into mode α_{01}. This causes dependency between the linear and angular velocities, and the energy redistribution is computed which determines the forces that the rod exerts in the horizontal direction, $F_{A,x}$, and vertical direction, $F_{A,y}$. If the rod-length and angle of collision are such that $|F_{A,x}| > \mu F_n$, 1_S comes *on* (i.e., the model transitions into mode α_{11}) and the rod begins to slide. Otherwise, it sticks and rotates around the point of contact, i.e., mode α_{01}. In case the rod starts to slide, its initial kinetic energy before contact is redistributed over the angular and vertical momentum to ensure the vertical velocity of the rod-tip, $v_{A,y}$, is 0. This corresponds to mode α_{11}. In this mode, the horizontal velocity of the rod-tip, $v_{A,x}$, is determined by the angular velocity, ω, and the friction force F_f, which is initially 0 given $v_{A,x} = 0$. However, because of the discontinuous change of ω upon collision, $v_{A,x}$ changes discontinuously as well, and, therefore, the system changes from the operational mode where $F_f = 0$ to mode α_{21} where $F_f = \mu F_n$ given $v_{A,x} < 0$. Note that in modes α_{11} and α_{21} the initial vertical momentum is distributed only over its posterior angular momentum and vertical momentum to ensure y_A does not change at the point of contact. If the energy state vector in the sliding mode, α_{11}, was computed from the intermediate stuck mode, α_{01}, it would have a horizontal velocity associated with its center of

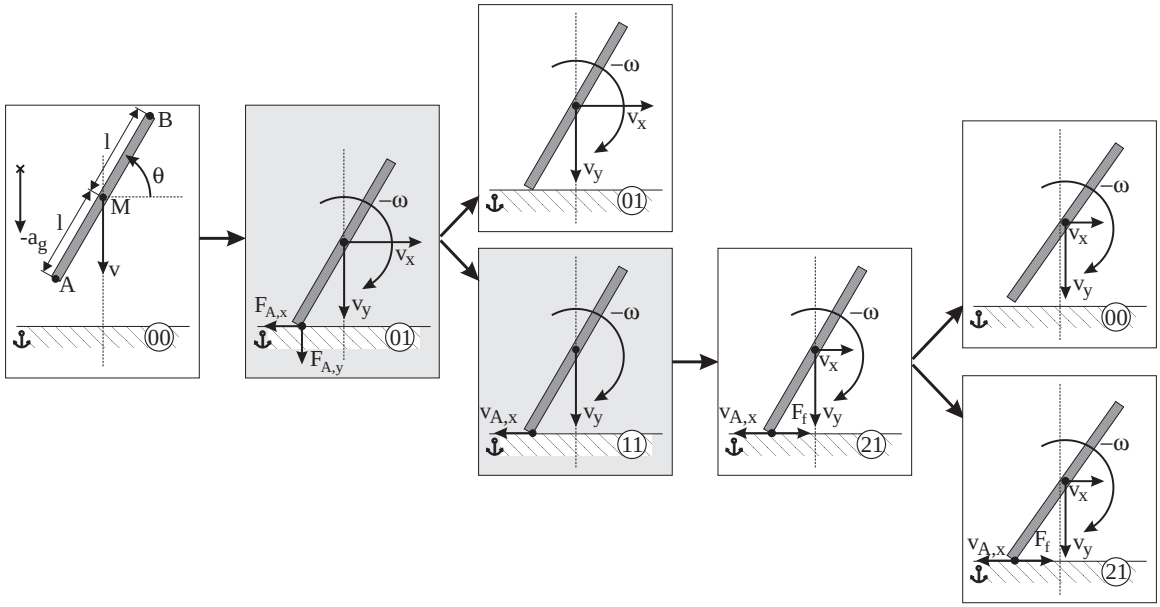


Figure 61: Possible modes of operation of a thin rigid rod falling onto a rigid floor.

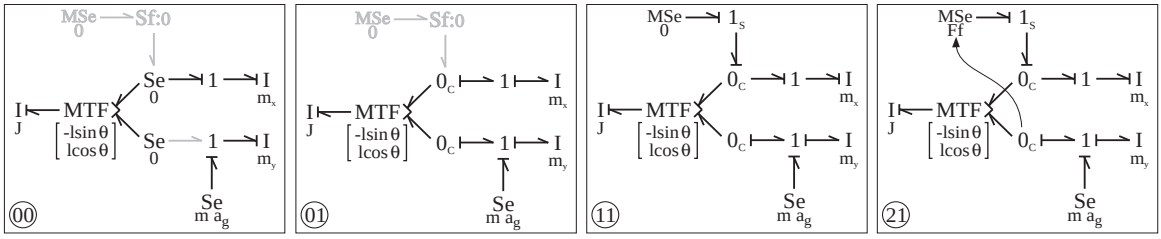


Figure 62: Dynamically generated model configurations for the colliding rod.

mass which would keep the rod-tip from moving in the x direction as well. This behavior would be incorrect. All this demonstrates the importance of the proper computation of the state vector across a series of discontinuous changes. As shown by this example, the hybrid bond graph approach provides a seamless integration of configuration changes based on local switches. Other examples of hybrid bond graph models are discussed in [77, 78, 79].

The Continuous Model

The continuous model can be directly derived as explicit ODEs in each operational mode from the hybrid bond graph in Fig. 62. The linear and angular velocities are chosen as state variables,

$$X = \{\omega, v_x, v_y\} \tag{40}$$

and the input vectors include friction and gravitation forces,

$$U = \{F_f, ma_g\}. \tag{41}$$

As an example, consider operational mode α_{21}. The bond graph shows the J inertia to be in integral causality, and, therefore, its angular velocity, ω, is completely determined by the torque, τ,

$$\dot{\omega} = \frac{\tau}{J} \tag{42}$$

The transformer determines the torque value as the sum of horizontal and vertical forces $\tau = -lsin\theta F_f + lcos\theta(F_y - ma_g)$ where the friction force, F_f, is determined by the normal force on the rod $F_f = \mu F_n = \mu(F_y - ma_g)$. Combining these equations and substituting $m\dot{v}_y$ for F_y, yields

$$\tau = (lcos\theta - \mu lsin\theta)(m\dot{v}_y - ma_g). \tag{43}$$

If the Coriolis component is neglected, $\dot{v}_y = lcos\theta\dot{\omega}$ and $\dot{\omega}$ can be solved as

$$\dot{\omega} = \frac{-ml(cos\theta - \mu sin\theta)}{J + ml^2 cos\theta(cos\theta - \mu sin\theta)} a_g \tag{44}$$

and

$$f_{\alpha_{21}} : \begin{cases} \dot{\omega} = \frac{-ml(cos\theta - \mu sin\theta)}{J + ml^2 cos\theta(cos\theta - \mu sin\theta)} a_g \\ \dot{v}_x = -\mu(lcos\theta\dot{\omega} + a_g) \\ \dot{v}_y = -lcos\theta\dot{\omega} \end{cases} \tag{45}$$

Similarly, the fields for the other operational modes can be derived as explicit ODEs and these are given in Appendix A.

The Discrete Model

The discrete model, as specified in the CSPEC part of the hybrid bond graph in Fig. 57 consists of two independent state machines, one of which is hierarchical. The corresponding state transition tables are given in Appendix A. The global state of the discrete model is a combination of the states of the **C** and **S** automata, indicated by two digits, one representing the state of **C** and one representing the state of **S**, e.g., $\alpha_{01} \rightarrow \mathbf{S}(0) \wedge \mathbf{C}(1)$ and $\alpha_{21} \rightarrow \mathbf{S}(2) \wedge \mathbf{C}(1)$, where 0 indicates the corresponding junction is *off* and any other number indicates the corresponding active state. The discrete event set consists of the events that capture Coulomb friction and the events that specify whether there is contact, $\sigma_{contact}$ and σ_{free}.

Interaction

Signal values from the continuous model are used to generate discrete events based on the function, γ. The events that determine whether there is contact or not are based on the vertical position of the rod-tip, y_A, whether the rod-tip has positive or negative momentum, $p_{A,y}$, and the normal force exerted by the floor, F_n. To determine the state of the Coulomb friction function, the force exerted by the rod-tip in the horizontal direction, $F_{A,x}$, the normal force, F_n, and the horizontal velocity of the rod-tip, $v_{A,x}$, are used. A threshold velocity, v_{th}, has to be maintained for the rod to keep sliding. To reiterate the convention in this thesis, energy variables constitute the *a priori* switching values. The energy variables for this falling rod example are

$p_{A,y}$ and y_A. All other signals are *a posteriori* values, indicated by a + superscript in the event generation function,

$$\gamma : \begin{cases} y_A \leq 0 \wedge p_{A,y} \leq 0 & \Rightarrow \sigma_{contact} \\ F_n^+ \leq 0 & \Rightarrow \sigma_{free} \\ |F_{A,x}^+| - \mu F_n^+ > 0 & \Rightarrow \sigma_{slide} \\ |v_{A,x}^+| - v_{th} \leq 0 & \Rightarrow \sigma_{stuck} \\ v_{A,x}^+ = 0 & \Rightarrow \sigma_{zero} \\ v_{A,x}^+ < 0 & \Rightarrow \sigma_{neg} \\ v_{A,x}^+ > 0 & \Rightarrow \sigma_{pos} \end{cases} \tag{46}$$

The signal values

$$S = \{y_A, v_{A,x}, v_{A,y}, F_n, F_{A,x}\} \tag{47}$$

are derived from the *a priori* or *a posteriori* state variables using the function h. Except for F_n and $F_{A,x}$, these signals are independent of the model configuration. This determines

$$h : \begin{cases} y_A = \int v_y dt - lsin\theta \\ v_{A,x} = v_x - l\omega sin\theta \\ p_{A,y} = m(v_y + l\omega cos\theta) \\ F_n = \begin{cases} 0 & \text{if } \alpha_{00} \\ m(\dot{v}_y - a_g) & \text{otherwise} \end{cases} \\ F_{A,x} = \begin{cases} 0 & \text{if } \alpha_{00} \\ m\dot{v}_x & \text{otherwise} \end{cases} \end{cases} \tag{48}$$

Note that the derivative nature of the forces results in Dirac pulses during mode changes if there are discontinuous changes in the posterior signal values, s^+.[1] If a

[1]This does not necessarily require the posterior values of the energy state x^+ to change discontinuously as well.

discontinuous change occurs, a Dirac pulse, $\delta(s^+ - s)$, is generated which has area $s^+ - s$. This signal dominates continuous variables in switching conditions and comparison between the Dirac pulses is based on their respective areas. For example, upon collision, $F_{A,x} = m\delta(v_x^+ - v_x)$ and $F_n = m(\delta(v_y^+ - v_y) - a_g)$. Because of the magnitude of the Dirac pulse, the a_g term in F_n can be neglected under discontinuous change of v_y, and the comparison of $|\delta(v_x^+ - v_x)| > \mu\delta(v_y^+ - v_y)$ can be reduced to a comparison of $|v_x^+ - v_x| > \mu(v_y^+ - v_y)$ to determine whether the σ_{slide} event is generated to activate the sliding mode.

When a new operational mode is activated by the discrete model, the function g specifies the mapping of the original state vector to the state vector specified in the new configuration. Discontinuous changes to the state vector are derived using Eq. (5). To illustrate, the derivation of $g^{\alpha_{01}}$ follows. Operational mode α_{01} shows dependency between the three storage elements, J, m_x, and m_y, with stored energy h_ω, p_x, and p_y, respectively. This implies that the center of mass has nonzero angular velocity but the rod-tip A does not move in the horizontal or vertical direction since it is in contact with the floor and stuck. Choosing J as the independent storage element, makes m_x and m_y dependent, therefore,

$$\begin{cases} a_{\delta,m_x} = p_{m_x}^+ - p_x, \; p_x^+ = r_{J,m_x}\frac{m_x}{J}h_\omega^+ \\ a_{\delta,m_y} = p_{m_y}^+ - p_y, \; p_y^+ = r_{J,m_y}\frac{m_y}{J}h_\omega^+, \end{cases} \tag{49}$$

and,

$$\sum_{storage} = (r_{J,m_x}\frac{m_x}{J}h_\omega^+ - p_x)r_{m_x,J} + (r_{J,m_y}\frac{m_y}{J}h_\omega^+ - p_y)r_{m_y,J}. \tag{50}$$

No δ-sources become active on switching, so $\sum sources = 0$. The complete expression

for the independent energy, h_ω^+ now yields

$$h_\omega^+ = h_\omega + r_{m_x,J} r_{J,m_x} \frac{m_x}{J} h_\omega^+ - r_{m_x,J} p_x + r_{m_y,J} r_{J,m_y} \frac{m_y}{J} h_\omega^+ - r_{m_y,J} p_y. \tag{51}$$

This can be transformed into the state variables by the translations $h_\omega = J\omega$, $p_x = mv_x$, and $p_y = mv_y$ and substitution of the gains of the respective routes, found by tracing power amplification along a route following causal strokes (Fig. 62),

$$\begin{cases} r_{m_x,J} = -lsin\theta, \ r_{J,m_x} = lsin\theta \\ r_{m_y,J} = lcos\theta, \ r_{J,m_y} = -lcos\theta \end{cases} \tag{52}$$

which yields

$$\omega^+ = \frac{\omega J - ml(cos\theta v_y - sin\theta v_x)}{J + ml^2} \tag{53}$$

and

$$g^{\alpha_{01}} : \begin{cases} \omega^+ = \frac{\omega J - ml(cos\alpha v_y - sin\alpha v_x)}{J + ml^2} \\ v_x^+ = l\omega^+ sin\alpha \\ v_y^+ = -l\omega^+ cos\alpha \end{cases} \tag{54}$$

A situation that involves δ-sources occurs when the rod achieves mode α_{21}, the sliding mode of operation (Fig. 56). In this mode there is dependency between the two storage elements, J and m_y, with stored energy h_ω and p_y, respectively. This implies that the center of mass velocity is such that the rod-tip A has no vertical motion since it is in contact with the floor. Choosing J as the independent storage element, this results in one dependent storage element, m_y,

$$a_{m_y} = p_{m_y}^+ - p_y, \ p_y^+ = r_{J,m_y} \frac{m_y}{J} h_\omega^+, \tag{55}$$

and,

$$\sum_{storage} = (r_{J,m_y} \frac{m_y}{J} h_\omega^+ - p_y) r_{m_y,J} \tag{56}$$

There is one δ-source, F_f, whose value is determined by the Dirac pulse on storage element m_y, by $F_f = \mu a_{m_y}$. Substitution of Eq. (55) gives

$$\sum_{sources} = \mu(r_{J,m_y}\frac{m_y}{J}h_\omega^+ - p_y)r_{F_w,J} \tag{57}$$

The complete expression for the independent energy, h_ω^+ now yields

$$h_\omega^+ = h_\omega + r_{m_y,J}r_{J,m_y}\frac{m_y}{J}h_\omega^+ - r_{m_y,J}p_y + \mu r_{F_w,J}r_{J,m_y}\frac{m_y}{J}h_\omega^+ - \mu r_{F_w,J}p_y \tag{58}$$

and

$$g^{\alpha_{21}} : \begin{cases} \omega^+ = \frac{\omega J - ml(cos\alpha - \mu sin\alpha)v_y}{J + ml^2 cos\alpha(cos\alpha - \mu sin\alpha)} \\ v_x^+ = -\mu(l\omega^+ cos\alpha + v_y) + v_x \\ v_y^+ = -l\omega^+ cos\alpha \end{cases} \tag{59}$$

In a similar manner, the state vector mapping can be derived for the other operational modes, described in Appendix A.

These mapping functions specify the change of system state, (ω, v_x, v_y), as a function of only the new mode, i.e., g^α. To find out whether a discontinuity occurs when the state is mapped between specific modes α and β, g_α^β can be derived from $g^\beta(x) = g_\alpha^\beta(g^\alpha(x))$, which, in case g^α is invertible, yields $g_\alpha^\beta(x) = g^\beta(g^{\alpha-1}(x))$. To verify discontinuity in state variables from α to β, a faster approach is to calculate $g^\beta(g^\alpha(x))$. If the mapping $g^\alpha(x)$ is invariant under g^β, no discontinuous change can occur from α to β.

These calculations can be avoided all together by inspecting the bond graph for changes in derivative causality between modes. Additional elements in derivative causality between modes indicate discontinuous changes in the state vector. For example, in Fig. 62 both the inertial elements m_x and m_y are in derivative causality for mode α_{01}, only m_y is in derivative causality for modes α_{11} and α_{21}. When transferring

the state from mode α_{01} to, say, α_{21} no discontinuous change occurs. However, when transferring the state from mode α_{21} to α_{00}, m_x goes into derivative causality and may enforce a discontinuous change in state. To verify, $g^{\alpha_{21}}(g^{\alpha_{01}}(x))$ is derived:

$$\begin{cases} \omega^+ = \frac{\omega^{\alpha_{01}} J - ml(cos\theta - \mu sin\theta) v_y^{\alpha_{01}}}{J + ml^2 cos\theta(cos\theta - \mu sin\theta)} \\ v_x^+ = -\mu(l\omega^+ cos\theta + v_y) + v_x \\ v_y^+ = -l\omega^+ cos\theta \end{cases} \tag{60}$$

Substituting $v_y^{\alpha_{01}} = -l\omega^{\alpha_{01}} cos\theta$ (see Eq. (54)) in ω^+, yields

$$\omega^+ = \frac{\omega^{\alpha_{01}} J + ml^2 cos\theta(cos\theta - \mu sin\theta)\omega^{\alpha_{01}}}{J + ml^2 cos\theta(cos\theta - \mu sin\theta)} \tag{61}$$

or $\omega^+ = \omega^{\alpha_{01}}$. Substituting this result in v_x^+ and v_y^+ yields

$$\begin{cases} \omega^+ = \omega^{\alpha_{01}} \\ v_x^+ = -\mu(l\omega^{\alpha_{01}} cos\theta + v_y^{\alpha_{01}}) + v_x^{\alpha_{01}} \\ v_y^+ = -l\omega^{\alpha_{01}} cos\theta \end{cases} \tag{62}$$

again substituting $v_y^{\alpha_{01}} = -l\omega^{\alpha_{01}} cos\theta$ shows that the complete state vector in α_{21}, $x^+ = x^{\alpha_{01}}$, and, therefore, no discontinuous change occurs. When performing the same computations on a mode switch from α_{21} to α_{01}, x^+ remains a function of x which implies that a discontinuity does occur. Intuitively, these results can be explained by noting that α_{01} represents a mode of operation where the rod is stuck and rotating around a stationary point. In α_{11} and α_{21}, the rod is sliding, and, therefore, it has an additional degree of freedom. When the system moves from α_{01} to either α_{11} or α_{21}, the degrees of freedom in the state vector increase by one, and this allows continuous state transfer. However, when the rod gets into the stuck mode from the sliding mode, its center of mass velocity in the x direction becomes dependent on the angular velocity, and this may require a discontinuous change in state.

Verification of the Hybrid Falling Rod Model

The correctness of the hybrid falling rod model can be established by demonstrating that the principles of temporal evolution of state (discussed in Chapter VI) and divergence of time (discussed in Chapter IV) are satisfied.

Temporal Evolution of State

Causality of physical system models is embodied by the principle of temporal evolution of state. To illustrate the importance of this principle, observe that switching conditions based on relations that include time-derivatives of signals are common in physical system models. When discontinuous changes occur, the derivative relations produce Dirac pulses, and this can often lead to ill defined behavior. For example, $|F_{A,x}| > \mu F_n$ relies on $\dot{v}_x$ and $\dot{v}_y$, and a discontinuous change in these variables when a mode change occurs produces a Dirac pulse, $\delta(t_s)$. The Dirac pulse is active at a well defined point in time, t_s, and may be generated by either $\circ\!\!\rightarrow\!\!\bullet$ (t_s^- to t_s) transitions, $\delta_1(t_s)$, or $\bullet\!\!\rightarrow\!\!\circ$ (t_s to t_s^+) transitions, $\delta_2(t_s)$ (see Fig. 63). The actual Dirac pulse at t_s is then the cumulative $\delta_c(t_s) = \delta_1(t_s) + \delta_2(t_s)$ and is determined by the $\circ\!\!\rightarrow\!\!\circ$ step. Therefore, switching conditions that are based only on $\delta_1(t_s)$ may be incorrectly executed if $\delta_2(t_s)$ interferes with the Dirac pulse computation.

Upon collision with the floor at t_s, the horizontal and vertical velocities of the center of mass of the rod change discontinuously. So, $v_x(t_s^-) = \lim_{t \uparrow t_s} v_x(t)$ differs from $v_x(t_s)$ which results in a collision impulse $P_{x,c}$ and $v_y(t_s^-) = \lim_{t \uparrow t_s} v_y(t)$ differs from $v_y(t_s)$ which results in a normal impulse P_n as shown in Fig 64. Since no other forces are active $P_{A,x} = P_{x,c}$, and if the force balance is such that $|P_{A,x}| > \mu P_n$ the model specifies that the rod goes into mode $\alpha_{21,a}$ and starts to slide. Consider a stiction

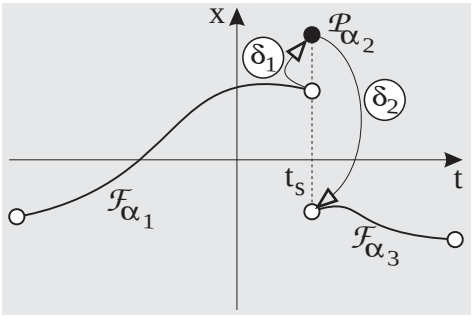


Figure 63: Switching around a point may require two jumps.

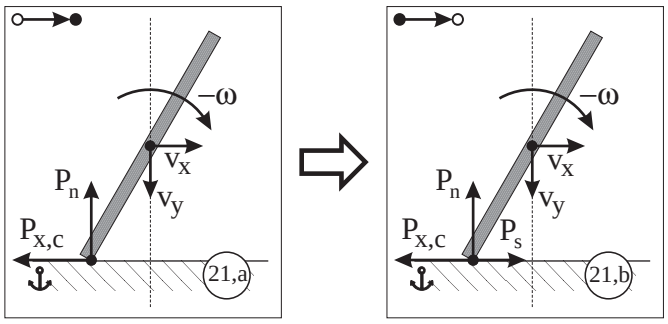


Figure 64: Impulses upon collision when a sliding mode with stiction is reached.

impulse that becomes active when the rod starts to slide (mode $\alpha_{21,b}$) and this causes a discontinuous change in the horizontal velocity of the rod, so $v_x(t_s^+) = \lim_{t \downarrow t_s} v_x(t)$ differs from $v_x(t_s)$, which causes $\delta_2(t_s)$. This stiction impulse, P_s, operates in a direction opposite to the collision impulse and the resultant impulse, $P_{A,x}$, may not satisfy the criterium for sliding, i.e., $|P_{A,x}| > \mu P_n$.

