

**MODELING THE PHYSICAL
AND LOGICAL PROPERTIES OF
MECHATRONICS SYSTEMS**
-A Manufacturing Systems Theory Approach-

Bassam A. Hussein

**Submitted to the Norwegian University of Science and Technology in partial
fulfillment of the requirements for a Doctor of Engineering degree.**

Trondheim 1999

© Bassam A. Hussein
1901.99

Preface

Work on this thesis started in September 1995 as a part of a joint program between the Norwegian University of Science and Technology - NTNU, The Foundation for Scientific and Industrial Research at the Norwegian Institute of Technology - SINTEF, and Norwegian industry. The overall objective of the program is to identify, to test and to improve methodology, methods and tools for design of mechatronics products, suitable for application in industry.

I would like to take this opportunity to express sincere gratitude to Professor Øyvind Bjørke for the trust and support he has given me and for the stimulating discussions during the last four years. I would also like to extend my special thanks to Professor Peter Falster at the Technical University of Denmark for his efforts to make my stay in Denmark possible in the spring of 1998. My thanks go to my family, friends and colleagues for their support, encouragement and help.

Abstract

Mechatronics systems are hybrid systems with interacting continuous and discrete components. The behavior of mechatronics systems is described by equations of motions that generally depend on both. In turn, these equations contain a mixture of dynamic equations and logic. The continuous structure represents *the physical system* which includes the actuators, the target system and the sensing system. The discrete structure represents *the logical system* which could be seen as a mode changer, a supervisor, a state machine or a tuner depending on the application. The logical system senses and tries to control the physical system by issuing discrete logical directives at event times that will bring the physical system to a desired mode or state.

The presence of such discrete events could cause the physical system to switch between different mode models. The overall system then evolve in a piecewise continuous manner, where governing equation changes at event time possibly accompanied by jumps in state variables. Mathematically, this causes physical state variables to switch or jump between different modes of operations. Our primary goal in this dissertation is to develop a consistent modeling framework that is capable of handling this mode-switching phenomenon. The formalism, which we adopt here, is based on utilizing multidimensional arrays as a modeling tool. The general features of this framework can be summarized as follows:

1. The logical system is modeled as an array-based inference engine that handles all deductive reasoning activities. The logical system is seen here as a higher level abstraction of the physical system whose logical variables have meanings grounded in the physical system and its environment.
2. The physical system is modeled as a set of hybrid differential equations that explicitly contain all modes of operations the physical system undergoes. In this formulation, the physical system model contains real as well as logic variables from the binary domain. These logic variables represent the directives issued by the logical system to bring the physical system to the desired mode.

The most remarkable advantages, in our opinion, of the presented modeling framework are:

1. Small set of operations will apply in general for model formulation.
2. Model generation can be fully automated.
3. Mixed discrete and continuous simulation can be performed in any array-oriented environment.

In the course of working with the subject of system modeling, we have examined the possibility of using array-based logic for feedback control. The intention was to extend the role of logical system from mainly a switching logic controller to include feedback controller. The results of this work are published in [34,35]. We have chosen not to include this part of the research in the dissertation in order to keep the focus on the continuous system modeling.