Correct physical models are enforced by determining the actual δ_c based on ⊶○ . However, the cumulative signal, $\delta_c(t_s)$ relies on $\delta_2(t_s)$ which is unknown at ⊶● switching. Therefore, a causal model requires no interference of δ_2 with δ_c which is achieved when $\delta_2 = 0$. This requires the signal that is involved to be continuous on the left-closed interval, $[t_s, \rightarrow>$ in time, which is called the principle of temporal evolution of state. In case of the example, the stiction impulse cannot become active

when the rod-tip has crossed some horizontal distance, but it has to be activated at the moment, t_s, the rod is inferred to start sliding, $\alpha_{21,a}$. Now its effect is taken into account along with $P_{x,c}$ and $P_{A,x}$ is derived properly.

To ensure $\delta_2(t_s) = 0$, the system specification can be analyzed in terms of h and g. The function h specifies the relations between signals and state variables and may compute switching signal values, $s_d = h_d(\dot{x}_d)$, using a derivative relation. To enforce continuity in left-closed time-intervals of s_d, both h_d and $\dot{x}_d$ have to be continuous between modes. Assume a transition sequence $\alpha_k \rightarrow \alpha_m$. The function h_d is continuous if it does not change between α_k and α_m. Across discrete changes, x_d is determined by $g_{\alpha_k}^{\alpha_m}$ and $\dot{x}_d$ is continuous if $x_d = g_{\alpha_k}^{\alpha_m}(x_d)$.[2] This results in the condition for continuity of signals s_d between $\alpha_k \rightarrow \alpha_m$

$$h_d^{\alpha_k} = h_d^{\alpha_m} \wedge x_d = g^{\alpha_m}(x_d) \tag{63}$$

If this conditions does not hold, either one of the following conditions has to hold for $\delta(t_s)$ to occur on $\circ\!\!\rightarrow\!\!\bullet$ and to satisfy temporal evolution of state:

- $\mathcal{F}_{\alpha_k} \setminus \mathcal{B}_{\alpha_k}$, the trajectory exits a patch on an open interval.
- $\mathcal{F}_{\alpha_m} \cap \mathcal{B}_{\alpha_m}$, the trajectory enters a patch on a closed interval.
- $\mathcal{P}_{\alpha_m}$, the trajectory enters a pinnacle.[3]

To verify temporal evolution of state for the colliding rod, first all mode changes for which $h_d^{\alpha_k} \neq h_d^{\alpha_m}$ are found. From $F_{A,x}$ and F_n in Eq. (48) this yields $\alpha_k = \alpha_{00} \wedge \alpha_m = \{\alpha_{01}, \alpha_{11}, \alpha_{21}\}$, and it has to be verified that $\alpha_{00} \rightarrow \alpha_m$ and $\alpha_m \rightarrow \alpha_{00}$ are $\circ\!\!\rightarrow\!\!\bullet$ transitions.

[2] It follows that $\dot{x}_d = 0$.

[3] Note that this implies that a sequence of pinnacles (as for example in a collision chain), which violates the interval-point paradigm, still satisfies temporal evolution of state.

Lemma VII.1 *$\alpha_{00} \to \alpha_m$ is a $\circ\!\!\rightarrow\!\!\bullet$ transition.*

Proof: For $\alpha_i \to \alpha_{00} \to \alpha_m, \alpha_i = \{\alpha_{01}, \alpha_{11}, \alpha_{21}\}, \alpha_m = \{\alpha_{01}, \alpha_{11}, \alpha_{21}\}$

$$\left.\begin{array}{l} \alpha_{00} : (g_{\alpha_i}^{\alpha_{00}}(x) = x) \Rightarrow \mathcal{F}_{\alpha_{00}} \\ \sigma_{contact} : (\gamma_{\alpha_{00}}^{\alpha_m} \leq 0) \Rightarrow \mathcal{F}_{\alpha_{00}} \setminus \mathcal{B}_{\alpha_{00}} \end{array}\right\} \Rightarrow \alpha_{00} \circ\!\!\rightarrow\!\!\bullet\ \alpha_m$$

■

Lemma VII.2 *$\alpha_m \to \alpha_{00}$ is a $\circ\!\!\rightarrow\!\!\bullet$ transition.*

Proof: For $\alpha_m \to \alpha_{00}$

$$\left.\begin{array}{l} \alpha_{00} : (g_{\alpha_m}^{\alpha_{00}}(x) = x) \Rightarrow \mathcal{F}_{\alpha_{00}} \\ \sigma_{free} : (\gamma_{\alpha_m}^{\alpha_{00}} \leq 0) \Rightarrow \mathcal{F}_{\alpha_{00}} \cap \mathcal{B}^{\alpha_{00}} \end{array}\right\} \Rightarrow \alpha_m \circ\!\!\rightarrow\!\!\bullet\ \alpha_{00}$$

■

Next, inspection of the bond graph determines that derivative causality arises when switching to mode α_{01} from either α_{11} or α_{21}, which implies that $x_d \neq g_{\alpha_m}^{\alpha_{01}}(x_d), \alpha_m = \{\alpha_{11}, \alpha_{21}\}$ with $x_d = v_x$, and it has to be verified that $\alpha_m \to \alpha_{01}$ are $\circ\!\!\rightarrow\!\!\bullet$ transitions.

Lemma VII.3 *$\alpha_m \to \alpha_{01}$ is a $\circ\!\!\rightarrow\!\!\bullet$ transition.*

Proof: From Eq. (46) γ generates σ_{slide} for $\alpha_m \to \alpha_{01} \to \alpha_i$ if $|F_{A,x}^+| - \mu F_n^+ > 0$, for $\alpha_i = \{\alpha_{00}, \alpha_{11}, \alpha_{21}\}, \alpha_m = \{\alpha_{11}, \alpha_{21}\}$. From Eq. (48) $F_{A,x}^+ = m\dot{v}_x^+$ and $F_n^+ = m\dot{v}_y^+ - ma_g$; for a mode change from $\alpha_{01} \to \alpha_i$, σ_{slide} is generated if

$$|m\delta(v_x^+ - v_x)| - \mu m\delta(v_y^+ - v_y) + \mu m a_g > 0.$$

For $x^+ = x = g_{\alpha_{01}}^{\alpha_i}$ this yields $\mu m a_g > 0$ with a_g the only negative constant, and, therefore, no immediate mode transition occurs. So,

$$\left.\begin{array}{l} \alpha_{01} : (\gamma_{\alpha_{01}}^{\alpha_i}(g_{\alpha_{01}}^{\alpha_i}(x), g_{\alpha_{01}}^{\alpha_i}(x))) > 0 \Rightarrow \mathcal{F}_{\alpha_{01}} \\ \sigma_{stuck} : (\gamma_{\alpha_m}^{\alpha_{01}}) \leq 0 \Rightarrow \mathcal{F}_{\alpha_{01}} \cap \mathcal{B}^{\alpha_{01}} \end{array}\right\} \Rightarrow \alpha_m \circ\!\!\rightarrow\!\!\bullet\ \alpha_{01}$$

■

In this case $\alpha_{01} \rightarrow \alpha_m$ need not be proved because $x_d = g^{\alpha_m}_{\alpha_{01}}(x_d)$ and $h^{\alpha_{01}}_d = h^{\alpha_m}_d$.

Divergence of Time

For the colliding rod, divergence of time is violated if the horizontal rod-tip velocity falls below v_{th} but its angle and length are such that $|F_{A,x}| > \mu F_n$ when the system moves into mode α_{01} where the rod is stuck. This would generate σ_{slide} but when sliding, $|v_{A,x}| \leq v_{th}$ and σ_{stuck} is generated. To eliminate this inconsistency, a modeling decision can be made to generate σ_{stuck} only if the forces in α_{01} are such that σ_{slide} is not generated. This requires the addition of a pre-condition $|F^{\alpha_{01}}_{A,x}| \leq \mu F^{\alpha_{01}}_n$ to σ_{stuck}, where $F^{\alpha_{01}}_{A,x}$ and $\mu F^{\alpha_{01}}_n$ are calculated from $h(g^{\alpha_{01}}(x))$.

In general, transition conditions are likely to be more complex with greater interaction among modes. In such situations, an exhaustive energy phase space analysis like in Chapter V can be applied [79, 82, 85].

Simulation of Hybrid System Models

Numerical simulation schemes like Euler and Runge-Kutta can be used for generating continuous system behavior. The flow graph in Fig. 65 illustrates that discrete events generated by γ trigger an event detection module to determine the switching time, t_s, within a margin of tolerance, ϵ. The continuous field, f_{α_k}, computes $x_{\alpha_k}(t_s^-)$, then real time is suspended, and the meta-level control model, ϕ, is activated. In some cases it may generate a sequence of discrete state transitions. The resulting model configuration is then established, and $x_{\alpha_k}(t_s^-)$ is transferred to this model configuration as $x_{\alpha_m}(t_s)$. In case of a pinnacle, further events are generated when the state vector is updated and the *a priori* switching values change. This may cause a

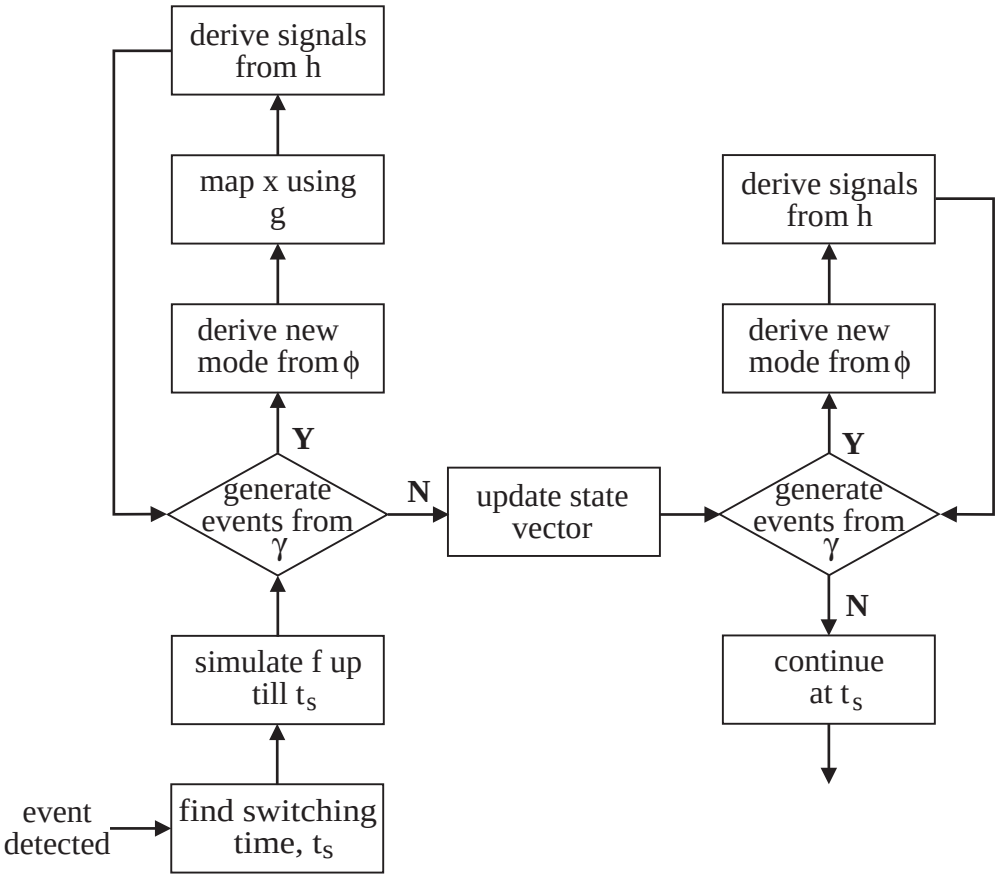


Figure 65: Flow diagram of hybrid system simulation.

new series of configuration changes that are executed by the discrete model using the different h functions in each mode. In the new continuous mode, α_n, f_{α_n} defines the simulation from time t_s with initial vector $x_{\alpha_m}(t_s)$ $(= x_{\alpha_n}(t_s^+))$. This implements simulation of f_{α_m} at t_s as a point in time and allows an energy redistribution at a point specified by algebraic equations. Note that in certain cases hybrid system simulation requires further sophistication [94], which is subject of future research.

Simulation of the Colliding Rod

To derive a numeric model of the continuous function, f_α, a 0-order, forward Euler approximation [130] is obtained by using $\dot{x} = \frac{x_{k+1}-x_k}{\Delta t}$, or $x_{k+1} = f\Delta t + x_k$.

Derivatives that are part of expressions in f are replaced analogously. For example, $\dot{v}_x = lcos\theta\dot{\omega}$ becomes $v_{x,k+1} = \frac{l(cos\theta_{k+1}\omega_{k+1} - cos\theta_k\omega_k)}{\Delta t}\Delta t + v_{x,k}$, or, $v_{x,k+1} = lcos\theta_{k+1}\omega_{k+1} - lcos\theta_k\omega_k + v_{x,k} = lcos\theta_{k+1}\omega_{k+1}$. The expressions for θ_{k+1} and $y_{M,k+1}$ are uniform across configurations. These equations combined with the rest of the numeric model constitute continuous behavior. Appendix A lists these models in full detail.

The only other function of the analytic specification that has to be represented by a numeric equivalent is h

$$h : \begin{cases} y_A = y_{M,k+1} - lsin\theta_{k+1} \\ v^+_{A,x} = v^+_{x,k+1} - lsin\theta_{k+1}\omega^+_{k+1} \\ p_{A,y} = m(v_{y,k} + lcos\theta_k\omega_k) \\ F^+_n = \begin{cases} 0 & \text{if } \alpha_{00} \\ m(\frac{v^+_{y,k+1} - v_{y,k}}{\Delta t} - a_g) & \text{otherwise} \end{cases} \\ F^+_{A,x} = \begin{cases} 0 & \text{if } \alpha_{00} \\ m\frac{v^+_{x,k+1} - v_{x,k}}{\Delta t} & \text{otherwise} \end{cases} \end{cases} \tag{64}$$

The derivative terms in h are simulated as Dirac pulses. So, as long as no discontinuous change occurs (e.g., $v^+_{y,k+1} = v_{y,k+1}$) the magnitudes of the forces are numerically estimated as shown. However, in case of a discontinuous change (e.g., $v^+_{y,k+1} \neq v_{y,k+1}$) the derivative term represents a Dirac pulse that has infinite magnitude, and, therefore, dominates the other terms. Due to numerical approximation, the pulse magnitude may be small compared to other terms (e.g., a_g) which interfere with the correct analytical solution. Therefore, in case of discontinuous change, the time derivative terms are treated as Dirac pulses and comparison is based on their areas.

The remainder of the analytical specifications has no temporal aspects and can be directly used for simulation. Note the difference between $p_{A,y}$, and $v_{A,x}$ which are

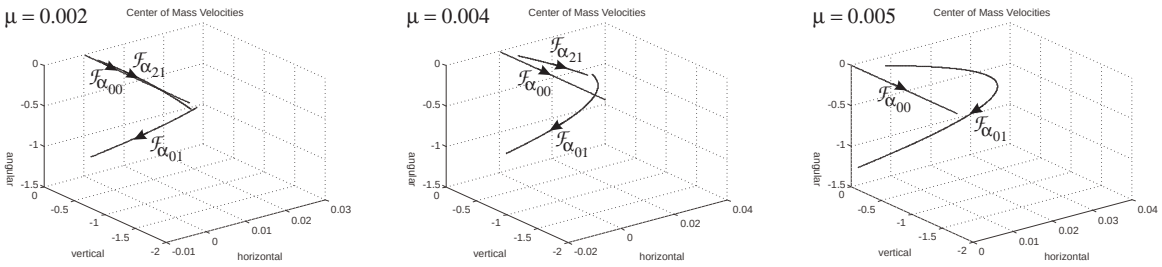


Figure 66: A number of trajectories in phase space of the colliding rod, $v_{th} = 0.0015, \theta = 0.862, l = -0.1, y_0 = 0.23$.

based on *a priori* and *a posteriori* values, respectively. Simulation results of three trajectories in phase space for different values of the friction coefficient, μ, are shown in Fig. 66. Initially, the system is in $(0, 0, 0)$, and flow $\mathcal{F}_{\alpha_{00}}$ determines that the vertical velocity of the rod increases its magnitude with time. When the rod-tip, point A, touches the floor the rod may start to slide, governed by flow $\mathcal{F}_{\alpha_{21}}$ ($\mu = 0.002$ and $\mu = 0.004$), or it may get stuck and behavior is governed by flow $\mathcal{F}_{\alpha_{01}}$ ($\mu = 0.005$). The discontinuous jumps between flows are illustrated in Fig. 66. Also, for simulations with $\mu = 0.002$ and $\mu = 0.004$, the sliding mode, α_{21}, emerges immediately after α_{00}. This is determined by performing a force balance in the mode, α_{01}, when the rod is stuck. However, the state vector is not modified as a result from this mode and it has no representation in phase space until the rod gets stuck after an interval of sliding.

When sliding, the center of mass accelerates in the horizontal direction, and the negative velocity at the rod-tip decreases. When it falls below a threshold value, the rod gets stuck and the system jumps to $\mathcal{F}_{\alpha_{01}}$. If the transition conditions were not properly specified, the force balance computations in the newly arrived stuck mode may imply that the rod should start sliding again. However, if it starts sliding, it does not have sufficient momentum to maintain a velocity larger than the threshold value. This would result in a loop of instantaneous mode changes which violates the

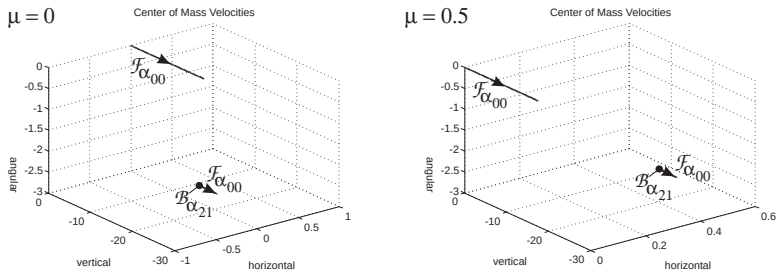


Figure 67: A boundary in phase space of the colliding rod, $v_{th} = 0.0015, \theta = 0.862, l = -10, y_0 = 23$.

principle of divergence of time.

If the rod length is increased, the rod may hit the floor and start to slide, but the moment the sliding motion starts, the balance of forces causes it to disconnect. Therefore, the rod is in the sliding mode of operation at a point in time only after which it moves into the mode of operation where it is free again, even though the collision is modeled as prefectly non-elastic, i.e., there is no restitution of momentum in any of the operational modes ($\epsilon = 0$). The corresponding phase spaces are shown in Fig. 67 and demonstrate how $\mathcal{B}_{\alpha_{21}}$ changes the state vector between both of the flows in α_{00}. Note that a field governs behavior in α_{21}, so the corresponding point in phase space is a boundary point rather than a pinnacle.

Summary

This chapter presented a complex, nonlinear, physical system and went through the process of systematically deriving its analytic and numeric models. The application of formal methodologies for model verification based on the principles of temporal evolution of state and divergence of time was illustrated. The rigor of the verification mechanism makes it suitable for algorithmic implementation but cumbersome to perform manually. At a more intuitive level, temporal evolution of state can be ensured

by modeling configuration changes where events that cause derivative causality are generated based on $\leq$ conditions instead of $<$. This ensures Dirac pulses are generated from well defined limit values. Divergence of time can be achieved by ensuring that CSPEC *on*/*off* and *off*/*on* transition conditions are mutually exclusive. This is easiest if they are based on *a priori* values since these do not change during mode changes.

CHAPTER VIII

MODEL BASED DIAGNOSIS

The complexity and sophistication of the new generation of aircraft, automobiles, satellites, chemical plants, and manufacturing lines, along with growing demands for their reliability and safety while keeping cost low, is being met by more automated control and monitoring systems, and the use of *functional redundancy* techniques for fault detection and isolation (FDI).

Introduction

Typically system models capture relations between measured variables and system or component parameters. Simulation and reasoning methodologies generate system behavior from these models, and when combined with techniques for identifying and analyzing observed deviations can be used to isolate a large number of possible faulty situations [8, 9, 39, 62, 101]. Some faults, such as a pipe blockage that completely isolates two parts of a system, change system structure and require a change in the system model itself. This renders these faults hard to diagnose unless failure mode models are explicitly incorporated into the analysis scheme [70]. This thesis studies FDI techniques that apply to complex dynamic systems that do not undergo such structural model changes. The focus of this research is on extracting discriminating information from *transients* in dynamic behavior caused by discontinuous parameter changes (faults). The aim is to quickly identify the root-causes for discrepancies in system behavior [75, 86].

Overview

FDI utilizes system models to predict operating values for a chosen set of system variables in a given mode of operation (Fig. 68) [41, 50]. This set of variables, called *observations*, is continuously monitored during normal operation.

Definition 2 (Observation) *An observation is a variable in the system model that is measured.*

Comparison of predicted operating values against observations help identify deviations from normal operation. Simple models may include a margin of error on sensors. When error thresholds are exceeded, the diagnosis system responds by setting corresponding alarms. In general, the diagnosis system maps observations, y, that deviate from predicted normal behavior, $\hat{y}$, onto a system model (Fig. 68). Analysis of descrepancies, r in the context of the model helps to *generate* one or more hypothesized *root-causes*, f, that explain the observed deviations. Hypothesized faults suggest modifications to the system models which are then employed to predict future system behavior. Continued monitoring and comparison with these predictions helps *refine* the initial fault set, f. Faults whose predictions remain consistent with the observations determine the root-causes for the observed problems. The goal is to continue the monitoring, comparison, and refinement process till the exact set of faults occurring in the system are isolated. The overall process of monitoring, generating hypothetical faults, prediction, and fault isolation using system models as the primary basis is referred to as *model based diagnosis*.

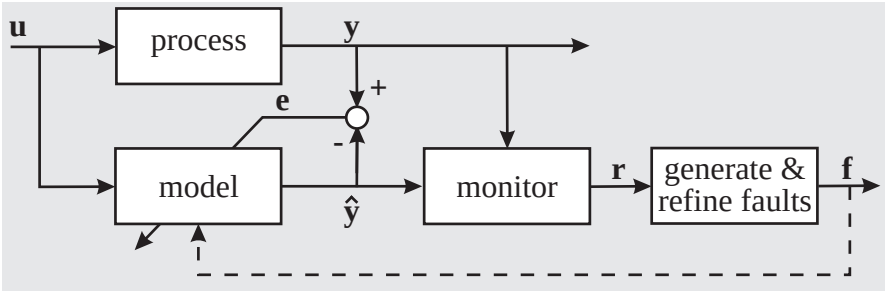


Figure 68: Diagnosis of dynamic systems.

The ultimate goal of model based diagnosis is to accurately[1] isolate problems and restore the system to normal operation by: (1) replacing faulty components, and (2) making control changes to bring system behavior back to desired operating ranges or quickly move the system into a *safe* mode of operation. This defines a paradigm that covers methodologies for fault detection, isolation, and recovery (FDIR) which requires the integration of methods for *prediction*, *monitoring*, and *measurement selection*. Prediction is a difficult problem even for experienced plant operators, therefore, it is all the more important to develop useful techniques to predict future behavior from given fault situations. From a computational viewpoint, the better the prediction, the easier it becomes for FDI algorithms to quickly prune the search space through continued monitoring and comparison with predictions. The monitoring stage is critical to successful FDI. Monitoring parameters like sampling rates govern the measurement interpretation process, and, therefore, fault hypothesis generation and refinement. The implementation of the monitoring algorithms determines whether faults may not be distinguishable from others, and this determines the overall diagnostic accuracy. Finally, a critical issue in FDI is sensor placement

[1]On accuracy and precision: Accuracy is the assessment of whether the actual value is contained by the estimate. Precision is the deviation of the estimate from the actual value.

and measurement selection. This is all tied into *diagnosability* analysis, i.e., selection of measurements that help isolate and differentiate among possible faults that may occur in the system [21, 71].

Nominal Values

The diagnosis scheme compares actual measurements with predicted nominal values of process variables that characterize normal operation. This comparison process is termed fault *detection*. In processes that operate in steady state, nominal values can often be retrieved from design specifications or documentation created by process engineers. To account for the effects of noise and measurement inaccuracies, a margin of error is added to the nominal values to increase robustness and avoid false alarms [111]. However, this decreases sensitivity, which is acceptable provided the delayed detection does not result in dramatic errors. For systems that typically operate in steady state modes, design documents often specify the upper and lower limits on nominal values of all system parameters and measured variables.

For systems whose normal operation modes include transients and dynamic behaviors, nominal values of process variables are harder to obtain. A fairly accurate process model is required to run in parallel with the process. Given the same initial state and the same input as the process the simulation mechanism should predict the process output in normal operation. In reality, approximations in the models and drift in the system may result in the estimated state vector slowly deviating from the actual system values. To prevent this, an observer mechanism shown in Fig. 69 can be used to estimate and make corrections to the estimated state vector. A critical issue when applying this scheme to obtain nominal values is the model adaptation

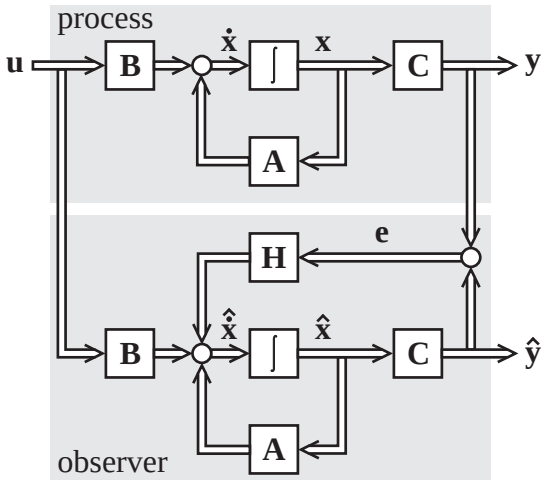


Figure 69: A general observer scheme.

rate, especially in case of incipient faults. If this rate is too fast, the model quickly adapts to changes in the system variables due to faults and generates nominal values that do not indicate a deviation.

Instead of just providing nominal values, the state estimation scheme can be used for diagnosis by reconstructing the entire set of states of the process if the process parameters have been estimated precisely. The reconstructed results are then compared and the set of most consistent states is chosen as the best estimate. This set can then be used to generate residuals based on the actual observations to detect whether a fault occurred.

To *identify* faults, the set of system equations are modified so that the three basic types of faults listed below can be explicitly identified as parameters and terms of the equations:

- Instrument faults; which refers to sensor faults.
- Actuator faults.

– Component faults; which refers to the different parts or sub-systems in the process where the fault can occur.

By explicitly incorporating these faults into the parameters of system equations as part of the observer system, diagnosis algorithms can be designed to detect and isolate faults. A unifying representation, in discrete form, is given by

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k + \mathbf{E}\mathbf{d}_k + \mathbf{K}\mathbf{f}_k \\ \mathbf{y}_k &= \mathbf{C}\mathbf{x}_k + \mathbf{F}\mathbf{d}_k + \mathbf{G}\mathbf{f}_k \end{aligned} \tag{65}$$

where $\mathbf{d}_k$ represents a disturbance term due to noise, and $\mathbf{f}_k$ represents the effects introduced by the fault term. Entries of $\mathbf{K}$ can be used to model actuator and component faults and entries of $\mathbf{G}$ can be used to model sensor faults [41].

A Comprehensive Diagnosis Scheme

State estimation requires parameter estimation to determine precise models of the process under scrutiny. Like state estimation schemes, diagnosis schemes can be based on parameter estimation techniques. The advantage of these schemes is the close relation between estimated values and physical coefficients. Given that nominal values can be derived from state estimation schemes or from design documents, comprehensive diagnosis can proceed by performing one or several of the following techniques (Fig. 70):

– Quantitative *parameter estimation* is derived from numerical models of the process. Typically filtering methods may be applied to estimate system parameters based on a vector of residuals. These parameters represent aggregate behavior of process components, and, therefore, a fault may cause a number of parameter

deviations. This process is computationally intensive, and may be subject to the convergence problems that occur in numerical estimation. To improve performance, other techniques can be used to narrow down the parameter search space.

- Performance of diagnosis algorithms can be enhanced by incorporating sophisticated *sensitivity analysis* schemes. The degree to which different faults affect measurements can be exploited by the parameter estimation procedure to rank possible causes by the sensitivity of the observables to the hypothesized causes.

- *Failure mode mappings* can further enhance the fault identification and isolation process. They represent a discrete event systems approach that requires knowledge about how process components may fail, and what effects these failures have on system parameters.

- *Dependency analysis* techniques rely on a topological functional model of the process and capture a weighted dependency between parameters and measured variables. These weights could be a function of various parameters, such as proximity to the observed fault. Sometimes the weights may capture process delay times, and in such cases the dependency graph represents a dynamic model of system behavior. Observed deviations can be traced back to parameter values which can be ordered in terms of when their effects become active.

Note that the techniques described above are not mutually exclusive. Any of these methods can be effectively developed into a diagnosis system. However, developing a diagnosis framework that integrates two or more of these approaches is likely to

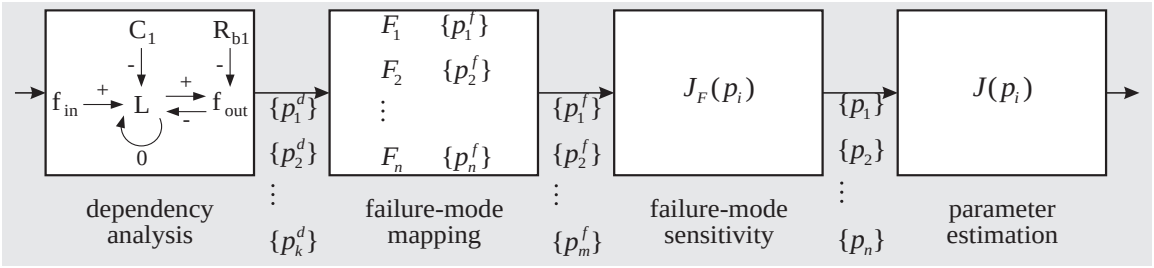


Figure 70: Complementary diagnosis schemes.

produce more efficient and robust systems. It remains a challenge to see how best to develop such systems.

The Model Based Diagnosis System

This thesis develops an approach for monitoring, prediction, and fault isolation that focuses on the use of dependency relations between parameters and observed variables. In previous work, static models were successfully applied to diagnosis tasks for moderately complex systems based on qualitative constraint equations [8] and the signed directed graph (SDG) [102]. However, due to the understrained models that are used, these approaches run into combinatorial problems. Furthermore, in case of the SDG, since system dynamics are not part of the model that is used for diagnosis, temporal feedback effects have to be re-introduced on an *ad hoc* basis [39, 101].

Modeling for Diagnosis

Successful modeling for diagnosis requires a unique set of requirements.

- The models should describe normal and faulty system behavior. The former provides the reference variable values for the monitoring task, and the latter forms the core for the prediction algorithm.

- The model should incorporate sufficient behavioral detail so deviations in observed variables can be mapped onto system components and parameters.

- The model should generate dynamic behavior, especially when faults cause transients that take the system away from normal steady state operation. Faults cause changes in system parameters, therefore, the assumption of constant parameters does not hold and their temporal effects have to be included.

- When faults occur, the system may undergo a structural change. Though structural changes is beyond the scope of this paper, they constitute an important category of failures. To make the presented framework extendable to incorporate this phenomenon, it is important to not preclude it from the onset.

In addition, to constrain the inherently exponential search space for diagnosis, it is important that the model impose all relevant physical constraints on the search process. Also, given the limits of purely qualitative and purely quantitative schemes that have been discussed elsewhere [41, 50, 102], models that generate and use both qualitative and quantitative information are preferred. This prevents loss of *a priori* information that may be useful for generating and further refining candidate sets.

Bond Graphs for Diagnosis

Bond graphs [107] provide a systematic framework for building consistent and well constrained models of dynamic physical systems across multiple domains with inherent causality constraints that provide effective and efficient mechanisms for diagnosis. An added advantage of the bond graph derived representation is their direct applicability to qualitative processing, which makes them applicable in situations where

precise numerical information may not be available. However, analytic system models derived from bond graphs are also amenable to quantitative simulation and analysis.

In this thesis, a causal dependency graph is derived from a bond graph to provide the system model. Links among system parameters and variables in a causal dependency graph are extended by temporal properties. Propagating effects of deviant observations to hypothesized causes (i.e., faults) can now be classified as instantaneous versus those that have delayed effects. Delayed effects can be further classified by the order of the effect, e.g., first order, second order, etc. The causal temporal models are derived from a bond graph model that adequately captures the dynamic characteristics of system behavior.

An important issue in diagnosing parameter deviations of physical systems concerns the transfer of the state vector to the failure mode after an abrupt fault occurs. Though changes in dissipative effects do not affect the state vector expressed in terms of power variables (e.g., pressure and velocity) of the independent energy storage elements (e.g., tanks, springs, and masses), abrupt changes in parameter values of energy storage elements may require an abrupt change of the state vector based on the principle of conservation of state (Chapter III). To illustrate, assume that at time t_f, a rock falls into an open tank with capacity C and outflow resistance R for a connected outlet pipe. The capacity of the tank decreases abruptly, and, therefore, the pressure, p, at the bottom changes instantaneously, however, the amount of liquid in the tank, q, is conserved. The effects of this parameter change on the state variable and describing field equation is shown in Fig. 71. The new pressure is directly derivable from the constituent equation $p = \frac{q}{C}$ but the effect is lost in the time-derivative form $f = C\frac{dp}{dt}$ which only shows a change of slope $\frac{dp}{dt} = \frac{f}{C}$ if C changes. If the time-varying

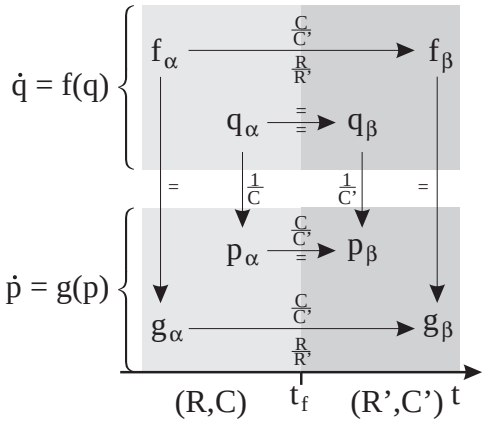


Figure 71: Transfer of the state vector and its field across discontinuous change.

effect of the C parameter is taken into account, the constituent equation becomes $f = C\frac{\partial p}{\partial t} + p\frac{\partial C}{\partial t}$ which determines the pressure as $\frac{\partial p}{\partial t} = \frac{f}{C} - \frac{p}{C}\frac{\partial C}{\partial t}$, and an abrupt change in C results in an instantaneous jump in $\frac{\partial C}{\partial t}$, which then causes an immediate abrupt change in p. Now the effect on the system state has become explicit. In the integral form of constituent equations, $p = \frac{q}{C} = \frac{1}{C}\int f dt$, used in bond graphs, this effect is automatically incorporated, since $p = \frac{1}{C}\int_{t_f}^{t} f dt + \frac{p_0(t_f)}{C}$.

To extend bond graph modeling for component oriented diagnosis requires establishing correspondence between individual components and bond graph elements. In the bond graph framework, primitive elements, such as resistors and capacitors represent mechanisms which may not always be in one-to-one correspondences with individual system components [10]. An individual component may have multiple aspects represented in the bond graph. For example, a component such as a pipe may be represented in the bond graph by its build-up of flow momentum (I) and resistance to flow (R). Biswas and Yu [10] describe a methodology for deriving bond graph models for diagnosis from a physical system description so that the bond graph

elements directly correspond to system components and mechanisms under diagnostic scrutiny. The modeling methodology has been further instantiated by Mosterman and Biswas [77, 79, 87]. In this thesis, a deviation of a bond graph parameter from its normal value is referred to as a fault.

Definition 3 (Fault) *A fault is a model parameter that deviates from its value in normal operation.*

Transient Based Diagnosis

To exploit process dynamics effectively for diagnosis requires that faults create transients in process behavior, which can be detected and their characteristics identified reliably by the monitoring process. Therefore, the approach is applicable to abrupt, possibly intermittent faults, but less suited to processing incipient faults. For example, the diagnosis methodology is well-suited to detecting sudden blockages in pipes which cause significant dynamic transients in pressure and flow values, whereas a pipe that accumulates dirt and slowly blocks may not manifest significant dynamic characteristics. It is more likely that the slowly blocking pipe will cause a gradual drift in the system steady state behavior.

Time Constants

Time constants play a key role in characterizing the dynamic behavior of physical systems. As discussed earlier, faults bring about instantaneous change in some system variables. For other variables, energy storage elements acting as buffers introduce propagation delays, thereby slowing down the rate of change in these variable values. In general, measurement variables with larger time constants in response to a

disturbance caused by a fault take longer to show significant changes in their values as compared to measurement variables that are associated with smaller time constants. If measurement snapshots are available from the system at rates that are faster than the smallest time constant, it becomes easier to track and relate system behavior back to primary fault causes. In this thesis, without much discussion, this is assumed to be true.

Assumption 1 (Time scale of observation) *The sampling rate for observations is faster than system time constants in both normal and faulty operation.*

Relations between hypothesized faults and measurements that do not embody temporal behavior propagate abrupt changes instantaneously. Physical systems are inherently continuous but these abrupt changes occur on a time scale that is much smaller than the time scale of observation, and, therefore, are observed to manifest as discontinuous changes. Therefore, abrupt changes are a sampling artifact attributed to the time scale of observation.

Definition 4 (Discontinuity) *A change in a signal value that happens on a time scale much smaller than the time scale of observation is considered to be abrupt and called a discontinuity.*

Observed transient effects in system behavior are often associated with multiple time constants which combine to define the overall delay. Combined effects of these behaviors is determined by the convolution rather than the sum of their partial effects [74]. To illustrate this, consider the two first order systems with time constants τ_1 and τ_2 in Fig. 72. The combined effect of these systems is given by $\tau_1 * \tau_2$ (convolution) whereas the sum of their individual delay times is shown by $\tau_1 + \tau_2$. Notice

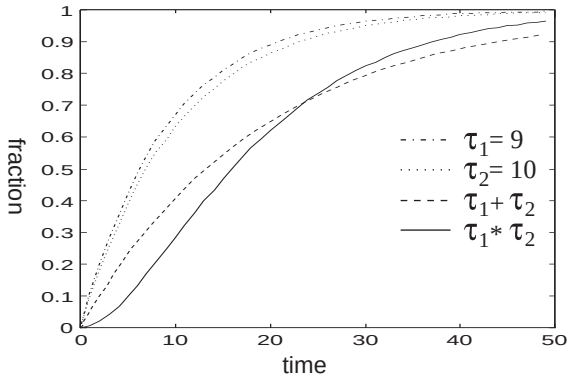


Figure 72: Delay times of two first order systems (τ_1 and τ_2), their sum ($\tau_1 + \tau_2$), and the actual delay time of their combined effects ($\tau_1 * \tau_2$).

that there is just one point in time where the two coincide which makes it difficult to derive combined delay times from individual values. The use of delay times is even further complicated given that designed time constants change when faults occur.

Even if one qualitatively analyzes effects, there can be uncertainty in temporal ordering of the observed deviations in two variables where one variable embodies a first order effect and the other a second order effect. Typically a measurement is considered *normal* if it is within a certain percentage (say 2%–5%) of its nominal value. Fig. 73 shows two variables, a first order effect, x_1, and a second order effect, x_2, and their delay times, t_{d1} and t_{d2}, respectively in crossing the error-threshold. At times between t_{d2} and t_{d1}, x_2 is reported deviant but x_1 is reported normal. Although x_2 embodies a second order effect with a 0 value 1^{st} order derivative at the point of failure, it crosses the error threshold before a first order effect. This is contrary to expectations where a first order effect is expected to dominate (i.e., be much faster than) a second order effect. Fig. 74 shows that the first order effect does have a faster effect when two signals have the same first order time constant, but one of them has a second order time constant too.

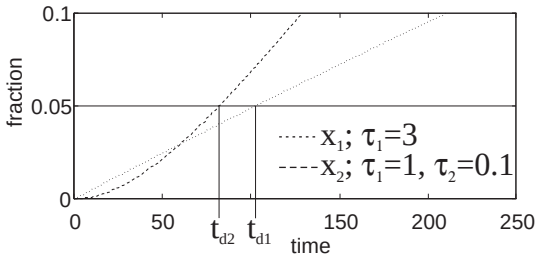


Figure 73: Delay times for observing deviations.

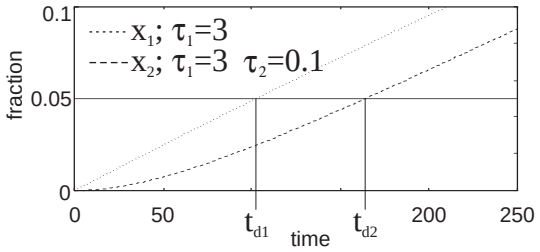


Figure 74: A second order effect with one constant equal to the time constant of a first order effect.

This brings up an important issue when dealing with normal values and deviation from normal in a qualitative reasoning framework for diagnosis of continuous-valued systems. A temporal ordering of first and higher order effects deviating from normal is in general impossible. Unless the sensor system is wired and calibrated with extreme care to guarantee a temporal ordering in response times, an observation being reported normal at a given time may really be an artifact of the implementation, and, therefore, cannot be used to refute faults. Moreover, it can, and in general will, lead to incorrect consistency based diagnoses.

In this thesis, deviant observations are individually analyzed to generate sets of single fault hypotheses. On the other hand, normal observations are not necessarily used to refute faults. This is because it is hard to differentiate between a truly normal signal versus one that is changing slowly, and at some point in time in the future

will deviate from normal. Only in situations where discontinuities can be reliably detected if they occur, can normal observations be used to refute faults which would have definitely caused a discontinuity for that observation. In this thesis, the use of normal observations is an optional parameter.

Feature Detection

Earlier discussion indicated that temporal ordering of measured deviations between signals is fragile and should be used cautiously in diagnosis. This makes individual signal features the prime discriminating factor between competing fault hypotheses. Prudence must be exercised in distilling information from signals, especially when they are noisy. Magnitude, or zero order, changes are typically measurable within a given error tolerance based on sensor characteristics. Filtering techniques help in deriving slopes, or first order derivatives, from measured signals, at least as a qualitative $\pm$ or no change value. However, deriving or measuring higher order derivatives can be quite unreliable. Dedicated transducers (e.g., accelerometers) may help measure 2^{nd} order derivatives, their use, in general, is often impractical. In this thesis, monitoring and feature detection focuses on magnitudes and slopes of individual measurements. The previous discussion indicates that in a qualitative framework magnitude values of relevance for diagnosis are *above/below* normal. Similarly, a slope within bounds cannot be considered to be 0, only when it is measured to have a significant value can it be identified as *positive/negative*.

The previous conjectures were made based on a minimalist basis. In general, tracking a signal will reveal many more characteristics, especially when dedicated feature detection algorithms are applied. As an example, consider a discontinuity detection

mechanism that considers a change discontinuous when it detects that magnitude and slope of an observed signal have opposing signs. This can be interpreted in a physical system perspective because discontinuous changes in physical systems due to parameter deviations are typically caused by abrupt changes in energy storage parameters. In steady state the stored energy in a system does not change, therefore, energy storage parameters have no effect on steady state behavior. After an initial discontinuous change, the system returns to its original point of operation. This discontinuity detection scheme has been successfully applied to the hydraulics domain.

Another general characteristic of most physical systems is that dissipation forces them to return to a steady state after a transient phase. This yields another feature that can be used for diagnostic analysis – determining whether the eventual steady state is detected as being above, below, or at the previous steady state value before the transient caused by the fault occurred. This results in three qualitative features that are detectable in actual measurements.

- Magnitude $\rightarrow$ discontinuously low, normal, discontinuously high.
- Slope $\rightarrow$ negative, positive.
- Steady state $\rightarrow$ below, at, above original.

For one observation this results in $3 \times 2 \times 3 = 18$ feature permutations which would maximally allow for the identification of 18 faults. If measurements of the process are such that normal observations and predictions can be used this improves to $3 \times 3 \times 3 = 27$ detectable faults, whereas if discontinuities cannot be reliably detected this degrades to $2 \times 2 \times 3 = 12$ detectable faults.

Summary

Abrupt faults cause dynamic system behavior and the resulting transients take system behavior from its nominal steady state of operation to a new steady state. Based on a model of the system dynamics, these transients can be effectively and efficiently applied to quickly isolate root-causes for deviating behavior. Magnitudes, slopes, and discontinuous changes at the time of failure for individual observations can be applied in a qualitative reasoning framework. Furthermore, the new steady state that the system achieves can be used as a final mechanism for fault isolation.

CHAPTER IX

IMPLEMENTATION

This thesis describes a diagnosis methodology illustrated in Fig. 75, which combines fault detection, fault hypothesis generation, prediction, and monitoring. This chapter describes how the bond graph model of the physical system[1] is used to derive a *temporal causal graph* which captures the dynamic characteristics of system behavior. Next, it presents diagnosis algorithms for each of the diagnosis modules (Fig. 75) that utilize the temporal causal graph. Observations need to be mapped onto a qualitative representation to detect discrepancies and generate faults hypotheses. Predictions of future behavior for each fault are then monitored against new observations to refine the set of possible faults

The Temporal Causal Graph

The temporal causal graph is derived in two steps [80, 92]:

[1] In the diagnosis models that are used, bond graph elements operate in so-called integral causality only [107].

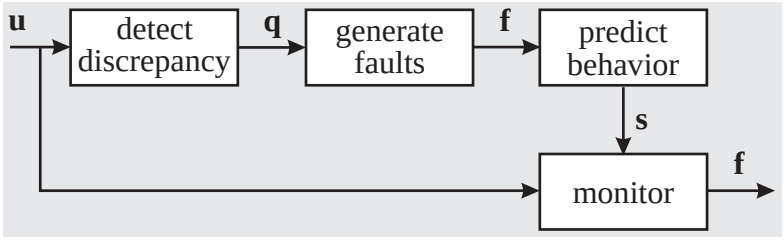


Figure 75: The diagnosis process.