CONTENTS

Preface -----	3
Abstract -----	5
1 INTRODUCTION-----	9
1.1. Mechatronics-----	10
1.1.1. Some history -----	12
1.1.2. Multidisciplinary nature of mechatronics -----	14
1.2. Mathematical Actors-----	16
1.2.1. Mode switching phenomenon -----	17
1.2.2. Computational problems associated with the multiplicity of models	19
1.2.3. Switching elements -----	20
1.3. Thesis Outline -----	21
2 PRINCIPLES OF SYSTEM MODELING -----	23
2.1. Elements of Modeling -----	23
2.2. Physical System Modeling-----	25
2.2.1. Physical primitive elements-----	25
2.2.2. Connections-----	28
2.3. Logical System Modeling - Array Based Logic-----	31
2.3.1. Logical elements -----	32
2.3.2. Connections-----	34
2.3.3. Sources -----	35
2.4. Summary -----	40
3 INTERFACE -----	41
3.1. Interface Through Sources -----	42
3.2. Application Example: A Manufacturing System-----	43
3.2.1. System modeling -----	44
3.2.2. Simulation-----	48
3.3. Summary -----	52
4 SYSTEM INTERFACE -----	53
4.1. Initial Value Problem -----	54
4.2. Variable State Space Dimension -----	55
4.3. Switching Elements -----	59
4.4. Model Formulation-----	59
5 NON-IDEAL SWITCHING ELEMENTS-----	61
5.1. Application Example: Gearbox-----	65
5.1.1. Logical conditions for switching -----	67
5.1.2. System modeling -----	69
5.1.3. Simulation-----	70
5.2. Summary -----	72
6 IDEAL SWITCHING ELEMENTS -----	73
6.1. The Switch As a Connection Element-----	75
6.2. Types of Switch Connection-----	76

6.2.1.	Switch is connected in series with an element-----	77
6.2.2.	Switch is connected between two free nodes -----	82
6.3.	Application Example: Gearbox – Ideal Case -----	91
6.3.1.	System modeling -----	92
6.3.2.	Simulation-----	96
6.4.	Summary -----	98
7	CONCLUSIONS-----	99
	REFERENCES -----	101
	APPENDIX A: MANUFACTURING SYSTEM-----	105
A.1.	Physical System Model-----	105
A.1.1.	Modeling boring spindle subsystem-----	108
A.1.2.	Modeling hydraulic subsystem-----	110
A.1.3.	Total physical system model -----	114
A.2.	Logical System Modeling-----	114
	LIST OF FIGURES -----	117
	LIST OF TABLES -----	120
	LIST OF EXAMPLES -----	120

1 INTRODUCTION

A central idea involved in the study of any physical, sociological or economical system is the idea of *modeling* and *simulation* [50]. These two concepts have become central to all discipline of engineering and science. Modeling and simulation are used in the analysis of real systems where they help us to gain understanding of the functioning of our real world. They are also important to the design of new systems where they enable us to predict their behavior before they are built.

Modeling can be described as *the process of organizing knowledge about a given system* [18]. Models of systems are simplified, abstracted constructs used to represent some aspects of the real system. Scaled physical models are well known in engineering, in this category falls the wind tunnel models, oil platform models and plastic models of metal parts used in photo-elastic stress analysis. These physical models are mini-versions of the real system and thus contain only those aspects of the real system that are supposed to be important to the characteristics under study.

In this dissertation, another type of models often regarded as *hybrid models* is considered. Hybrid models are mergers of dynamic equations and logic [2,16]. This type of models is more abstract than the scaled physical models but is well suited to study and analyze hybrid systems.

In our context, a hybrid system is a mechatronics system. Mechatronics is a term used to denote large number of systems and processes which contain elements from multiple domains (mechanics, electronics and software) all brought together to form a system in its own right [43]. Examples of mechatronics systems include, disk drives, flexible manufacturing systems, and constrained robots. The reader may imagine more or consult the literature [30].

Colloquially, hybrid systems are those in which a melding of two worlds, the digital and analog world exists and they are intertwined to the extent that “one-world” description is not possible or not traceable. Such systems seem to arise in variety of applications, due to autonomous or controlled phenomena [16,17]. In the autonomous case, the system evolution itself may fall naturally into a finite number of different phases or modes due to natural phenomena; examples of the later are systems involving collision as in the famous bouncing ball problem.

The controlled hybrid system arises as a mean of dealing with the complexity of modern engineering systems through for example, using logic to switch between various controllers each with predictable behavior [46]. The simplest example of a controlled hybrid system is the thermostat [5,6].

By performing experiments on these models, we can gain understanding of how the real system will behave if some predefined conditions are met. The process of experimentation with models is known as *simulation*. Except by experimentation with real systems, simulation is the only technique available for the analysis of system behavior.

In the following sections, we shall begin by elaborating further on some important keywords. We shall start by reviewing the hybrid nature of mechatronics as well as modeling and simulation problems associated with this hybrid nature.

1.1. Mechatronics

It is generally accepted that the operation of any target system or process requires three fundamental activities [8]:

- Decision-making
- Power Delivery
- Measurements

A knowledge base, which contains qualitative and quantitative models of the process under control, is used to support these activities. A simplified system structure for these activities is shown in Figure 1.1.

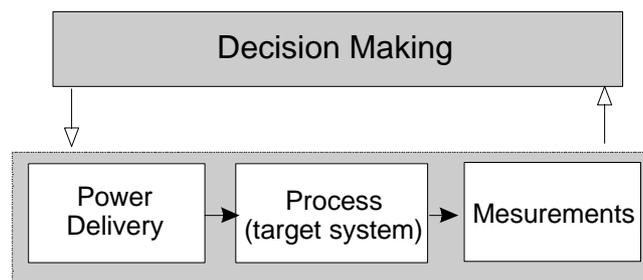


Figure 1.1. Fundamental activities in a controlled process

Decision-making: This is a complex activity that utilizes both quantitative and qualitative models and data from all scales of measurements in order to infer *what should be done*. Decision-making capabilities are the indicator that differentiates modern engineering systems from one to another. This activity is what effectiveness, functionality, intelligence, creativity and control by all its forms is all about, and it lies at the center of any controlled physical, economical, sociological or biological system.

In an automatic vending machine, decision-making implies that the system should determine what should be the next operation when a customer inserts a coin into the machine and punches a button to receive an item.

Another example, in an adaptive system, decision-making implies the selection of the next control law to be used in order to adapt to process variations so that the system will be brought into a desired state. Adaptive-control implements methods such as; gain scheduling, model reference adaptive control, and self-tuning control [57]. With gain scheduling, for instance, look-up table containing an arrangement of controller gains is employed. The full range of available gain settings cover every circumstances the plant undergo, see Figure 1.2.

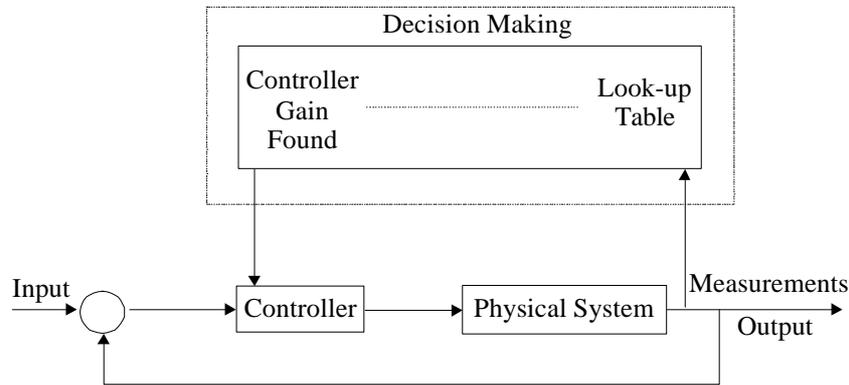


Figure 1.2. Gain scheduler in an adaptive system

This type of decision-making signifies the so-called *logical decision-making*. Here the system infer what should be done next based on a logical model that describe purely logical relation between the input, output and internal state of the process.

Decision-making could also imply the determination of the amount of power required for performing an operation, or even the duration of an operation. Here decision-making utilizes quantitative process models which describe quantitative relations between process variables.

Power delivery: Power delivery or actuation is required in order to execute the decision by means of some power source such as electrical motors, hydraulic pumps and so on.

Measurements: involving collection of both quantitative and qualitative *data* from the process and its environment. These measurements give a picture of *what is happening* inside and outside the target system. Collecting these measurements requires instrumentation in order to collect and transform these real world data from an energy medium to another suitable for use by other process.