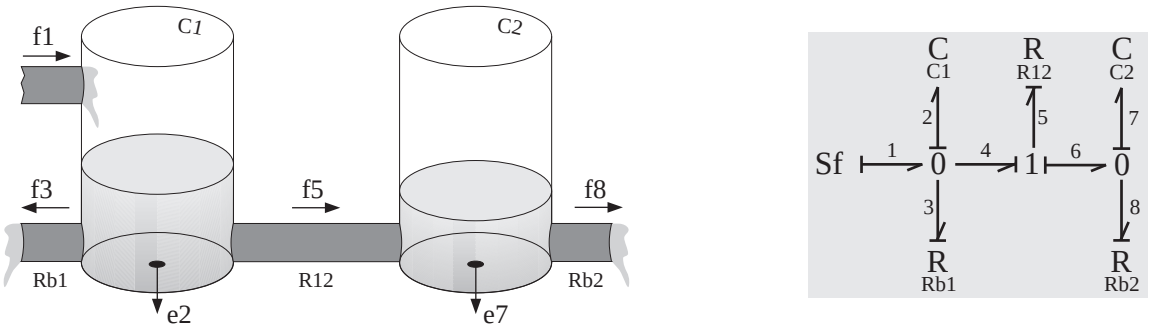


Figure 76: The bi-tank system and its causally augmented bond graph model.

1. The SCAP or its extended versions [129] are applied to generate a causal assignment among the power variables associated with the bond graph. These power variables are vertices in the temporal causal graph.

2. Components are linked one on one to individual edges in the temporal causal graph, and additional temporal and magnitude constraints are added to them.

The temporal causal graph for the bi-tank system in Fig. 76 (see also Chapter II) is shown in Fig. 77. The graphical structure represents effort and flow variables as vertices, and relations between the variables as directed edges. The relations can be attributed to junctions and system components. Junction relations add labels -1, 1, and $=$ to a graph edge. The $=$ implies that the junction constrains the two variable vertices associated with the edge to take on equal values, 1 implies a direct proportionality and -1 implies an inverse proportionality for the variable associated with the two incident vertices. When the edge is associated with a component, it represents the component's constituent relation. For example, for a resistor with flow causality, the edge between effort and flow is labeled $\frac{1}{R}$ and for a capacitor the edge is labeled $\frac{1}{C}dt$.

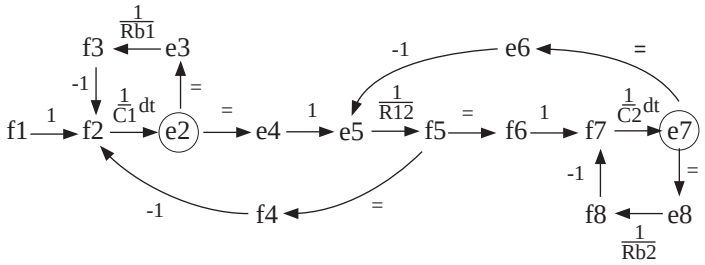


Figure 77: Temporal causal graph of the bi-tank system.

Junctions, transformers, and resistors introduce magnitude relations that are instantaneous, whereas capacitors and inductors also introduce temporal effects. In general, these temporal effects are *integrating*, and their associated rate of change is determined by the path that links an observed variable to the initial point where a deviation occurs. Note that the bond graph formalism presents one way to derive temporal causal graphs. Other modeling formalisms that support the physical modeling paradigm and allow for the generation of a temporal causal graph may be employed in its place. Also note the natural feedback mechanisms of dynamic physical systems that result in closed paths in the temporal causal graph. Between passive elements, these feedback mechanisms always have a negative gain [130] and if they include an integrating effect, as a result from a state variable in the system, these closed paths are referred to as state loops.[2]

Definition 5 (State loop) *A closed causal path with one and only one time-integrating effect is called a state loop.*

[2]In other diagnosis work, where temporal aspects of relations were not modeled, these ubiquitous negative feedback mechanisms caused difficulties in assigning a consistent graph mapping of deviations. Rather than trying to resolve this problem by breaking the causal paths somewhere, it can be used advantageously by incorporating the temporal phenomena.

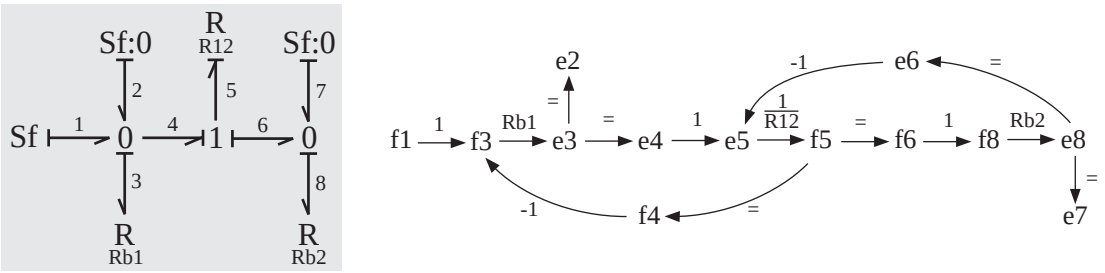


Figure 78: Steady state bond graph of the bi-tank system and its corresponding causal graph.

An added advantage of bond graph models is that they allow automatic derivation of the steady state model of the system. In case of the bi-tank system, both the tank capacities in steady state can be replaced by flow sources with value 0, since no change of stored energy takes place. The steady state bond graph and its resulting steady state causal graph are shown in Fig. 78. Notice that the causality links in the steady state graph differ from the causality links in the dynamic behavior graph (Fig. 77) and have less meaning. Since there is no temporal ordering, a steady state graph represents a set of algebraic equations rather than differential equations. Causality helps solve these equations, but its actual assignment is not critical. Independent of causality assignment in steady state graphs, the set of algebraic equations is invariant and equal effects of parameter deviations are generated for different causality assignments.

Component Parameter Implication

When a discrepancy between measurement and nominal value is detected, a backward propagation algorithm (Algorithm 2) is invoked on the temporal causal graph to implicate component parameters. Implicated component parameters are labeled $-$ (below normal) and $+$ (above normal). The algorithm propagates observed deviant

values backward along the directed edges of the temporal causal graph and consistent $-$ and $+$ deviation labels are assigned sequentially to vertices along the path if they do not have a previously assigned value. An example is shown in Fig. 79 for a deviant pressure, ${e_7}^+$, in the right tank of the two tank system (shown in Fig. 76). When e_7 is measured to be above its nominal value, backward propagation starts along $f_7 \stackrel{\frac{1}{C_2}dt}{\rightarrow} e_7^+$ and implicates C_2 as C_2^- (i.e., C_2 is below normal) or f_7 as f_7^+ (i.e., f_7 is above normal). Backward propagation from f_7^+, the proportional relation on $f_6 \stackrel{1}{\rightarrow} f_7^+$ implies f_6^+, and the inverse relation on $f_8 \stackrel{-1}{\rightarrow} f_7^+$ implies f_8^-. Propagation is terminated along a path when a conflicting assignment is reached.

Because backward propagation does not explicitly take temporal effects into account deviant values are propagated along edges with instantaneous relations first. This ensures that no faults due to higher order effects conflict with faults identified with lower order effects. An example is shown in Fig. 80. Following the path $e_1 \stackrel{1}{\leftarrow} e_2 \stackrel{dt}{\leftarrow} e_4 \stackrel{a}{\leftarrow} e_5$ backward, a^+ is generated based on the observation e_1^+. Note that the link $e_2 \stackrel{dt}{\leftarrow} e_4$ introduces a first order effect. However, the path $e_1 \stackrel{1}{\leftarrow} e_3 \stackrel{-1}{\leftarrow} e_4 \stackrel{a}{\leftarrow} e_5$ also includes e_1^+, which implies a^-, and this path has no temporal delays. From the time of failure, the $a^- \rightarrow e_1^+$ effect which is instantaneous will occur before the $a^+ \rightarrow e_1^+$ effect which has a first order delay.[3] Therefore, temporal effects need to be considered in implicating parameters and backward propagation is along instantaneous edges first. All component parameters along a propagation path are possible faults. As discussed in Chapter VIII, observed normal measurements do not terminate the backward propagation process. The end result of backward propagation

[3]Since this behavior pertains to the same signal, lower order effects always dominate during the transient stage.

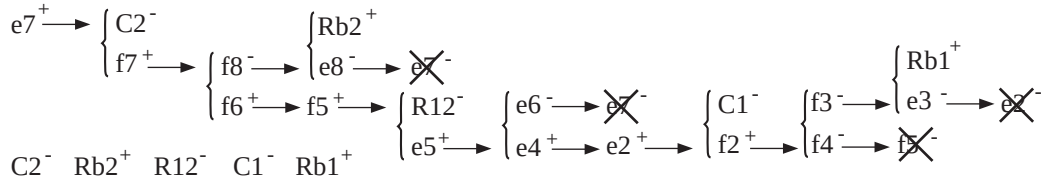


Figure 79: Backward propagation to find faults.

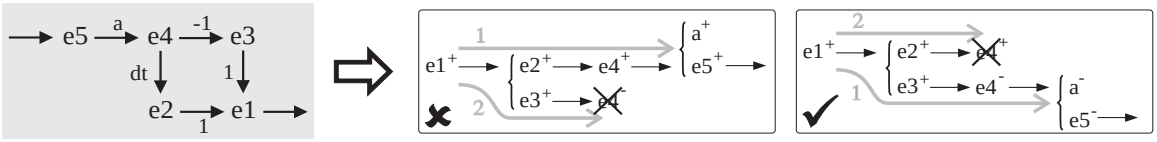


Figure 80: Instantaneous edges propagate first.

is a set of hypothesized single faults that are consistent with the reported deviant observations.

Algorithm 2 Identify Possible Faults

```
add observed vertex to list v_list
mark vertex with qualitative value
while v_list is not empty do
  v_current ← the last vertex in v_list
  while v_current has unsearched ancestors do
    if ancestor relation includes a parameter then
      add the relation to the set of faults
    end if
    if ancestor vertex is unmarked then
      ancestor value ← new_value(current value, relation)
      if relation is instantaneous then
        add the ancestor vertex to the beginning of v_list
      else
        add the ancestor vertex to the end of v_list
      end if
    end if
  end while
end while
```

Prediction

The next step is refinement of the hypothesized faults by prediction and monitoring. A complete version of the prediction module incorporates schemes for determining mode changes caused by abrupt structural faults and parameter value changes in the system. This may result in performing model switches before future behavior of the system can be predicted [77]. This thesis does not consider mode switches, and it is assumed that the system model remains valid after faults occur in the system.

Assumption 2 (No structural changes) *In case of faults, the system model does not undergo configuration changes.*

The main task of the prediction module in this case is to predict the dynamic, transient, behavior of the observed variables and also the eventual steady state behavior of the system under the fault conditions. Prediction of future behavior is in qualitative terms of temporal effects like magnitude (0^{th} order time-derivative), slope (1^{st} order time-derivative) and higher order effects.

Definition 6 (Signature) *The prediction of 0^{th}, 1^{st}, and higher order time-derivative effects of a system variable as a qualitative value: below normal (low), normal, and above normal (high) in response to a fault is called its signature.*

Forward Propagation

Prediction of future behavior is attained by forward propagation of the effects of parameter faults (Algorithm 3) to establish a qualitative value for all measured system variables. Forward propagation may occur along instantaneous and temporal edges. Temporal edges imply integration, therefore, the cause variable affects the derivative

of the effect variable. Initially, all deviation propagations are 0^{th} order, i.e., they are based on the magnitude of variable values. When an integrating edge is traversed, the magnitude change becomes a 1^{st} order change, i.e., the first derivative of the affected quantity changes, shown by an $\uparrow$ ($\downarrow$) in the propagation example in Fig. 81 for the temporal causal graph in Fig. 77. Similarly, a first order change propagating across an integrating edge creates a second order change (i.e., the second derivative of the affected variable), shown by $\uparrow\uparrow$ ($\downarrow\downarrow$) in Fig. 81. Second order changes propagate to third order changes, and so on.

Algorithm 3 Predict Future Behavior

```
add initial vertex to list v_list
mark vertex 0th order derivative with qualitative value
while v_list is not empty do
  v_current ← the last vertex in v_list
  while v_current has unsearched successors do
    if successor relation includes a time integral effect then
      increase current derivative order
    end if
    if derivative order ≤ maximum order then
      if successor derivative is no_mark then
        successor derivative value ← new_value(current value, relation)
      else if successor derivative has opposite value of current then
        successor derivative value ← conflict
      end if
      add the successor to end of v_list
    end if
  end while
end while
for all vertex derivatives do
  if value = no_mark and any higher order derivative ≠ no_mark then
    replace no_mark with normal
  end if
  if value = conflict then
    replace conflict with no_mark
  end if
end for
```

Forward propagation with increasing derivatives terminates when a signature of *sufficient* order is generated. The sufficient order of a signature is determined by a

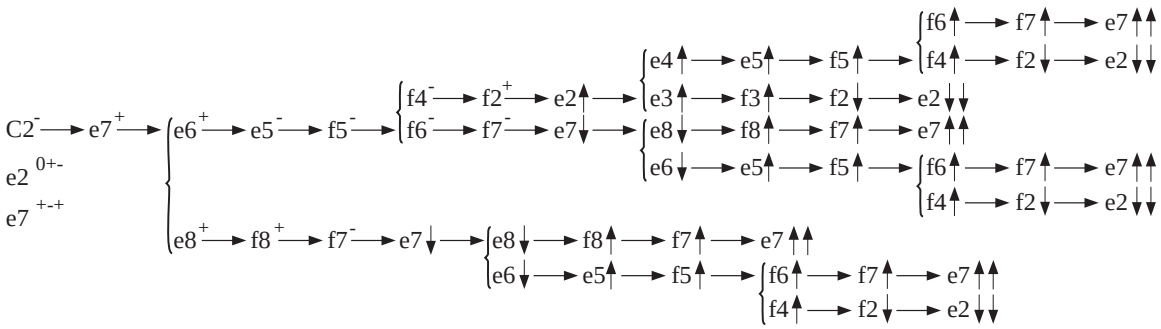


Figure 81: Forward propagation of the effect of an implicated component to establish its signature.

measurement selection algorithm and depends on the observed vertices in the temporal causal graph, and the desired precision for the diagnosis algorithm. A smaller number of total observations on a system typically imply that higher order signatures of the observed variables are required to achieve a level of precision that one could obtain with more observations but lower order signatures for each observation. Limiting the order of signatures has other practical advantages. As discussed earlier, higher order effects take longer to propagate, and, therefore, their effects are observed at later points in time from the time point at which failure occurs. This allows other phenomena and effects from other parts of the system to affect the observed variables and change their characteristics. For example, a fault may cause an abrupt increase in a variable value, but negative feedback effects may soon diminish the increasing value, and tend to bring the value down toward its previous steady state. This problem is compounded even more when cascading faults occur. They cause conflicts with lower order effects and hamper fault isolation.

A *complete* signature contains derivatives specified to its sufficient order. When the complete signature of an observed variable has a deviant value, monitoring will

eventually report a non normal value for this variable. When the signature is incomplete, the variable has an assigned deviation for higher order derivatives but the lower order derivatives are not assigned values. This implies that the lower order derivatives of the prediction for the fault under scrutiny are normal (i.e., non-deviating), and, therefore, are marked normal. Note that other faults could have caused deviations of lower order derivatives, and the conservative use of normal observations to refute candidates prevents elimination of the fault with predicted normal lower order derivatives.

Steady State

Signatures corresponding to an implicated component are used to track the transient system behavior. Eventually, most systems without catastrophic faults tend to come back to a steady state. The steady state causal graph derived from the bond graph model of the system then determines the final steady state value that each observed variable will achieve under the faulty conditions. Typically, the system returns to its previous steady state or converges to a new one. In the qualitative framework, steady state values are predicted to be *below the original*, *at the original*, or *above the original* steady state value for the variable. This predicted steady state value for each observed variable is attached to the signature and used in the monitoring stage.

Monitoring

The signatures of the observed variables generated in the prediction module are input to the monitoring module which compares actual observations, as they change dynamically after faults have occurred, to the reported signatures. It is only here that

a number of previously discussed practical issues are incorporated into the diagnosis engine. This allows localized mechanisms for measuring realistic dynamic effects that can be easily adjusted to experiment with several assumptions about the quality and characteristics of measurements and behavior. This section discusses a number of such localized measures that are employed to improve the robustness of the monitoring task.

Sensitivity to the Time Step

The time step employed in the monitoring process is a critical factor in establishing its success. The choice of the step size is dependent on the different rates of response that the system exhibits. Too small a time step may result in lack of sensitivity to changes, and too large a time step may produce incorrect inferences. Consider the signal (solid curve) shown at the left in Fig. 82. A large monitoring time step ($> t_1$) gives the appearance that this signal undergoes a discontinuous change (dashed curve). Decreasing the time step may help in differentiating between discontinuities (abrupt changes) and continuous effects. On the other hand, if the time step is too small when applied to a variable with a relatively slowly decreasing slope as shown at the right of Fig. 86, it appears that the signal does not change for a period of time, therefore, it is reported to be normal or to have reached steady state. In actuality it is decreasing, and reporting it as normal may result in premature elimination of true faults.

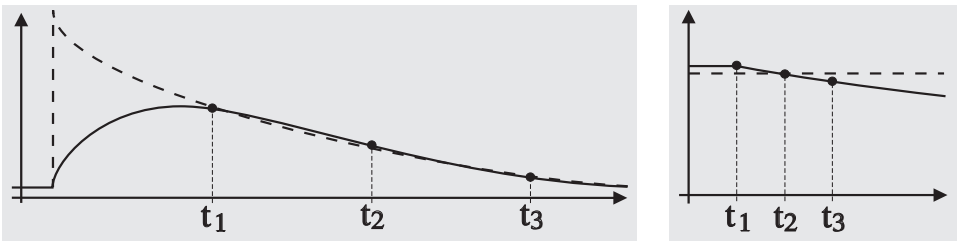
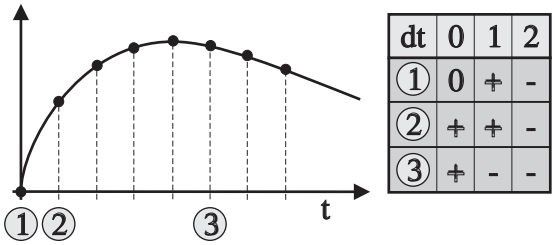


Figure 82: Signal interpretation.



dt	0	1	2
①	0	+	-
②	+	+	-
③	+	-	-

Figure 83: Progressive monitoring.

Progressive Monitoring

Transients that occur at the time of failure usually change over time as other effects from the system affect the observed variables. The signatures for a candidate fault can change dynamically. For example, a variable may have a 0^{th} order derivative which is normal and a 1^{st} order derivative which is above normal. Over time, the variable value will go above normal. Including the effect of higher order derivatives in the monitoring process is referred to as *progressive monitoring*. It replaces derivatives that do not match with the observed value with the value of derivatives of higher order in the signature. An example of this is shown in Fig. 83, where at time stamps marked 1, 2, and 3 a lower order effect is replaced by a higher order effect that has become manifest. If the higher order derivatives match the observed value, the fault under consideration is still plausible, otherwise it is rejected.

Progressive monitoring is activated when there is a discrepancy between a predicted value and a monitored value (this applies to 0^{th} and higher order derivatives). From the point in time when discrepancies occur between an observation and its prediction, the next higher derivative of the measurement is checked to see whether it could make the prediction consistent with the observation. If this next higher derivative value is predicted to be normal, the next higher derivative value is considered, and so on, till there is either a conflict between the prediction and observation, a confirmation, or an unknown value is found.

To illustrate, Fig. 84 shows the prediction-monitoring output for a sudden increase in outflow resistance R_{b2} in the bi-tank system in Fig. 76, where $-1, 0, 1$ maps onto *low, normal, high* and a period indicates the value is *unknown*. The two observed variables are the outflow of the left tank, f_3, and the pressure in the right tank, e_7. Each box depicts the monitored values at time steps when the set of hypothesized faults changes, the possible faults, and their signatures. The values on the top of each box represent the actual observations of the 0^{th} order, 1^{st} order, and 2^{nd} order behavior expressed in qualitative terms.[4] The lower section represents the prediction of the behavior of the observed variables for each of the candidates in terms of their 0^{th} order, 1^{st} order and 2^{nd} order derivatives, respectively. An example of the application of progressive monitoring is shown between step 9 and step 23 in Fig. 84. The signature for observation e_7 assuming fault R_{b1}^{+} changes from $0, 0, 1$ to $1, 1, 1$. This is based on the assumption that the 2^{nd} derivative, which is positive, makes an impact on both the 1^{st} derivative and magnitude of the signal. Updating the prediction in this

[4]Note that the actual observations only deal with magnitude and slope (1^{st} derivative). The 2^{nd} order derivative is never determined.

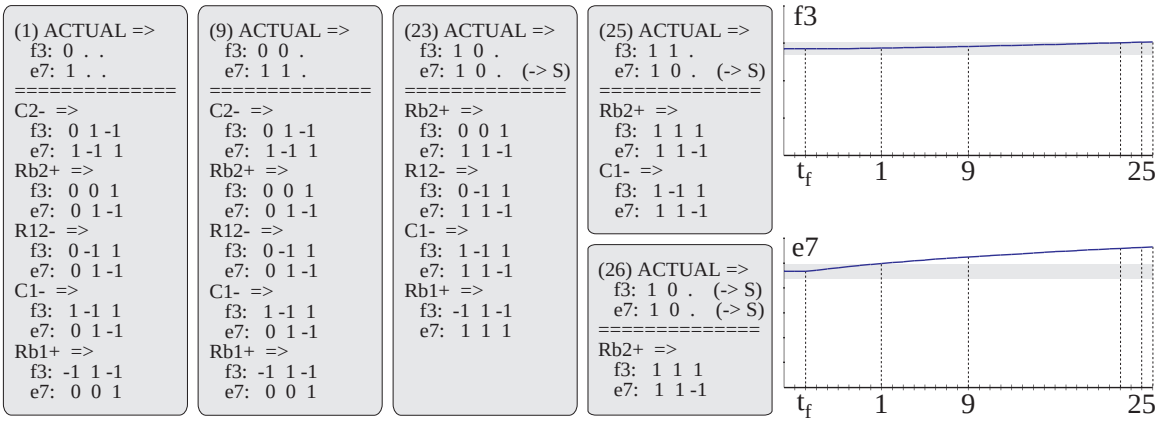


Figure 84: Progressive monitoring for fault R_{b2}^{+}.

manner keeps the signature consistent with the observation that e_7 is above normal. As a result, R_{b1}^{+} is still considered a viable fault hypothesis. Note that based on the qualitative observations, the time of failure, t_f, is unknown.

Fig. 85 shows the output when the fault C_2^{-} was introduced into the bi-tank system and discontinuity detection was used. In this case, since the actual deviation of the magnitude changes over time, a fourth value is added to the actual observations that captures whether a discontinuous deviation occurred at the time of failure as soon as it can be inferred. Now, progressive monitoring does not apply to the 0^{th} order prediction because this value is used to match discontinuous changes. Fig. 85 shows that the final diagnosis result was obtained in two time steps [80], and transient detection was suspended for both f_3 and e_7 at time step 7. The diagnosis engine can correctly detect and isolate all single fault parameter deviations if pressure in one tank and outflow of the other were measured and a first order signature is used. In this case, discontinuity detection is not required but steady state detection is. If steady state detection is not possible, either three observations and discontinuity detection are required or a second order signature without discontinuity detection can be used.

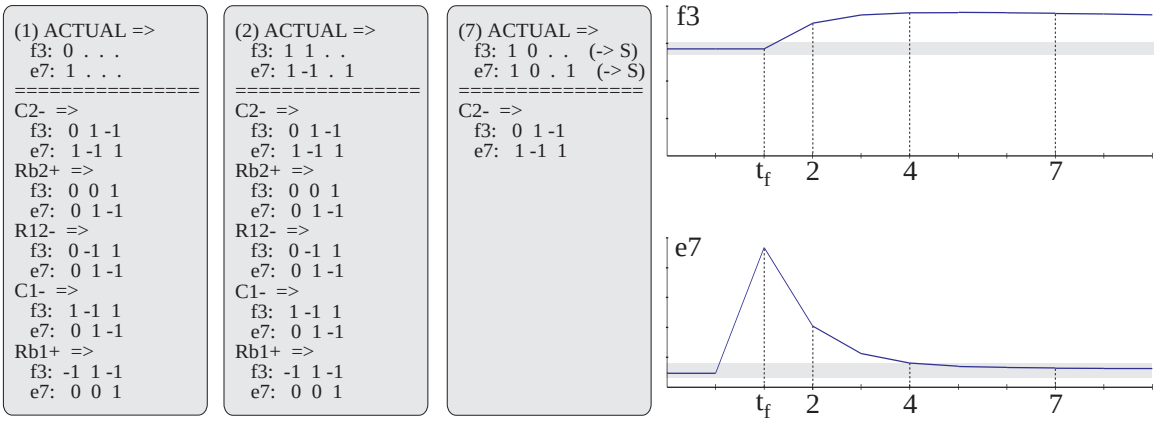


Figure 85: Results of the diagnosis system with C_2^- faulty and discontinuity detection is used.

Similar results were obtained on a three-tank system [80].

Fault Interaction

When the single fault assumption is relaxed multiple faults can occur simultaneously in the system. Since normal observations are not used to eliminate hypothesized faults, it is unlikely that real faults will be dropped during the hypothesis generation and prediction stages. However, when faults interact, and the faults have opposite predicted deviations on observations, the fault that has the less dominant effect is likely to be dropped from contention because its effects are masked out by the other fault. A pragmatic solution to this problem is to use an adequate number of observations so faults can be more easily discriminated. Moreover, for the observed variables, if the sufficient order of the signatures is kept low, the focus is more on the immediate effect of transients as opposed to situations where longer propagation delays are considered which increases the chances of interaction. Allowing the longer periods of interaction also allows for the possibility of cascading faults, where an initial fault

may change variable values sufficiently to cause other components to start failing.

For example, in the bi-tank system in Fig. 76, if R_{b1} fails with a high value, it may cause R_{b2} to fail with a below normal parameter value because the pressure e_7 may increase to large enough values. The observed pressures for these faults can become high or low, depending on which effect dominates. For multiple fault situations, R_{b1}^{+} and R_{b2}^{-}, and reported magnitude deviations e_2^{+} and e_7^{-} in the observations,[5] the diagnosis engine picks R_{b1}^{+} only if its predicted effect on e_7 is consistent with the observation e_7^{-}. The 2^{nd} order prediction for R_{b1}^{+} is $e_7^{0\ 0\ +}$, and this is in conflict with the measurement e_7^{-}, therefore, R_{b1}^{+} is refuted as a possible fault. However, if the sufficient order for the signatures were set to 1 (i.e., consider up to 1^{st} order effects only), the signature for R_{b1}^{+} becomes $e_7^{0\ ?}$, which is not inconsistent with the observations.

This demonstrates that a natural partitioning of a physical system can be created based on the order of its interactions. A measurement selection algorithm discussed in the next section, establishes diagnosability criteria for different parameters. For the bi-tank example, 1^{st} order signatures are sufficient for identifying all single faults when observing one of the pairs $\{e_2, f_8\}, \{f_3, f_5\}, \{f_3, e_7\}, \{f_3, f_8\}, \{f_5, f_8\}$. This gives a high single fault resolution and makes the diagnosis engine robust in a cascading fault environment.

Temporal Behavior

Two distinct characteristics of signals in response to fault disturbances, transients and steady state, carry the most distinctive discriminative information for diagnosis.

[5] This is based on the assumption that R_{b1} dominates e_2 and R_{b2} dominates e_7.

For monitoring it is important to know when, after a time of failure t_f, the transient detection phase terminates, and the system moves into the steady state mode, requiring steady state detection to be activated. Palowitch [102] reports that signals may exhibit a *compensatory* or an *inverse response* (Fig. 86). A compensatory response exhibits a decreasing slope and gradually moves towards steady state. For an inverse response, after an initial increase or decrease, the signal may reverse direction. An additional phenomenon resulting from abrupt faults can be categorized as a *reverse response*. A reverse response occurs if a discontinuous signal overshoots and, consequently, its qualitatively interpreted magnitude reverses sign (i.e., goes from above normal to below normal or *vice versa*). In the qualitative analysis framework, these behaviors are detected from an initial magnitude deviation by noting that:

- For an inverse response the magnitude and slope deviations have opposing signs and there was no discontinuous change of magnitude at t_f. If a discontinuous magnitude change took place, the transient at t_f could indicate a decrease of this magnitude, and this results in a slope of opposite sign. However, this is not an inverse response since the transient effects are the same as those exhibited at t_f and not affected by time at all.

- For a compensatory response the slope has become 0.

- For a reverse response the signal has a discontinuous initial magnitude deviation with sign that is opposite of the current magnitude deviation.

When any of these situations are detected, transient verification for that particular signal only is suspended (stage t in Fig. 86), and steady state detection activated (stage s in Fig. 86). Therefore, after a period of time, some signals may be processed

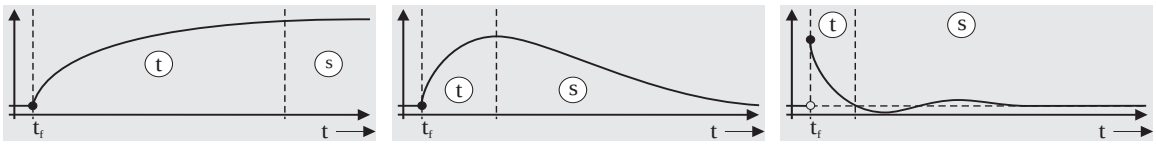


Figure 86: Typical signal transients in physical systems that exhibit different qualitative behavior over time.

in the transient mode, whereas others are processed in the steady state mode. Steady state is detected when a first order derivative becomes 0 for a sufficient period of time. The sufficient period of time is usually based on design information. As part of future research, more sophisticated steady state detection techniques will be investigated.

Algorithm 4 Monitor Predictions of 0^{th} Order Derivative

```
Require: predicted and observed 0th order derivative ≠ no_mark
  for all observed vertices do
    if use discontinuities then
      if observed discontinuity is consistent with prediction then
        assume fault is consistent
      else
        add to measure of inconsistency
      end if
    else
      predictions ← progressive_monitor(observed 0th order derivative)
    end if
    if observed 0th order derivative = prediction then
      assume the 0th order effect is consistent
    end if
    if not sensitive to normal observations and observed 0th order derivative is normal then
      assume the 0th order effect is consistent
    end if
  end for
```

Summary

To achieve robustness in analysis, the following measures are introduced:

1. A given signal's behavior is used to analyze faults only after the monitoring scheme reports it to be deviant. This circumvents the problem of insensitivity for small time steps.

Algorithm 5 Monitor Predictions of 1^{st} Order Derivative

Require: predicted and observed 1^{st} order derivative $\neq$ `no_mark`
  for all observed vertices **do**
    assume 1^{st} order effect is inconsistent
    if not sensitive to `normal` observations and observed 1^{st} order derivative is `normal` **then**
      assume the 1^{st} order effect is consistent
    else
      predictions $\leftarrow$ **progressive_monitor**(observed 1^{st} order derivative)
    end if
    if observed 1^{st} order derivative = prediction **then**
      assume the 1^{st} order effect is consistent
    end if
  end for

2. Comparison of a predicted signature with monitored observations is carried out for a number of time steps, and only if the results are inconsistent for the greater part of the comparisons is the fault rejected.

3. During this transient monitoring stage a progressive monitoring scheme defines the dynamic characteristics of the initial fault transients.

4. A sufficient order of the signatures allows for quick fault isolation before cascading faults occur.

5. After a period of time, signal behavior may deviate significantly from behavior at the time of failure (e.g., it may reverse its slope) and the transient prediction and verification process is suspended, and steady state analysis and verification is activated. This is based on three characteristic qualitative signal behaviors.

Measurement Selection

Measurement selection is a critical task in designing an economically viable and effective diagnosis system. For example, consider the one-tank system in Fig. 87. While measuring the pressure at the bottom of this tank, a discontinuous change of

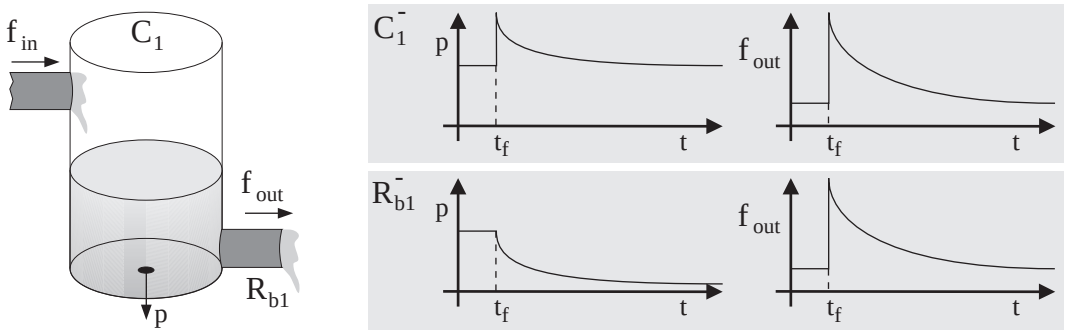


Figure 87: Characteristic response of pressure and outflow to two different faults.

this observable can only be caused by an abrupt change of tank capacity. Any other change affects the pressure only through integrating effects, which smooth out the abrupt change. Note that measuring the outflow would not yield this discriminative information since a discontinuous change in the outflow can be caused either by an abrupt change of tank capacity or by a change of outflow resistance. However, with multiple outflow resistances, no distinction can be made between the outflow resistance faults by observing just the tank pressures. On the other hand, with resistive failures steady state behavior of the outflow *does* differ. So, the flow measurement can be used for discrimination, if time permits.[6] This provides additional information associated with the outflow resistance to determine whether it has failed or not. So, additional discriminative information becomes available and the importance of *measurement selection* becomes apparent.

Before diagnosis, given a set of possible observations, a measurement selection algorithm is utilized off-line to determine sub-sets of observations that cover the parameter space completely. To this end, the signatures for all parameter deviations (high and low) are determined. Next, each combination of observations is checked to

[6]It is obvious that in case of certain failures it is preferred not to wait for the process to reach steady state but to take corrective measures or shut it down immediately.

determine if all parameter deviations have a unique set of characteristics. From this the minimal set of observations for a diagnosable system are determined.

Definition 7 (Diagnosability) *Diagnosability represents the sets of possible faults that can be distinguished from each other based on a given set of observations and signature order.*

Definition 8 (Complete diagnosability) *For a completely diagnosable system the sets of distinguishable faults all have one element only.*

To illustrate this algorithm, consider the bi-tank system in Fig. 76. Possible measurement points are selected to be the left and right outflow, the flow from the left to the right tank, and the pressure in both tanks. The parameters of the system are the left and right tank capacities and the flow resistances. First, magnitude changes in the observed variables are determined for positive deviations of all parameters, which yields:

$$\begin{bmatrix} f_3 \\ e_2 \\ f_5 \\ e_7 \\ f_8 \end{bmatrix} = \begin{bmatrix} - & 0 & 0 & - & 0 \\ \mathbf{0} & 0 & \mathbf{0} & - & 0 \\ 0 & - & 0 & - & + \\ \mathbf{0} & 0 & \mathbf{0} & 0 & - \\ 0 & 0 & - & 0 & - \end{bmatrix} \begin{bmatrix} R_{b1} \\ R_{12} \\ R_{b2} \\ C_1 \\ C_2 \end{bmatrix} \tag{66}$$

Similarly, slopes of all observable signals are determined for all positive parameter deviations:

$$\begin{bmatrix} \dot{f}_3 \\ \dot{e}_2 \\ \dot{f}_5 \\ \dot{e}_7 \\ \dot{f}_8 \end{bmatrix} = \begin{bmatrix} + & + & 0 & + & - \\ + & + & \mathbf{0} & + & - \\ + & + & - & + & - \\ \mathbf{0} & - & + & - & + \\ 0 & - & + & - & + \end{bmatrix} \begin{bmatrix} R_{b1} \\ R_{12} \\ R_{b2} \\ C_1 \\ C_2 \end{bmatrix} \tag{67}$$

Finally, steady state analysis completes the signature of each observable for all possible positive parameter deviations:

$$\begin{bmatrix} f_3^{\infty} \\ e_2^{\infty} \\ f_5^{\infty} \\ e_7^{\infty} \\ f_8^{\infty} \end{bmatrix} = \begin{bmatrix} - & + & + & 0 & 0 \\ + & + & + & 0 & 0 \\ + & - & - & 0 & 0 \\ + & - & + & 0 & 0 \\ + & - & - & 0 & 0 \end{bmatrix} \begin{bmatrix} R_{b1} \\ R_{12} \\ R_{b2} \\ C_1 \\ C_2 \end{bmatrix} \tag{68}$$

The effects of negative parameter deviations are computed analogously. In this system, these influences are the opposite of positive parameter deviations.[7] An exhaustive search is now applied to identify which set of minimal measurements produces unique feature characteristics for all parameter deviations. To this end, 0 value slopes are not used because of the ambiguity in detecting these. The resulting measurement sets are $\{e_2, f_8\}$,$\{f_3, f_5\}$,$\{f_3, e_7\}$,$\{f_3, f_8\}$, and $\{f_5, f_8\}$. Notice that $\{e_2, e_7\}$, which represents both pressures does not result in a completely diagnosable system. Inspecting the matrices, one learns that for these observations parameter deviations in R_{b1} and R_{b2} have the same magnitude and steady state deviations. Though the slopes differ between normal (bold faced) and a non-normal value, typically this can not be used to refute either of the two faults. It is interesting that these variables constitute the system state in a systems theory sense. So, simply observing the state of a system may not be the most efficient way of doing diagnosis.

[7] In general, this is not true, notably when structural changes occur.

CHAPTER X

THE LIQUID SODIUM COOLING SYSTEM

The effectiveness and use of the diagosis methodology based on transients was demonstrated in Chapter IX [80, 92, 93]. As a next step, to investigate the scalability of this methodology, its performance was tested on a model of a real system, the secondary liquid sodium cooling loop in a fast breeder nuclear reactor. The need for a qualitative approach in this system is motivated by the fact that it is a nonlinear complex system, modeled as sixth order. This makes it hard to develop accurate numeric models for generating system behavior in different modes. Moreover, the precision of the sodium flow sensors used in the system is limited and hardware redundancy is hard to achieve because of the expense involved in adding flow sensors.

The Secondary Sodium Cooling Loop

In a nuclear reactor, heat from the reactor core is transported to the turbine by a primary and secondary cooling system. The primary cooling sub-system connects directly to the reactor and transfers heat to the secondary cooling sub-system which then transfers heat carried by the liquid sodium to the steam in the generator (Fig. 88). Heat transfer from the primary cooling loop to the liquid sodium in the secondary loop happens through an intermediate heat exchanger to drain heat. The heated sodium is then pumped through two stages: the superheater and the evaporator vessel, both of which heat up the water and steam in the steam-water loop that then drives the turbine.

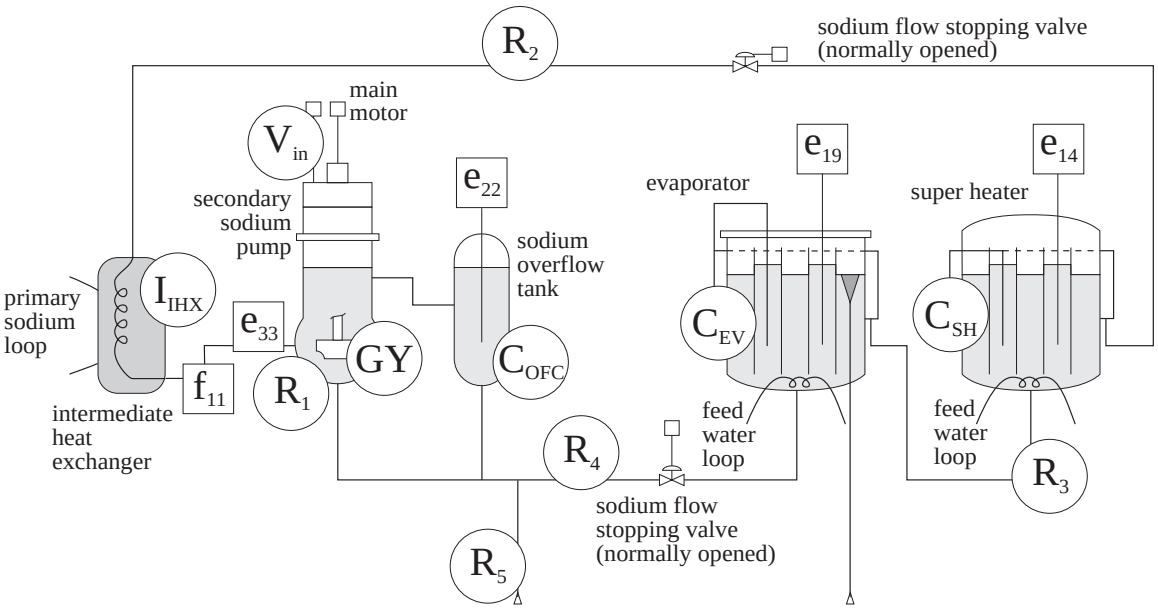


Figure 88: Secondary sodium cooling loop.

Modeling Hydraulics

When developing a model for the hydraulics of the system, it is important to note that bond graphs model *total pressure* rather than *static pressure*. Static pressure is the pressure of a fluid on the pipe wall and depends on the fluid flow velocity and elevation. Moving fluid contains kinetic energy, and combined with potential energy due to elevation from a reference level this yields total pressure. If ρ is the fluid density and a_g the gravitational acceleration, then the potential energy of a fluid volume, η, at height, h, is represented by $\rho\eta a_g h$. Given a fluid volume, η, flowing at velocity, v, its kinetic energy is $\frac{1}{2}\rho\eta v^2$. Combined with the hydraulic energy as a result from p_{static}, this equates the required energy to move a fluid volume η against its total pressure, ηp_{total}, and after dividing by η, total pressure is then expressed as [130]

$$p_{total} = p_{static} + \rho a_g h + \frac{1}{2}\rho v^2 \tag{69}$$

This is illustrated by Fig. 89, where in case of piping without resistance, the total pressure at each of the measurement points, p_0, p_1, p_2, p_3 is the same. However, the pressure measured at the pipe wall, depends on the velocity of the fluid flow at the opening of the manometer, and the height of the measurement point. The fluid velocity at the point where the manometer measures pressure p_0 is 0, therefore, this pressure is solely a function of the pressure in the tank. The manometer that measures p_1 is located at a lower level, and, therefore, it measures the pressure in the tank combined with the potential energy due to the difference in height, which results in a measured pressure that is higher. Pressure measurement p_2 is less than p_1, though at the same level, because it measures pressure of a fluid moving with a flow velocity, v_w, in the wide pipe segment. Pressure p_3 is even less because of the higher fluid velocity in the narrow pipe segment, v_n.

Note the higher fluid flow velocity in the narrow pipe segment. The flow variable in the hydraulics domain is the *volume* flow rather than flow *velocity*. Since the wide pipe transports the same volume of liquid in the same amount of time as the narrow pipe, because of its smaller area, the flow has a higher velocity in the narrow pipe. This represents an inertial effect because the fluid velocity is larger and thus, because of its mass, builds up relative momentum.

The Bond Graph Model

The model used for diagnosis applies energy and mass balance of the system in the hydraulics domain combined with the mechanical characteristics of the main motor and pump. The bond graph that captures system behavior in these domains is a nonlinear, sixth-order model (Fig. 90). The main motor driver (Fig. 91) is a

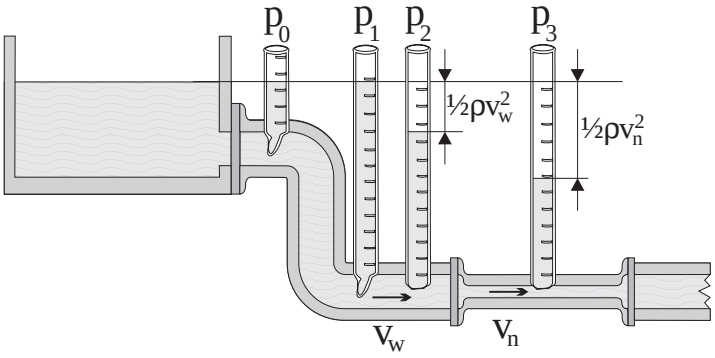


Figure 89: Total pressure versus static pressure.

synchronous, *ac* motor, and as an assumption, its electrical field is considered to be present as soon as it is turned on. Therefore, dynamic electrical effects are not modeled, and the electrical part of the motor system can be represented as a source of mechanical energy with a given torque/angular velocity characteristic. The inertia of the rotor and the mass of transmission gear is modeled by m_1, and the transmission ratio between motor and pump by n. Pump losses in the fluid connection between the motor and pump are modeled by a dissipation element, R_1, and the pump inertia is represented as m_2. The model of a centrifugal pump (Fig. 92), can be derived using conservation of power and momentum [130]. This generates:

$$\tau\theta = p_{out}\phi_{out}, \tag{70}$$

where τ is the input torque, θ represents angular velocity of the pump rotor, p_{out} is pump pressure, and ϕ_{out} the corresponding mass flow. Conservation of momentum states that the mechanical momentum, $\int \tau dt$, equals the hydraulic momentum. The amount of mass moved by the pump depends on the total area of its veins, a, minus the effective loss in moved mass due to the curvature of the veins, b. This is given as $\int a\theta - b\phi_{out}dt$. If the pump veins are not curved, $b = 0$. The hydraulic momentum of

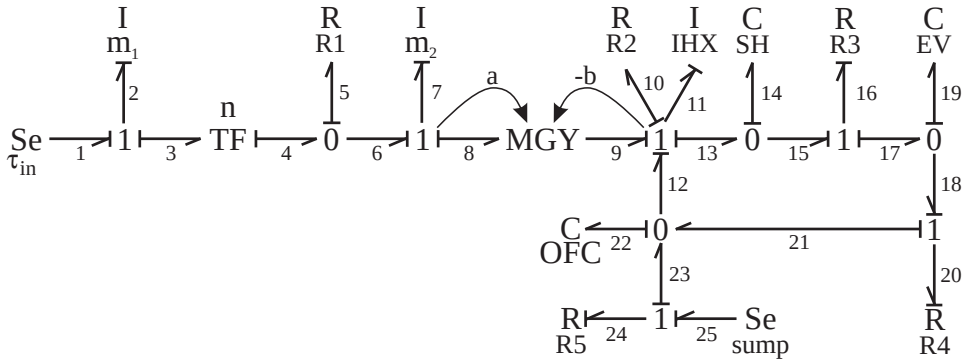


Figure 90: Bond graph of the secondary sodium cooling system.