The history of machines shows that the *means* of performing these activities have evolved from *all mechanical*, with or without human intervention to a fusion of mechanical, electronic, and software components in mechatronics

systems forming a hybrid system. The following section is a brief review of this evolution.

1.1.1. Some history

In the beginning, when Man acquired his first primitive tool made of rocks, Man utilized the available resources in his primitive world such as his strength (power delivery) in order to move, halt, and to accelerate the tool. All aspects related to controlling, coordinating and managing the tool's motion and himself (decision-making) were then carried out in human's brain. This tool represented prehistoric Man's attempt to direct his own physical strength under the supervision of human intelligence. Thousands of years later, Man developed simple mechanical devices and machines by which he could magnify the power of his own strength. These devices, such as levers and pulleys shown in Figure 1.3, were still directed and coordinated by a complex decision-making process executed in the human brain.

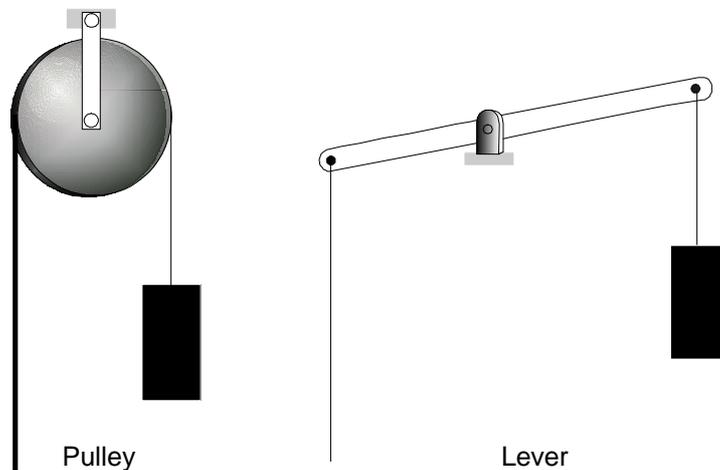


Figure 1.3. Examples of early machines

Next, came the development of powered machines that did not require human strength in order to operate, such as waterwheels and windmills, steam engines, electrical motors and others.

Aeolipile was the first steam turbine invented in the first century AD by *Heron* of Alexandria and described in his *Pneumatica* [20]. The *aeolipile* is the first known device to transform steam into rotary motion. *Heron* used this device to power an all-mechanical mechanism of pulleys, screws and wedges assembled to open and close the first *automatic door* in the history of invention [19].

This mechanical structure was an early attempt to utilize mechanical components for implementing logic control. For example, a lever element

was used to realize a NOT gate, a lever and a spring was used to realize an AND gate. In fact, many of these elements are still being used in many automated systems [25].

Although, *Heron's* main objective by his invention was only to impress the monks in the Temple of Alexandria, his invention remains an elegant *all mechanical* device for logic control. The all-mechanical device was an early example of how the decision-making is done in the same medium as the major power flows in the system.

Each development in the history of powered machines has brought with it an increased requirement for utilizing new means in order to harness the power of the machine and to replace the need for humans. The earliest steam engines required a person to open and close the valves, first to admit steam into the piston chamber and then to exhaust it. Later a slide valve mechanism was devised to automatically accomplish these functions. The only need of the human operator was then to regulate the amount of steam that controlled the engine's speed and power. This requirement for human attention in the operation of the steam engine was eliminated by the flying-ball governor, invented by James Watt in England in 1796 [27].

The flying-ball governor shown in Figure 1.4, remains a classical example of a negative feedback control system, in which the increasing output of the system is used to decrease the activity of the system. Flying-ball governor represented a control device with built in decision-making for controlling the engine's speed.