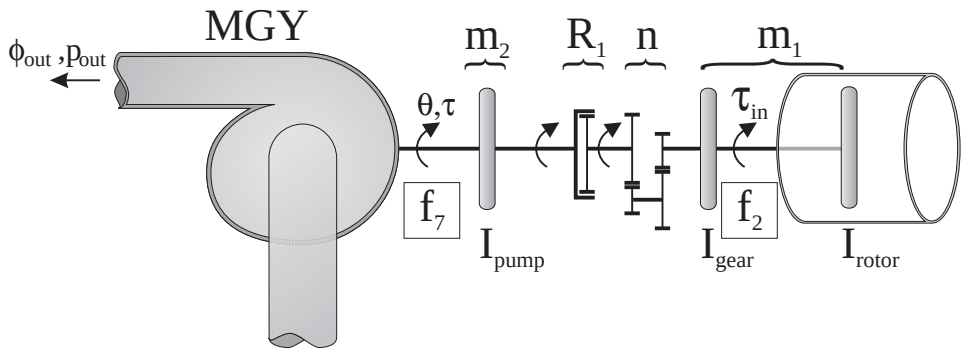


Figure 91: Synchronous *ac* motor that drives a pump.

the pump is represented by $\phi_{out} \int a\theta - b\phi_{out} dt$. The resulting equation

$$\int \tau dt = \phi_{out} \int a\theta - b\phi_{out} dt \tag{71}$$

can be written as

$$\tau = \phi_{out}(a\theta - b\phi_{out}) + \dot{\phi}_{out} \int a\theta - b\phi_{out} dt, \tag{72}$$

which for relatively low flow acceleration compared to flow velocity, yields the constituent relation $\tau = (a\theta - b\phi_{out})\phi_{out}$. Combined with Eq. (70) this yields $p_{out} = (a\theta - b\phi_{out})\theta$ which describes a modulated gyrator with modulus $a\theta - b\phi_{out}$.

The hydraulics of the sodium loop are modeled by a closed power loop (Fig. 90). The coil in the intermediate heat exchanger accounts for flow momentum build-up,

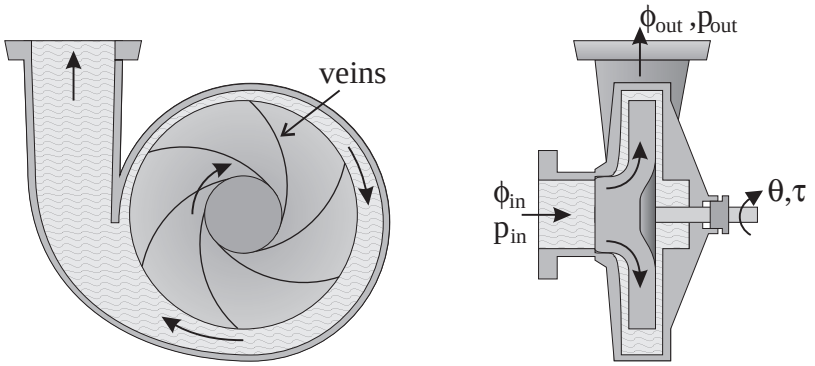


Figure 92: Operation of a centrifugal pump.

represented by a fluid inertia, I_{IHX}. The piping from the main pump through the heat exchanger to the evaporator vessel is represented by resistance R_2. The two sodium vessels are modeled by capacitances, C_{EV} and C_{SH} and the connecting pipe by its resistance, R_3. An overflow column, C_{OFC}, maintains a desired sodium level in the main motor, and the piping between the evaporator and this column is represented by resistance R_4. Both these storage facilities are connected to a sump, S_e, by a pipe with resistance, R_5.

The corresponding steady state bond graph is shown in Fig. 93. Solving the algebraic equations in this model results in a third order equation because of the quadratic modulus of the gyrator. A closed form symbolic solution was derived using Mathematica. This solution has one real root that represents the steady state solution, and symbolically provides the nominal values of the operation.

The Temporal Causal Graph

For diagnosis, the dynamic temporal causal graph of the system is derived from the bond graph (Fig. 94). Because of its nonlinear character, the MGY requires more detailed analysis. The derivation of the causal relations of the modulated gyrator

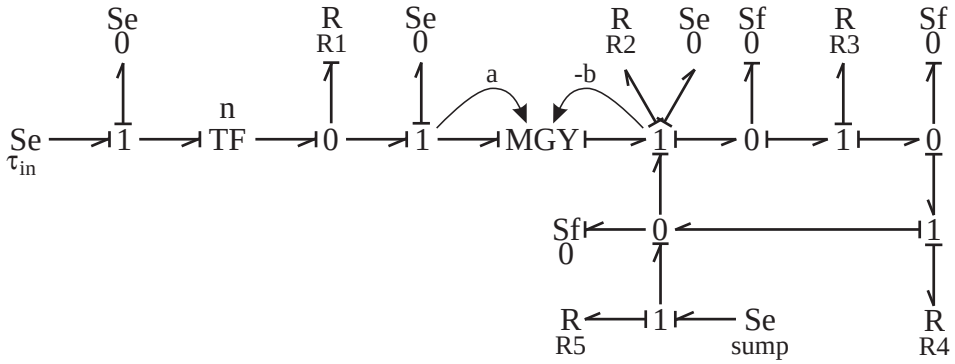


Figure 93: Steady state bond graph of the sodium cooling system.

is shown in Fig. 95. First it is observed that the modulation factor $g = af_8 - bf_9$ is directly proportional to f_8 and inversely proportional to f_9, $g(f_8, -f_9)$. The dependency of g on f_8 and $-f_9$ can be explicitly modeled by edges between these variables and the affected variables. In case of the dynamic behavior, the affected variables are e_8 and e_9 (Fig. 90) and the corresponding edges are added to the causal graph (Fig. 95). In the secondary sodium cooling loop $g > 0$, and, therefore, the added influences on e_8 result in ambiguity. This is revealed by a detailed study of the sensitivity of e_8 to f_9. From the bond graph $e_8 = (af_8 - bf_9)f_9$, so $e_8 = af_8f_9 - bf_9^2$. This relation, plotted in Fig. 96, reveals that the sensitivity of e_8 to f_9 can be positive or negative depending on the values of f_8 and f_9. Given the nominal values of the steady state operation of the system, which is parameter dependent, the weight of $f_9 \rightarrow e_8$ can be determined as a positive (1) or negative (-1) influence. However, once a deviation occurs, f_8 and f_9 may differ from their nominal values and a different operating point may be reached. Since these new values are likely to be caused by failures, and, therefore, will be unknown at monitoring time, the influence may reverse. This can only occur if e_8 is predicted to be high based on the proportional influence (-1 or 1). So, only a predicted decrease in e_8 is unambiguous, and, therefore,

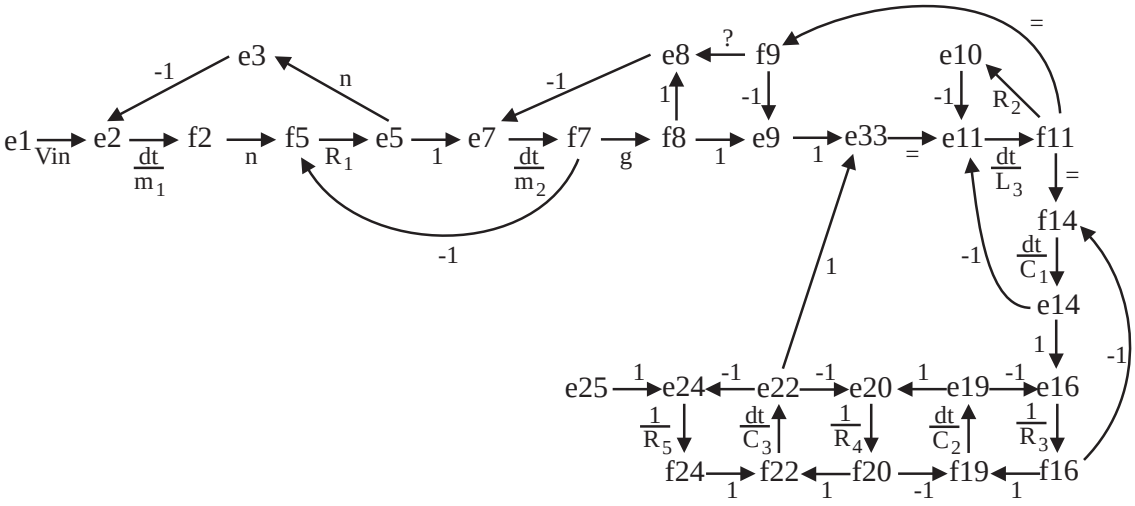


Figure 94: Temporal causal graph of dynamic behavior.

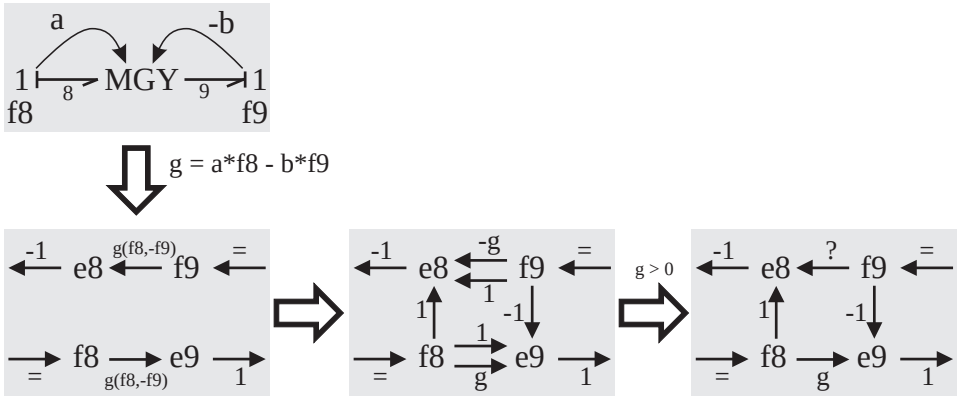


Figure 95: Temporal causal graph of a modulated gyrator.

propagated. A predicted increase in e_8 is propagated as *unknown*. Note that the two pump parameters a and b are represented by one positive parameter, g, in the temporal causal graph. This prohibits making a distinction between deviation of either, but since $\{a, b\}$ is attributed to the same component, the centrifugal pump, this set can be treated as one fault. Also note the variable e_{33} that represents $e_9 + e_{22}$. This variable is introduced because it is a measured value in the actual system.

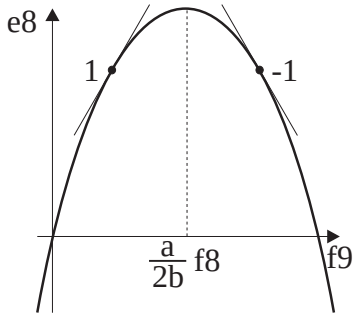


Figure 96: Detailed sensitivity analysis of $\frac{\partial e_8}{\partial f_9}$.

Measurement Selection

The measurement selection algorithm was applied to find a minimal set of observations required to establish a diagnosable system. In principle, measurement selection is based on the temporal structure of the graph, and, therefore, no numeric information is used. However, when f_9 or e_8 is low numeric information can be applied to determine whether the relation $f_9 \rightarrow e_8$ is 1 or -1 instead of generating *unknown* as is done when f_9 or e_8 is high. Furthermore, because of the critical nature of failures in this type of system, it is not desired to wait until the system has reached steady state after an abrupt failure. Therefore, measurement selection does not utilize differences in steady state behavior between faults.

Ideal Observations

Possible observations are selected to be the efforts on 0 junctions, $\{e_5, e_{14}, e_{19}, e_{22}\}$, and the flows on 1 junctions, $\{f_2, f_7, f_{11}, f_{16}, f_{20}, f_{24}\}$. From the temporal causal graph (Fig. 94) it can be seen that n and R_1 have similar temporal effects, i.e., they are not separated by temporal, integrating, effects. To distinguish between failure of these two parameters the difference in angular velocity of the driving motor and the

Table 3: Minimum sets of measurements for completely diagnosable single faults, in case of one pump parameter.

Order	**Measurements**												
	f_2	f_5	e_5	f_7	f_{11}	e_{14}	f_{16}	e_{19}	f_{20}	e_{22}	f_{24}	e_{33}	total
5	√	√				√		√				√	5
3	√	√					√				√	√	5
2	√	√	√		√	√			√			√	7
2	√	√	√		√		√	√				√	7
2	√	√	√		√		√		√			√	7
2	√	√	√		√		√			√		√	7
2	√	√	√		√		√				√	√	7
2	√	√	√			√	√				√	√	7
2	√	√	√			√			√		√	√	7
2	√	√			√		√		√		√	√	7

rotating pump, f_5, is added to the set of possible measurements. Finally, observations that are presently available in the secondary cooling system, indicated by the boxed variables in Fig. 88, show the pressure generated by the centrifugal pump, e_{33}, as an additional observation. Adding this measurement to the set of possible observations results in a completely diagnosable system for single faults (Table 3). If up till 5^{th} order is predicted sets of 5 observations can be used while for predictions of up till 2^{nd} order 7 observations are required. Note that f_7 is never used.

Actual Measurements

The ideal observations selected to achieve complete diagnosability of single faults are not necessarily the best measurements to choose in practice. This may be because some of the chosen sensors are expensive. As an alternative, measurements that are not part of the ideal set of observations may be available, for example, regulations may require certain pressures and flows to be monitored. In case of the secondary cooling system, variables in the system that are typically hard to measure are flow

Table 4: Fault discrimination based on actual measurement points in case of one pump parameter.

Measurements	Order	Indistinguishable Parameters
$\{f_2, f_7, f_{11}, e_{14}, e_{19}, e_{22}, e_{33}\}$	2	$\{n, R_1, R_4\}\{n, R_1, R_3, R_5\}$
	3	$\{n, R_1\}$
	4	$\{n, R_1\}$
	5	$\{n, R_1\}$
$\{f_2, f_7, f_{11}, e_{14}, e_{19}, e_{33}\}$	2	$\{n, R_1, R_4\}\{n, R_1, R_3, R_5\}\{g, C_3\}$
	3	$\{n^+, R_1^+, R_5^-\}\{n^-, R_1^-\}$
	4	$\{n, R_1\}$
	5	$\{n, R_1\}$
$\{f_2, f_7, e_{14}, e_{19}, e_{33}\}$	2	$\{n, R_1, R_2, R_3, R_4, R_5\}\{g, C_3\}$
	3	$\{n, R_1, R_3\}\{n^+, R_1^+, R_5^-\}$
	4	$\{n, R_1\}\{n^-, R_1^-, R_3^-\}$
	5	$\{n, R_1\}$

variables because flow sensors for liquid sodium are imprecise and expensive.

Using only the observations that are currently available, $\{f_2, f_7, f_{11}, e_{14}, e_{19}, e_{22}, e_{33}\}$, depicted by boxed variables in Fig. 88 and Fig. 91, single faults in the system are equally well diagnosable, except for $\{n, R_1\}$ (Table 4) which can also be achieved with a sub-set. The number of observations in this set and the ideal set of selected measurements are the same, but higher order predictions are required to achieve the same diagnosis for the currently used set. This implies there are more likely interactions in multiple, cascading, fault scenarios for the current observations.

Simulation Results

The numerical simulation model for the secondary cooling loop utilizes the forward Euler integration, $x_{k+1} = \dot{x}\Delta t + x_k$. To ensure stability, the numerical time step was chosen based on the smallest time constant of the model, τ_{min}, such that $\Delta t < \frac{\tau_{min}}{3}$ [130].

From system specification documents and by consulting domain experts, the parameter values listed in Table 5 were chosen. Those values are relative to each other (not the exact parameter values), and they suffice to generate behavior that matches the actual system behavior in a qualitative sense. The EV_{max} parameter indicates the maximum level of the liquid sodium in the evaporator vessel. This overflow mechanism was modeled but not taken into account in the temporal causal graph that was used for diagnosis because it would introduce a model configuration change. The simulation used a numerical time step of $\Delta t = 0.001$, which produced numerically stable simulation in all situations, normal and failure. This indicates a $\tau_{min} < 0.003$ in the model which relates to a minimal time constant in the actual system that is in the order of minutes. This time step is also chosen as the monitoring time step which, given a minimum time constant of one minute in the actual system, equates to a maximum sample rate of 20 seconds. Choosing a larger monitoring time step may result in incorrect diagnosis results. Smaller time steps imply added computations, but the monitoring algorithms are only sensitive to an upperbound on the step size, and this is likely to be the preferred alternative. Practically, the monitoring step should be chosen to be less than $\frac{\tau_{min}}{3}$.

Failure was simulated in the system by changing the model parameters by a factor 5. Conservation of state (Chapter III) was applied in case any of the capacitance or inductance elements failed, and keeping its stored momentum or liquid constant results in an abrupt change of angular velocity/flow or pressure, respectively. Simulation was stopped when either the transients of all observations were detected or 3913 samples had passed.[1]

[1]This number is derived from the time it takes a signal with time constant 10 to reach its steady

Table 5: Parameter values for the model of the secondary sodium cooling loop.

R_1	10	C_{SH}	20	m_1	0.1	n	0.25
R_2	1	C_{EV}	20	m_2	0.5	a	1
R_3	0.143	C_{OFC}	1.6	I_{IHX}	1	b	0.1
R_4	0.232						
R_5	1	τ_{in}	7	sump	2	EV_{max}	2.2

Fault Detection and Isolation Without Noise

The first set of experiments were conducted for the situation in which no noise was added to the measured signals, observations, generated from the simulation model. In this case, the quality of the results depended on (1) the parameter differences in the model and (2) unmodeled configuration changes. The results established a benchmark for subsequent experiments when noise was added to the observations. For detection of high and low values for signals, a qualitative margin of error of 2% was used in conformance with typical experiences in real situations to avoid spurious deviations due to noise. Of course, in case of no noise the margin of error could be further reduced, and this would have produced better diagnosis results because deviations would be detected sooner, and interactions with other phenomena would not corrupt the detected transients.

Table 6 summarizes the results. Columns 1 and 4 are the introduced faults, column 2 and 5 list the faults reported by the diagnosis system, and columns 3 and 6 indicate the number of measurement samples required to arrive at the diagnosis result. Three single faults were not accurately detected or isolated, R_3^-, R_4^+, and C_{EV}^-. Because of the overflow mechanism in the evaporator vessel, a decrease in capacity, C_{EV}^-, does not result in an increase in level and this is not detected. To detect this failure,

state value within 2%.

Table 6: Fault detection for $\{f_2, f_7, f_{11}, e_{14}, e_{19}, e_{22}, e_{33}\}$ with $\Delta t = 0.001$, $order = 3$, $q_{margin} = 2\%$.

Fault	Diagnosis	Samples	Fault	Diagnosis	Samples
R_1^+	n^+, R_1^+	58	R_1^-	n^-, R_1^-	43
R_2^+	R_2^+	27	R_2^-	R_2^-	46
R_3^+	R_3^+	1255	R_3^-	$\emptyset$ (R_3^-)	699 (699)
R_4^+	R_5^+ (R_4^+, R_5^+)	3429 (378)	R_4^-	R_4^-	43
R_5^+	$n^+, R_1^+, R_2^-, R_3^+, R_4^+, R_5^+$	2	R_5^-	R_3^-, R_4^-, R_5^-	687
C_{SH}^+	C_{SH}^+	73	C_{SH}^-	C_{SH}^-	16
C_{EV}^+	C_{EV}^+	45	C_{EV}^-	-	-
C_{OFC}^+	C_{OFC}^+	9	C_{OFC}^-	C_{OFC}^-	3
m_1^+	m_1^+	6	m_1^-	m_1^-	2
m_2^+	m_2^+	2	m_2^-	m_2^-	2
I_{IHX}^+	I_{IHX}^+	16	I_{IHX}^-	I_{IHX}^-	2

flow of sodium through the overflow mechanism has to be monitored. This type of configuration changes that are introduced by faults are not handled in this thesis. The two other faults, R_3^- and R_4^+, were detected but not correctly isolated, again because the overflow mechanism was not modeled in the temporal causal relations. If this phenomenon is included by tagging a predicted value *unknown* instead of *high* when it would have predicted an evaporator level that is *high*, the faults would be accurately isolated as indicated by the entries in parentheses in Table 6. Not all faults can be uniquely isolated ($\{n, R_1\}$) because of the lack of required measurements, or certain predicted deviations are too small to be observed, e.g., e_{22} in Fig. 97.

To gain precision, effects of order higher than 3 can be predicted but this may cause additional multiple fault interference. Alternatively, candidate generation can be modified to incorporate temporal effects as well. In its current implementation, one deviating signal is propagated throughout the graph irrespective of delay times

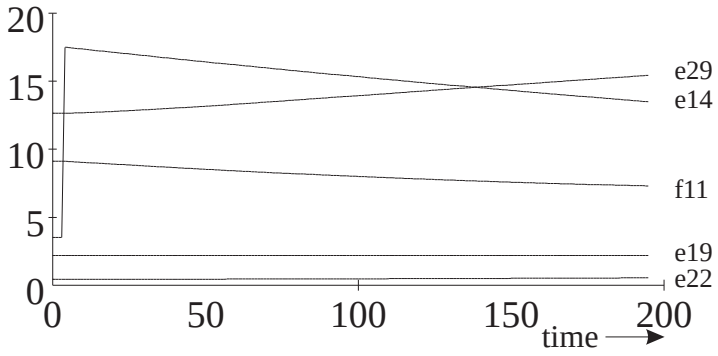


Figure 97: Characteristic responses for C_{SH}^{-}.

(integrating edges), whereas the prediction for each of the hypothesized faults is limited to a maximum order of derivative. So, an observed deviation may generate faults that cannot be monitored and refined unless other signals deviate noticeably, because their effect on the one initially deviating observation is of too high an order. To eliminate this effect, candidate generation can be limited by a total number of integrating edges traversed in the temporal causal graph as well. Preliminary experiments show that this implementation results in diagnosed faults for, e.g., $R_5^+ \rightarrow \{R_4^+, R_5^+\}$ in 2 steps, if only one integrating edge lies between the observed deviation and all generated faults. Note that, from a practical perspective, it is not required to pin down one root-cause exactly. A small set of likely candidates often suffices, as long as this set is accurate, i.e., it contains the true cause.

Fault Detection and Isolation With Noise

To investigate the effects of noise on the measured signals, 2% uniformly distributed measurement noise was added to the output values to model discretization noise due to analog to digital conversion (Fig. 98). Because of the uniform distribution, all signal deviations were within 2% of the actual value, but derivative

computations could accumulate errors as high as 4%, and the margin or band for *normal* behaviors was set at 5% to extract the qualitative signal values: low, normal, and high. The noise effects were well within the qualitative margin of error, and results similar to the ideal case were expected. Table 7 confirms this, and when compared to Table 6, indicates that the diagnosed faults were less precise but still accurate in the presence of discretization noise. In some simulation runs, the addition of noise resulted in more precise diagnosis because deviations that got masked in the original 2% qualitative margin were accentuated by the noise, and crossed the error threshold earlier. The only problem observed was with C^{+}_{OFC}. For this fault, the e_{33} signal stayed within the margin of error, and, therefore, was not reported as deviating (Fig. 99). Therefore, the fault was never detected. Increasing the magnitude of error in the fault caused sufficient deviation in e_{33}, thus producing the correct diagnosis even for the noisy signals. An interesting observation that can be made here is that different signals may have different sensitivities to possible faults. If they are not taken into account in setting the error thresholds, faults may be missed. Distilling abrupt changes to characterize transient behavior from noisy signals is also nontrivial. This is especially true when normal distributed noise is present, in which case there is always the probability of a measurement to be beyond the margin of error. This requires the addition of probabilistic measures to the diagnosis algorithms which will present faults ranked according to the probability of their occurrence. Furthermore, other methods [6] need to be tried to improve robustness of the monitoring system. These issues are part of future research.

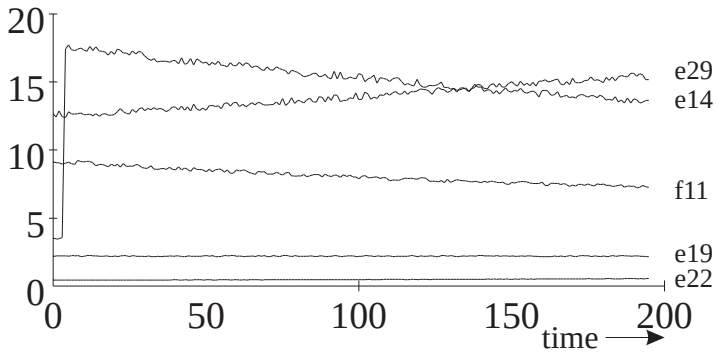


Figure 98: Characteristic responses for C_{SH}^{-} with 2% uniformly distributed measurement noise.

Table 7: Typical fault detection for $\{f_2, f_7, f_{11}, e_{14}, e_{19}, e_{22}, e_{33}\}$ with $\Delta t = 0.001$, $order = 3$, $q_{margin} = 5\%$, $noise = 2\%$.

Fault	**Diagnosis**	**Samples**	**Fault**	**Diagnosis**	**Samples**
R_1^+	n^+, R_1^+	84	R_1^-	n^-, R_1^-	38
R_2^+	R_2^+	38	R_2^-	R_2^-	68
R_3^+	R_3^+	2580	R_3^-	$\emptyset$ (R_3^-, R_5^+)	547 (543)
R_4^+	R_5^+ (R_4^+, R_5^+)	2514 (709)	R_4^-	R_4^-	86
R_5^+	$n^+, R_1^+, R_2^-, R_3^+, R_4^+, R_5^+$	2	R_5^-	$n^-, R_1^-, R_2^+, R_3^-, R_4^-, R_5^-$	2
C_{SH}^+	C_{SH}^+	128	C_{SH}^-	C_{SH}^-	21
C_{EV}^+	C_{EV}^+	80	C_{EV}^-	-	-
C_{OFC}^+	$\emptyset$	18	C_{OFC}^-	C_{OFC}^-	6
m_1^+	m_1^+	10	m_1^-	m_1^-	3
m_2^+	m_2^+	3	m_2^-	m_2^-	2
I_{IHX}^+	I_{IHX}^+	20	I_{IHX}^-	I_{IHX}^-	2

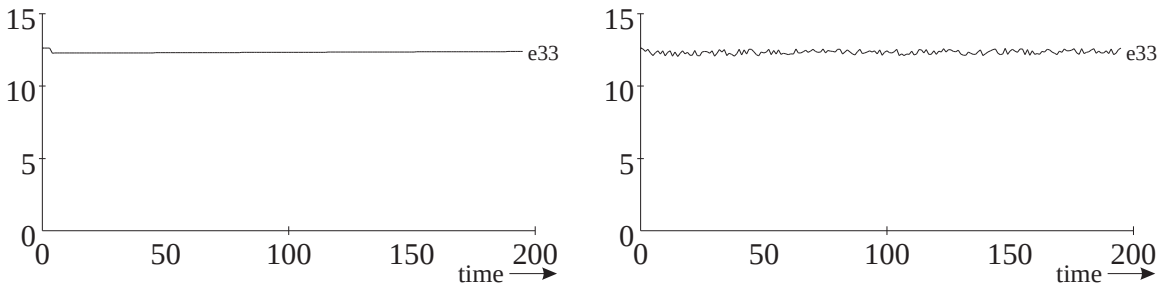


Figure 99: Response of e_{33} to C_{OFC}^+ with and without noise.

CHAPTER XI

CONCLUSIONS AND DISCUSSION

The primary contributions of this thesis cover two main areas: (1) modeling and analysis of hybrid physical systems, and (2) monitoring, prediction, and diagnosis of dynamic continuous systems.

Hybrid Modeling

Physical system behavior is governed by continuity of power, but to simplify models and reduce the computational complexity in analyses, models are abstracted to allow for discontinuous changes, which results in model behaviors being piecewise continuous with discontinuities occurring at points in time.

Summary and Conclusions

The first part of this thesis contributes to advances in hybrid modeling of physical systems.

1. A theory of discontinuities in physical system models is developed, and abstractions are classified as *time scale* and *parameter* abstractions. Two principles that govern physical system behavior across discontinuities:

   - *conservation of state*, and
   - *invariance of state*

   are formulated to ensure that hybrid models generate correct behaviors.

2. A systematic hybrid bond graph modeling paradigm is developed. Its primary components are:

   - An ideal switching element is added to the traditional bond graph elements to model discontinuous changes in system configurations.
   - A control structure based on the principles of conservation of state and invariance of state that is embodied by the Mythical Mode Algorithm is developed to ensure correct state transfer and to ensure that the principle of *divergence of time* is not violated.

3. A formal verification methodology is developed for the divergence of time principle based on

   - multiple energy phase space analysis, and
   - a model partitioning method based on areas of instantaneous propagation to keep the analysis computationally manageable.

4. A mathematical model for hybrid dynamical systems is derived which

   - relies on physical model semantics as embodied by the principles of conservation of state and invariance of state, and
   - formalizes the use of *a priori* and *a posteriori* values when discontinuous changes occur.

   The resultant principles developed in this work are:

   - *interval-point behavior*,
   - *temporal evolution of state*, and

- *generalized invariance of state.*

5. An implementation model for hybrid dynamical systems is presented, which facilitates

   - a classification of hybrid dynamical systems as *weak*, *mild*, and *strong*, and
   - model verification based on the principles of divergence of time and temporal evolution of state.

6. A mapping from hybrid bond graphs to the implementation model of hybrid dynamic systems is presented to support the derivation of model components based on a systematic modeling approach.

Discussion

Strömberg, Top, and Söderman introduced the notion of idealized switching elements for modeling discontinuous physical system behavior in the bond graph framework [117]. The work developed in this thesis extends this concept and demonstrates the need for control models to correctly handle global implications in the form of sequences of configuration changes and transfer of the energy state.

Iwasaki *et al.* [51] introduced hybrid systems in the artificial intelligence community by developing the notion of *hypertime* to advance time over infinitesimal intervals during discontinuous changes. This ensures divergence of time, but the semantics may result in incorrect behaviors when sequences of instantaneous configuration changes occur. In contrast, Henzinger *et al.* [47], Deshpande and Varaiya [31], and Guckenheimer and Johnson [44] introduce semantics where sequences of instantaneous

changes occur at one point in time, and all intermediate model configurations have a representation on the time-line. This thesis shows that this may generate incorrect state vectors in new model configurations after discontinuous changes. To solve this problem, this thesis introduces more specific model semantics based on physical system constraints, such as conservation of state, divergence of time, and temporal evolution of state to ensure the hybrid models generate physically correct behavior.

From a control perspective, Lennartson *et al.* [59] present a formal hybrid implementation model for embedded control systems. This thesis presents a more elaborate framework for defining model components and presents a general architecture for model building by using compositional methodologies. The difficult part of deriving the specifications for each of the model parts is addressed by a systematic mapping from a hybrid bond graph model onto this architecture.

Future research on hybrid dynamic systems will focus on developing methods to establish reachability, controllability, and observability in the mathematical modeling framework. Furthermore, a general purpose hybrid system modeling and simulation tool needs to be developed based on the hybrid bond graph modeling methodology. Special provisions for simulation of hybrid systems need to be developed and included [94]. Also, automated model verification based on divergence of time and temporal evolution of state has to be facilitated. Finally, collision chains, such as Newton's cradle present interesting behavior where sequences of pinnacles are traversed, which violates the interval-point principle. These phenomena need to be study in further detail.

Model Based Diagnosis

The basis of this work was that abrupt faults in physical systems cause discontinuities and transients in system behavior that can be exploited for developing efficient diagnosis strategies. Hybrid models of physical systems are exploited to develop efficient and robust diagnosis algorithms.

Summary and Conclusions

The diagnosis models conform to mild hybrid dynamic systems. Monitoring, prediction, and fault isolation methodologies are developed for diagnosis of dynamic continuous systems. The salient features of this work can be summarized as:

1. Hybrid bond graphs are shown to provide a suitable framework for qualitative, model based diagnosis.
   - They result in properly constrained models that incorporate conservation of energy and continuity of power principles during continuous operating regimes while adhering to the conservation of state and the invariance of state principles during discontinuous change.
   - A steady state model that follows the transient period caused by faults can be systematically derived as an extension of dynamic transient behavior.
   - Continuous system constraints are incorporated into the modeling methodology as topological properties, and this facilitates qualitative reasoning.
2. Temporal causal graphs of dynamic behavior that include time-derivative effects can be derived from bond graph models. Algorithms based on this graph are developed for

- generating a set of hypothesized faults that explain initially observed deviant behavior, and
- predicting future behavior for each of these faults.

3. A transient based diagnosis methodology and a monitoring algorithm are developed to:
   - Detect and verify behavior *signatures* that can be attributed to
     * discontinuities,
     * magnitude deviations,
     * slope deviations, and
     * steady state behavior.
   - Perform *progressive monitoring* to continue to follow observed measurements with predicted behaviors, sometimes invoking higher order derivatives to ensure a match between measurements and predictions. Hypothesized faults and their signatures are dropped only if the higher order derivatives cannot resolve conflicts.
   - A *feedback detection* mechanism to suspend feature detection in a timely manner. This is based on a classification of feedback behavior as:
     * *compensatory response*,
     * *inverse response*, and
     * *reverse response*.

4. A measurement selection algorithm based on an exhaustive search to determine the *sufficient* order of predicted behavior or the required measurement points

to achieve *diagnosability*.

Discussion

In the control community, numeric methods like state estimation and parameter estimation are typically used for fault detection and isolation [23, 41, 50]. These approaches are computationally intensive and require precise numerical information. In contrast, the artificial intelligence community has applied qualitative reasoning methodologies based on static system models [8, 33, 101]. However, transients in response to abrupt faults are dynamic and diagnosis based on static models only is underconstrained and results in combinatorial problems. Dynamic models exploit the typical fault characteristics of transients and allow for conservation of energy and continuity of power constraints to reduce the parameter search space. Moreover, it supports quick fault isolation, eliminating the problems due to multiple, cascading, faults.

Feedback in physical systems affects transients at the time of failure and higher order effects become manifest over time. A progressive monitoring scheme addresses these previously unsolved problems. In due time, transients are convoluted to the extent that qualitative characteristics at the time of failure cannot be reliably determined anymore. Three qualitative classes of behavior are categorized based on which transient detection is suspended and steady state detection activated.

Finally, this thesis shows the balance between the order of predictions and the number of measurement points in the system to obtain equivalent diagnostic precision.

The presented diagnosis engine has been tested in simulation and the next step involves its application to monitor the lubrication of a one-cylinder engine. This

will bring out a number of issues, the most important being noisy environments. The system was tested on uniform distributed noise, but normal distribution has distinctly different characteristics that require a probabilistic implementation of the deterministic algorithms to obtain robustness. Also, it needs to be investigated how the system behaves under modeling approximations.

Another important aspect is multiple independent faults (rather than cascading). In bond graphs, one component may be represented by several parameters, and, therefore, component failure is likely to affect a number of parameters which are independent on a bond graph level. Failure mode modeling appears to be a promising solution. This also applies to structural changes of the model which need to be investigated.

The presented approach relies on basic qualitative values high, normal, and low. However, often more precise information is available (at least as an order of magnitude estimate). It needs to be investigated how to systematically include this into the diagnosis engine.

Finally, measurement selection and diagnosability are very important notions in diagnosis, and require further attention. The presented measurement selection algorithm has proven helpful but has not utilized to its full potential. As a particular implementation, it is interesting to research how to select measurements on-line out of a large set of possible measurement points to quickly narrow down the search space without becoming computationally expensive.

Appendix A

THE FALLING ROD SPECIFICATION

The Analytic Model

$$X = \{\omega, v_x, v_y\} \tag{73}$$

$$U = \{F_f, ma_g\} \tag{74}$$

$$S = \{y_A, v_{A,x}, v_{A,y}, F_n, F_{A,x}\} \tag{75}$$

$$\Sigma = \{\sigma_{contact}, \sigma_{free}, \sigma_{slide}, \sigma_{stuck}, \sigma_{zero}, \sigma_{neg}, \sigma_{pos}\} \tag{76}$$

$$
f:\begin{cases}
\alpha_{00}:\begin{cases}\dot{\omega}=0\\ \dot{v}_x=0\\ \dot{v}_y=a_g\end{cases}\\
\alpha_{01}:\begin{cases}\dot{\omega}=\frac{-mlcos\theta}{J+ml^2}a_g\\ \dot{v}_x=lsin\theta\dot{\omega}\\ \dot{v}_y=-lcos\theta\dot{\omega}\end{cases}\\
\alpha_{11}:\begin{cases}\dot{\omega}=\frac{-mlcos\theta}{J+ml^2cos^2\theta}a_g\\ \dot{v}_x=0\\ \dot{v}_y=-lcos\theta\dot{\omega}\end{cases}\\
\alpha_{21}:\begin{cases}\dot{\omega}=\frac{-ml(cos\theta-\mu sin\theta)}{J+ml^2cos\theta(cos\theta-\mu sin\theta)}a_g\\ \dot{v}_x=-\mu(lcos\theta\dot{\omega}+a_g)\\ \dot{v}_y=-lcos\theta\dot{\omega}\end{cases}
\end{cases}
\tag{77}
$$

Table 8: State transition table specifying **C**.

ϕ_C	$\sigma_{contact}$	σ_{free}
0	1	
1		0

$$
g: \begin{cases}
\alpha_{00}: \begin{cases} \omega^+ = \omega \\ v_x^+ = v_x \\ v_y^+ = v_y \end{cases} \\
\alpha_{01}: \begin{cases} \omega^+ = \frac{\omega J - ml(cos\alpha v_y - sin\alpha v_x)}{J+ml^2} \\ v_x^+ = l\omega^+ sin\alpha \\ v_y^+ = -l\omega^+ cos\alpha \end{cases} \\
\alpha_{11}: \begin{cases} \omega^+ = \frac{\omega J - mlcos\alpha v_y}{J+ml^2 cos^2\alpha} \\ v_x^+ = v_x \\ v_y^+ = -l\omega^+ cos\alpha \end{cases} \\
\alpha_{21}: \begin{cases} \omega^+ = \frac{\omega J - ml(cos\alpha - \mu sin\alpha)v_y}{J+ml^2 cos\alpha(cos\alpha - \mu sin\alpha)} \\ v_x^+ = -\mu(l\omega^+ cos\alpha + v_y) + v_x \\ v_y^+ = -l\omega^+ cos\alpha \end{cases}
\end{cases}
\tag{78}
$$

$$
h: \begin{cases}
y_A = \int v_y dt - lsin\theta \\
v_{A,x} = v_x - l\omega sin\theta \\
p_{A,y} = m(v_y + l\omega cos\theta) \\
F_n = \begin{cases} 0 & \text{if } \alpha_{00} \\ m(\dot{v}_y - a_g) & \text{otherwise} \end{cases} \\
F_{A,x} = \begin{cases} 0 & \text{if } \alpha_{00} \\ m\dot{v}_x & \text{otherwise} \end{cases}
\end{cases}
\tag{79}
$$

Table 9: State transition table specifying **S**.

ϕ_S	σ_{free}	σ_{slide}	σ_{stuck}	σ_{zero}	σ_{neg}	σ_{pos}
0		1				
1	0		0		2	3
2	0		0	1		3
3	0		0	1	2	

$$\gamma : \begin{cases} y_A \leq 0 \wedge p_{A,y} \leq 0 & \Rightarrow \sigma_{contact} \\ F_n^+ \leq 0 & \Rightarrow \sigma_{free} \\ |F_{A,x}^+| - \mu F_n^+ > 0 & \Rightarrow \sigma_{slide} \\ |v_{A,x}^+| - v_{th} \leq 0 & \Rightarrow \sigma_{stuck} \\ v_{A,x}^+ = 0 & \Rightarrow \sigma_{zero} \\ v_{A,x}^+ < 0 & \Rightarrow \sigma_{neg} \\ v_{A,x}^+ > 0 & \Rightarrow \sigma_{pos} \end{cases} \tag{80}$$

$$
x_{k+1}:\begin{cases}
\theta_{k+1}=\theta_k+\omega_k\Delta t\\
y_{M,k+1}=y_{M,k}+v_{y,k}\Delta t\\
\alpha_{00}:\begin{cases}\omega_{k+1}=\omega_k\\ v_{x,k+1}=v_{x,k}\\ v_{y,k+1}=ma_g\Delta t+v_{y,k}\end{cases}\\
\alpha_{01}:\begin{cases}\omega_{k+1}=\frac{-cos\theta_k}{J+ml^2}a_g\Delta t+\omega_k\\ v_{x,k+1}=lsin\theta_{k+1}\omega_{k+1}\\ v_{y,k+1}=-lcos\theta_{k+1}\omega_{k+1}\end{cases}\\
\alpha_{21}:\begin{cases}\omega_{k+1}=\frac{-ml(cos\theta-\mu sin\theta)}{J+ml^2cos\theta_k(cos\theta_k-\mu sin\theta_k)}a_g\Delta t+\omega_k\\ v_{x,k+1}=-\mu(lcos\theta_{k+1}\omega_{k+1}-lcos\theta_k\omega_k+a_g\Delta t)+v_{x,k}\\ v_{y,k+1}=-lcos\theta_{k+1}\omega_{k+1}\end{cases}
\end{cases}
\tag{81}
$$

$$
h:\begin{cases}
y_A=y_{M,k+1}-lsin\theta_{k+1}\\
v^+_{A,x}=v^+_{x,k+1}-lsin\theta_{k+1}\omega^+_{k+1}\\
p_{A,y}=m(v_{y,k}+lcos\theta_k\omega_k)\\
F^+_n=\begin{cases}0 & \text{if } \alpha_{00}\\ m(\frac{v^+_{y,k+1}-v_{y,k}}{\Delta t}-a_g) & \text{otherwise}\end{cases}\\
F^+_{A,x}=\begin{cases}0 & \text{if } \alpha_{00}\\ m\frac{v^+_{x,k+1}-v_{x,k}}{\Delta t} & \text{otherwise}\end{cases}
\end{cases}
\tag{82}
$$

REFERENCES

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

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

[3] Rajeev Alur, Costas Courcoubetis, Thomas A. Henzinger, and Pei-Hsin Ho. Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems. In R.L. Grossman, A. Nerode, A.P. Ravn, and H. Rischel, editors, *Lecture Notes in Computer Science*, volume 736, pages 209–229. Springer-Verlag, 1993.

[4] Rajeev Alur and David L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126:183–235, 1994.

[5] Lee A. Barford. Discontinuities in phase space. E-mail communication, December 1995. Hewlett-Packard.

[6] M. Basseville and I.V. Nikiforov. *Detection of Abrupt Changes: Theory and Applications*. Prentice Hall, Englewood Cliffs, NJ, 1993.

[7] I.P. Bazarov. *Thermodynamics*. The MacMillan Company, New York, 1964.

[8] Gautam Biswas, Ravi Kapadia, and Xudong W. Yu. Combined qualitative quantitative steady state diagnosis of continuous-valued systems. *IEEE Transactions on Systems, Man, and Cybernetics*, 27(2):167–185, March 1997.

[9] Gautam Biswas, Stefanos Manganaris, and Xudong Yu. Extending component connection modeling for analyzing complex physical systems. *IEEE Expert*, 8(1):48–57, 1993.

[10] Gautam Biswas and Xudong Yu. A formal modeling scheme for continuous systems: Focus on diagnosis. In *Proceedings IJCAI-93*, pages 1474–1479, Chambery, France, August 1993.

[11] W. Borutzky. Exploiting differential algebraic system solvers in a novel simulation environment. *SAMS*, 17:165–178, 1995.

[12] A.M. Bos and M.J.L. Tiernego. Formula manipulation in the bond graph modelling and simulation of large mechanical systems. *Journal of the Franklin Institute*, 319(1/2):51–65, January/February 1985.

[13] Raymond M. Brach. *Mechanical Impact Dynamics.* John Wiley and Sons, New York, 1991.

[14] P.C. Breedveld. Multibond graph elements in physical systems theory. *Journal of the Franklin Institute*, 319(1/2):1–36, January/February 1985.

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

[16] Jan F. Broenink. *Computer Aided Physical Modeling and Simulation: A Bond Graph Approach.* PhD dissertation, University of Twente, Febodruk, Enschede, The Netherlands, 1990.

[17] Jan F. Broenink and Paul B.T. Weustink. A combined system simulator for mechatronic systems. In *Proceedings of ESM 96*, Budapest, Hungary, June 1996.

[18] Jan F. Broenink and Klaas C.J. Wijbrans. Describing discontinuities in bond graphs. In *Proceedings of the International Conference on Bond Graph Modeling*, pages 120–125, San Diego, California, 1993.

[19] Bernard Brogliato, Silviu-Iulian Niculescu, and Pascal Orhant. On the control of finite-dimensional mechanical systems with unilateral constraints. *IEEE Transactions on Automatic Control*, 42(2):200–215, February 1997.

[20] H.B. Callen. *Thermodynamics.* John Wiley & Sons, Inc., New York/London, 1960.

[21] Steve Chien, Richard Doyle, and Luiz Homem de Mello. Model-based reasoning for sensor placement. In *Proceedings of the AAAI-91 Workshop on Model-Based Reasoning*, Anaheim, CA, July 1991.

[22] Luca Chittaro, Giovanni Guida, Carlo Tasso, and Elio Toppano. Functional and teleological knowledge in the multimodeling approach for reasoning about physical systems a case study in diagnosis. *IEEE Transactions on Systems, Man, and Cybernetics*, 23(6):1718–1751, November-December 1993.

[23] R.N. Clark, P.M. Frank, and R.J. Patton. Introduction. In Ron Patton, Paul Frank, and Robert Clark (eds.), editors, *Fault Diagnosis in Dynamic Systems: Theory and Applications*, chapter 1, pages 1–19. Prentice-Hall, UK, 1989.

[24] Microsoft Corporation. Visual basic, version 3.0 professional edition. One Microsoft Way, Redmond, WA, 98052-6399, 1993.

[25] Stephen H. Crandall, Dean C. Karnopp, Jr. Edward F. Krutz, and David C. Pridmore-Brown. *Dynamics of Mechanical and Electromechanical Systems.* McGraw-Hill Inc., New York, 1968.

[26] Jim Davies and Steve Schneider. *An introduction to Timed CSP*. Oxford University Computing Laboratory, Oxford, UK, 1989.

[27] Theo J.A. de Vries, Peter C. Breedveld, and Piet Meindertsma. Polymorphic modeling of engineering systems. In *Proceedings of the International Conference on Bond Graph Modeling*, pages 17–22, San Diego, California, 1993.

[28] Theo J.A. de Vries, Arno P.J. Breunese, and Peter C. Breedveld. Max: a mechatronic model building environment. In John Sharpe and Vincent Oh, editors, *Computer Aided Conceptual Design*, pages 299–318, Lancaster, United Kingdom, April 1994. ISBN 0-901800-37-6.

[29] Tom DeMarco. *Structured Analysis and System Specification*. Prentice-Hall, Englewood Cliffs, New Jersey, 1979.

[30] Kenneth Denbigh. *The Principles of Chemical Equilibrium: With applications in chemistry and chemical engineering*. Cambridge University Press, London, New York, third edition, 1971.

[31] Akash Deshpande and Pravin Varaiya. Viable control of hybrid systems. In Panos Antsaklis, Wolf Kohn, Anil Nerode, and Shankar Sastry, editors, *Hybrid Systems II*, volume 999, pages 128–147. Springer-Verlag, 1995. Lecture Notes in Computer Science.

[32] Peter Dransfield and Jacek S. Stecki. Bond graph influenced research and learning in fluid power. *Journal of the Franklin Institute*, 328(5/6):535–545, 1991.

[33] Daniel Dvorak and Benjamin Kuipers. Model-based monitoring of dynamic systems. In Walter Hamscher, Luca Console, and Johan de Kleer (eds.), editors, *Readings in Model-Based Diagnosis*, pages 249–254. Morgan Kaufmann Publishers, Inc., San Mateo, CA, 1992. ISBN 1-55860-249-6.

[34] Hallvard Engja. Bond graph model of a reciprocating compressor. *Journal of the Franklin Institute*, 319(1/2):115–124, January/February 1985.

[35] Gottfried Falk and Wolfgang Ruppel. *Energie und Entropie: Eine Einführung in die Thermodynamik*. Springer-Verlag, Berlin, Heidelberg, New York, 1976. ISBN 3-540-07814-2.

[36] B. Falkenhainer and K. Forbus. Compositional modeling: Finding the right model for the job. *Artificial Intelligence*, 51:95–143, 1991.

[37] John B. Ferris and Jeffrey L. Stein. Development of proper models of hybrid systems: A bond graph formulation. In François E. Cellier and José J. Granada, editors, *1995 International Conference on Bond Graph Modeling and Simulation (ICBGM '95)*, number 1 in Simulation, pages 43–48, Las Vegas, January 1995. Society for Computer Simulation, Simulation Councils, Inc. Volume 27.

[38] Jenny Montbrun-Di Filippo, Marisol Delgado, Claude Brie, and Henry M. Paynter. A survey of bond graphs: Theory, applications and programs. *Journal of the Franklin Institute*, 328(5/6):565–606, 1991.

[39] Francis Eric Finch. *Automated fault diagnosis of chemical process plants using model-based reasoning.* PhD dissertation, Massachusetts Institute of Technology, August 1989.

[40] Kenneth D. Forbus. Qualitative process theory. *Artificial Intelligence*, 24:85–168, 1984.

[41] Paul Frank. Fault diagnosis: A survey and some new results. *Automatica: IFAC Journal*, 26(3):459–474, 1990.

[42] H.E. Garcia, A. Ray, and R.M. Edwards. A reconfigurable hybrid system and its application to power plant control. *IEEE Transactions on Control Systems Technology*, 3(2), June 1995.

[43] Fausto Giunchiglia and Toby Walsh. A theory of abstraction. *Artificial Intelligence*, 57:323–389, 1992.

[44] John Guckenheimer and Stewart Johnson. Planar hybrid systems. In Panos Antsaklis, Wolf Kohn, Anil Nerode, and Shankar Sastry, editors, *Hybrid Systems II*, volume 999, pages 202–225. Springer-Verlag, 1995. Lecture Notes in Computer Science.

[45] C.W. Ham, E.J. Crane, and W.L. Rogers. *Mechanics of Machinery.* McGraw-Hill Book Company, Inc, New York, New York, 4 edition, 1958.

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

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

[48] Tomas A. Henzinger and Pei-Hsin Ho. Model checking strategies for linear hybrid systems. Technical Report CSD-TR-94-1437, Cornell University, Computer Science Department, Cornell University, 1994.

[49] D. Hrovat and W.E. Tobler. Bond graph modeling and computer simulation of automotive torque converters. *Journal of the Franklin Institute*, 319(1/2):93–114, January/February 1985.

[50] Rolf Isermann. A review on detection and diagnosis illustrate that process faults can be detected when based on the estimation of unmeasurable process parameters and state variables. *Automatica: IFAC Journal*, 20(4):387–404, 1989.

[51] Yumi Iwasaki, Adam Farquhar, Vijay Saraswat, Daniel Bobrow, and Vineet Gupta. Modeling time in hybrid systems: How fast is "instantaneous"? In *1995 International Conference on Qualitative Reasoning*, pages 94–103, Amsterdam, May 1995. University of Amsterdam.

[52] Yumi Iwasaki and Herbert A. Simon. Causality in device behavior. *Artificial Intelligence*, 29:3–32, 1986.

[53] Dean Karnopp and Ronald C. Rosenberg. *System Dynamics: A Unified Approach*. The Massachusetts Institute of Technology, Cambridge, Massachusetts, 1968.

[54] Dean Karnopp and Ronald C. Rosenberg. *Analysis and Simulation of Multiport Systems*. John Wiley and Sons, New York, 1975. ISBN 0-471-45940.

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

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

[57] Benjamin Kuipers. Qualitative simulation. *Artificial Intelligence*, 29:289–338, 1986.

[58] J. Lefèvre. A didactic presentation of elementary bond graphs for non-engineering students. *Journal of the Franklin Institute*, 328(5/6):547–563, 1991.

[59] Bengt Lennartson, Michael Tittus, Bo Egardt, and Stefan Pettersson. Hybrid systems in process control. *IEEE Control Systems*, pages 45–56, October 1996.

[60] Bogdan Lent. *Dataflow Architecture for Machine Control*. Research Studies Press, Wiley, Taunton, England, 1989.

[61] D.A. Linkens and H. Wang. Qualitative bond graph reasoning in control engineering - part i: Qualitative bond graph modeling and controller design. In François E. Cellier and José J. Granada, editors, *1995 International Conference on Bond Graph Modeling and Simulation (ICBGM '95)*, number 1 in Simulation, pages 183–188, Las Vegas, January 1995. Society for Computer Simulation, Simulation Councils, Inc. Volume 27.

[62] D.A. Linkens and H. Wang. Qualitative bond graph reasoning in control engineering - part ii: Fault diagnosis. In François E. Cellier and José J. Granada, editors, *1995 International Conference on Bond Graph Modeling and Simula tion (ICBGM '95)*, number 1 in Simulation, pages 189–194, Las Vegas, January 1995. Society for Computer Simulation, Simulation Councils, Inc. Volume 27.

[63] F. Lorenz and H. Haffaf. Combinations of discontinuities in bond graphs. In François E. Cellier and José J. Granada, editors, *1995 International Conference on Bond Graph Modeling and Simulation (ICBGM '95)*, number 1 in Simulation, pages 56–64, Las Vegas, January 1995. Society for Computer Simulation, Simulation Councils, Inc. Volume 27.

[64] P. Lötstedt. Coulomb friction in two-dimensional rigid body systems. *Z. angew. Math. u. Mech.*, 61:605–615, 1981.

[65] David G. Luenberger. *Introduction to Dynamic Systems: Theory, Models, & Applications*. John Wiley & Sons, New York, 1979. ISBN 0-471-02594-1.

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

[67] Stefanos Manganaris. Introducing bond-graphs in qualitative reasoning about physical systems. MSc thesis, Vanderbilt University, Computer Science Department, June 1991.

[68] Jakob Mauss. Automatic transmission of a mercedes-benz. Personal communication, November 1995. Daimler-Benz.

[69] Jakob Mauss and Bernd Neumann. Diagnosis by algebraic modeling and fault-tree induction. In Wolfgang Nejdl, editor, *Sixth International Workshop on Principles of Diagnosis*, Goslar, Germany, October 1995.

[70] A. Misra, J. Sztipanovits, and R. Carnes. Robust diagnostic system: Structual redundancy approach. In *Proceedings of the SPIE's International Symposium on Knowledge-Based Artificial Intelligence Systems in Aerospace and Industry*, pages 249–260, Orlando, FL, April 1994.

[71] A. Misra, J. Sztipanovits, A. Underbrink, R. Carnes, and B. Purves. Diagnosability of dynamical systems. In *Proceedings of the Third International Workshop on Principles of Diagnosis*, pages 239–244, Rosario, WA, October 1992.

[72] Pieter J. Mosterman. Implicit modeling of nonlinear systems with neural networks. Technical Report #89R203, Control Laboratory, Electrical Engineering Department, University of Twente, 1989.

[73] Pieter J. Mosterman. Structured specifications for control system implementation. MSc thesis, University of Twente, Control Laboratory, 1991. Report Number: 91R076.

[74] Pieter J. Mosterman. Hybrid modeling of dynamic physical systems: An application in diagnosis. Technical Report #ECE-96-03, Center for Intelligent Systems, Electrical and Computer Engineering Department, Vanderbilt University, 1996. area paper.

[75] Pieter J. Mosterman and Gautam Biswas. Diagnosis of continuous valued systems in transient operating regions. *IEEE Transactions on Systems, Man, and Cybernetics*. in review.

[76] Pieter J. Mosterman and Gautam Biswas. Behavior generation using model switching a hybrid bond graph modeling technique. In François E. Cellier and José J. Granada, editors, *1995 International Conference on Bond Graph Modeling and Simulation (ICBGM '95)*, number 1 in Simulation, pages 177–182, Las Vegas, January 1995. Society for Computer Simulation, Simulation Councils, Inc. Volume 27.

[77] Pieter J. Mosterman and Gautam Biswas. Modeling discontinuous behavior with hybrid bond graphs. In *1995 International Conference on Qualitative Reasoning*, pages 139–147, Amsterdam, May 1995. University of Amsterdam.

[78] Pieter J. Mosterman and Gautam Biswas. Analyzing discontinuities in physical system models. In Yumi Iwasaki and Adam Farquhar, editors, *Qualitative Reasoning: The Tenth International Workshop*, pages 164–173, Stanford Sierra Camp, Fallen Leaf Lake, California, May 1996. AAAI Press, 445 Burgess Drive, Menlo Park, CA 94025. AAAI Technical Report WS-96-01, ISBN 0-57735-001-4.

[79] Pieter J. Mosterman and Gautam Biswas. A formal hybrid modeling scheme for handling discontinuities in physical system models. In *AAAI-96*, pages 985–990, Portland, Oregon, August 1996. AAAI Press, 445 Burgess Drive, Menlo Park, CA, 94025.

[80] Pieter J. Mosterman and Gautam Biswas. An integrated architecture for model-based diagnosis. In *Seventh International Conference on Principles of Diagnosis*, pages 167–174, Val Morin, Canada, October 1996.

[81] Pieter J. Mosterman and Gautam Biswas. Prediction, monitoring, and diagnosis in complex dynamic systems. Technical Report #ECE-96-04, Center for Intelligent Systems, Electrical and Computer Engineering Department, Vanderbilt University, 1996.

[82] Pieter J. Mosterman and Gautam Biswas. Verification of dynamic physical system models. In *ASME-96*, pages 707–714, Atlanta, GA, November 1996.

[83] Pieter J. Mosterman and Gautam Biswas. Formal specifications for hybrid dynamical systems. In *IJCAI-97*, August 1997.

[84] Pieter J. Mosterman and Gautam Biswas. Formal specifications from hybrid bond graph models. In *Qualitative Reasoning Workshop*, Italy, June 1997.

[85] Pieter J. Mosterman and Gautam Biswas. Hybrid modeling specifications for dynamic physical systems. In *1997 International Conference on Bond Graph Modeling and Simulation (ICBGM '97)*, pages 162–167, Phoenix, AZ, January 1997. Society for Computer Simulation, Simulation Councils, Inc.

[86] Pieter J. Mosterman and Gautam Biswas. Monitoring, prediction, and fault isolation in dynamic physical systems. In *AAAI-97*, Rhode Island, August 1997. AAAI Press, 445 Burgess Drive, Menlo Park, CA, 94025.

[87] Pieter J. Mosterman and Gautam Biswas. A theory of discontinuities in dynamic physical systems. *Journal of the Franklin Institute*, 334B(6), 1997.

[88] Pieter J. Mosterman, Gautam Biswas, and Janos Sztipanovits. Hybrid modeling and verification of embedded control systems. In *IFAC97*, Gent, Belgium, April 1997.

[89] Pieter J. Mosterman, John R. Bourne, Arthur J. Brodersen, and J. Olin Campbell. Electronics laboratory simulator instructor manual, laboratory manual, and reference manual. Technical report, Falcon Software, Wentworth, NH, 1995.

[90] Pieter J. Mosterman, J. Olin Campbell, Arthur J. Brodersen, and John R. Bourne. Design and implementation of an electronics laboratory simulator. *IEEE Transactions on Education*, 39(3):309–313, August 1996.

[91] Pieter J. Mosterman, Marcel A.M. Dorlandt, J. Olin Campbell, Craig Burow, René Bouw, Arthur J. Brodersen, and John R. Bourne. Virtual engineering laboratories: Design and experiments. *Journal of Engineering Education*, pages 279–285, July 1994.

[92] Pieter J. Mosterman, Ravi Kapadia, and Gautam Biswas. Using bond graphs for diagnosis of dynamic physical systems. In *Sixth International Conference on Principles of Diagnosis*, pages 81–85, Goslar, Germany, October 1995.

[93] Pieter J. Mosterman, Narasimhan Sriram, Gautam Biswas, Takashi Washio, and Shinji Yoshikawa. Process diagnosis in transient operating regions: Fault detection and isolation in a liquid sodium cooling system. In *IJCAI-97 QR Applications in Engineering*, August 1997. in review.

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

[95] P. Pandurang Nayak. Causal approximations. In *Proceedings AAAI-91*, pages 703–709, Menlo Park, CA, 1991. AAAI/MIT Press.

[96] P. Pandurang Nayak, Leo Joscowicz, and Sanjaya Addanki. Automated model selection using context-dependent behaviors. In *Proceedings AAAI-91*, pages 710–716, Menlo Park, CA, 1991. AAAI/MIT Press.

[97] Xavier Nicollin, Joseph Sifakis, and Sergio Yovine. From atp to timed graphs and hybrid systems. In J.W. Bakkers, C. Huizing, W.P. de Roeres, and G. Rozenberg (eds.), editors, *Lecture Notes in Computer Science*, volume 600, pages 549–571, Mook, The Netherlands, June 1991. Real Time: Theory and Practice.

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

[99] Guy Norris. Boeing's seventh wonder. *IEEE Spectrum*, pages 20–23, October 1995.

[100] Edward F. Obert. *Internal Combustion Engines: Analysis and Practice*. International Textbook Company, Scranton, Pennsylvania, 2 edition, 1950.

[101] O.O. Oyeleye, F.E. Finch, and M.A. Kramer. Qualitative modeling and fault diagnosis of dynamic process by MIDAS. In Walter Hamscher, Luca Console, and Johan de Kleer (eds.), editors, *Readings in Model-Based Diagnosis*, pages 249–254. Morgan Kaufmann Publishers, Inc., San Mateo, CA, 1992. ISBN 1-55860-249-6.

[102] B.L. Palowitch. *Fault Diagnosis of Process Plants using Causal Models*. PhD dissertation, Massachusetts Institute of Technology, August 1987.

[103] Kevin M. Passino and Ümit Özgüner. Modeling and analysis of hybrid systems: Examples. In *Proceedings of the 1991 IEEE International Symposium on Intelligent Control*, pages 251–256, Arlington, Virginia, USA, August 1991.

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

[105] Roger Penrose. *The emperor's new mind: concerning computers, minds, and the laws of physics*. Oxford University Press, Oxford, 1989.

[106] J.A.M. Remmerswaal and H.B. Pacejka. A bond graph computer model to simulate vacuum cleaner dynamics for design purposes. *Journal of the Franklin Institute*, 319(1/2):83–92, January/February 1985.

[107] Ronald C. Rosenberg and Dean Karnopp. *Introduction to Physical System Dynamics*. McGraw-Hill Publishing Company, New York, New York, 1983.

[108] W. Rudin. *Principles of Mathematical Analysis*. McGraw-Hill, New York, 3 edition, 1976.

[109] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D.C. Teneketzis. Failure diagnosis using discrete-event models. *IEEE Transactions on Automatic Control*, 40(9):1555–1575, September 1995.

[110] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D.C. Teneketzis. Failure diagnosis using discrete-event models. *IEEE Transactions on Control Systems Technology*, 4(2):105–124, March 1996.

[111] H. Schneider and P.M. Frank. Observer-based supervision and fault detection in robots using nonlinear and fuzzy logic residual evaluation. *IEEE Transactions on Control Systems Technology*, 4(3):274–282, May 1996.

[112] J.E.E. Sharpe and R.H. Bracewell. The use of bond graph reasoning for the design of interdisciplinary schemes. In François E. Cellier and José J. Granada, editors, *1995 International Conference on Bond Graph Modeling and Simulation (ICBGM '95)*, number 1 in Simulation, pages 116–121, Las Vegas, January 1995. Society for Computer Simulation, Simulation Councils, Inc. Volume 27.

[113] Jeffrey L. Stein. Modeling and state estimator design issues for model-based monitoring systems. *Transactions of the ASME*, 115:318–327, June 1993.

[114] Jeffrey L. Stein and Loucas S. Louca. A component-based modeling approach for system design: Theory and implementation. In François E. Cellier and José J. Granada, editors, *1995 International Conference on Bond Graph Modeling and Simula tion (ICBGM '95)*, number 1 in Simulation, pages 109–115, Las Vegas, January 1995. Society for Computer Simulation, Simulation Councils, Inc. Volume 27.

[115] Jeffrey L. Stein and Youngjin Park. Modeling and sensing issues for machine diagnosis. In *Proceedings of the 1988 American Control Conference*, Atlanta, Georgia, June 1988.

[116] Jan-Erik Strömberg, Ulf Söderman, and Jan L. Top. Bond graph supported estimation of discretely changing parameters. In *Proceedings of the International Conference on Bond Graph Modeling*, pages 47–52, San Diego, California, 1993.

[117] Jan-Erik Strömberg, Jan Top, and Ulf Söderman. Variable causality in bond graphs caused by discrete effects. In *Proceedings of the International Conference on Bond Graph Modeling*, pages 115–119, San Diego, California, 1993.

[118] C. Sueur and G. Dauphin-Tanguy. Bond graph approach for structural analysis of mimo linear systems. *Journal of the Franklin Institute*, 328(1):55–70, 1991.

[119] William Sweet. The glass cockpit. *IEEE Spectrum*, pages 30–38, September 1995.

[120] J. Sztipanovits, G. Karsai, C. Biegl, T. Bapty, A. Ledeczi, and A. Misra. Multigraph: An architecture for model-integrated computing. In *ICECCS'95*, Ft. Lauderdale, Fla, November 1995.

[121] Jean Thoma. Thermofluid systems by multi-bondgraphs. *Journal of the Franklin Institute*, 329(6):999–1009, 1992.

[122] Jean U. Thoma. Bond graph for thermal energy transport and entropy flow. *Journal of the Franklin Institute*, 292(2):109–120, 1971.

[123] Jean U. Thoma. Entropy and mass flow for energy conversion. *Journal of the Franklin Institute*, 299(2):89–96, 1975.

[124] Jean U. Thoma and Henri Atlan. Network thermodynamics with entropy stripping. *Journal of the Franklin Institute*, 303(4):319–328, April 1977.

[125] J.U. Thoma. Models, bond graphs and entropy. In J.J. van Dixhoorn and F.J. Evans (eds.), editors, *Physical Structure in Systems Theory: Network Approaches to Engineering and Economics*, pages 17–33. Academic Press, London, 1974.

[126] Jan Top. *Conceptual Modeling of Physical Systems.* PhD dissertation, University of Twente, CIP-DATA Koninklijke Bibliotheek, Den Haag, The Netherlands, 1993.

[127] J.L. Top, J.M. Akkermans, and P.C. Breedveld. Qualitative reasoning about physical systems. *Journal of the Franklin Institute*, 328(5/6):1047–1065, 1991.

[128] F.W. Umbach. *Organisatie van Instrumentatie.* H.E. Stenfert Kroese B.V., Leiden/Antwerpen, 1981.

[129] Johannes van Dijk. *On the role of bond graph causality in modelling mechatronic systems.* PhD dissertation, University of Twente, CIP-Gegevens Koninklijke Bibliotheek, Den Haag, The Netherlands, 1994. ISBN 90-9006903-8.

[130] J.J. van Dixhoorn and P.C. Breedveld. *Technische Systeemleer.* University of Twente, Twente, Netherlands, fourth edition, February 1985. class pack.

[131] Paul T. Ward. *Systems Development without Pain.* YOURDON, New York, N.Y., 1984.

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

[133] K.C.J. Wijbrans. *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, 1993. ISBN 90-9005933-4.

[134] Xudong Yu. *Multi-Level Reasoning and Diagnosis for Complex Continuous-Valued Systems.* PhD dissertation, Vanderbilt University, Computer Science Department, December 1992.