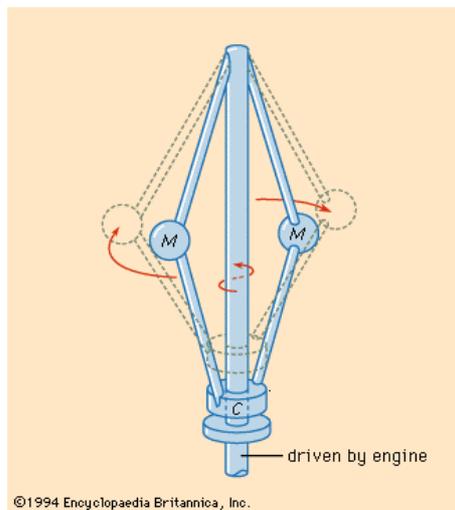


Figure 1.4. Flying-ball governor

The revolving masses are balls attached to a vertical spindle by link arms, and the controlling force consists of the weight of the balls (control law). If the load on the engine decreases, the speed will increase, the balls *M* will

move out (measurements), and the member C will slide up the vertical spindle and reduce the steam admitted to the engine (actuation). Thus, reducing the speed, an increase in the load will have the opposite effect.

The flying-ball governor was not the first regulator in the history of machine control. However, it draws its significance as it was used by J.C. Maxwell to demonstrate the importance and usefulness of quantitative mathematical models and methods in understanding complex phenomena and signaled the birth of mathematical systems and conventional control theory [42]. It was not until the invention of operational amplifier for electronic feedback by Bode, Nyquist and Black at Bell Laboratories, the computations was carried out electronically [20]. The development of the operational amplifications techniques opened the door for mechatronics. They provided a variety of linear, non-linear, static and dynamic computing actions [8].

1.1.2. Multidisciplinary nature of mechatronics

A number of significant developments in various fields have evolved during the 20th century: the digital computer, improvements in data-storage and software technology, advances in sensor technology, integrated circuits, microprocessors and communication technology. All these developments have provided mankind with a rich selection of new techniques and tools to enhance the three activities involved in the process; power delivery, measurements as well as implementing higher level decision-making as shown in Figure 1.5.

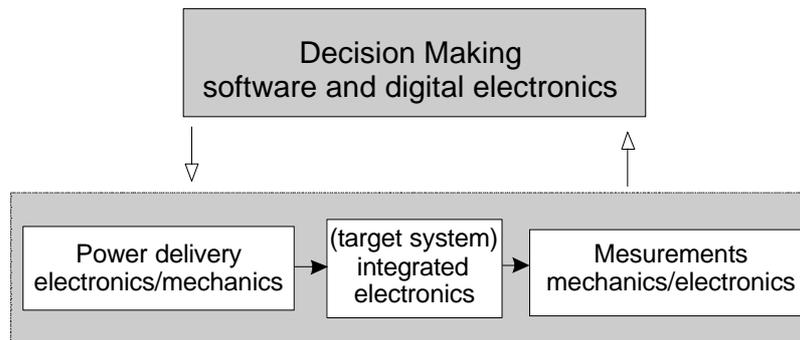


Figure 1.5. Multidisciplinary nature of mechatronics

Mechatronics recognizes that the integration of these techniques, and the resulting transfer of functionality from mechanical domain to software and digital electronics domain leads to the appearance of a new range of products with extra qualities in terms of appearance, size, flexibility, reliability, autonomy and so on [15,31,32,36,54]. With mechatronics multiple technologies are brought together to form an integrated system.

In the early days of the 70's mechatronics was viewed as a combination of mechanics and electronics and was mainly concerned with servo technology for higher performance [30]. For example, the mechanical commutation in direct current servomotors was duplicated by an electronic commutation using solid state switching elements. In swing machines, the gear mechanism was replaced by stepping motors with electronic control. Centralized drives with chains, belts and so on are replaced by decentralized drives under computer control, leading to simpler kinematics structure. All these innovations brought with it improvements in performance, reliability and productivity. Typical mechatronics product at that time included automatic doors, automatic vending machines, auto-focusing cameras, and so on.

The introduction of microprocessors technology in the 80's and advances in communication technologies considerably enhanced the computational capabilities of machines in terms of speed and amount of information to be handled in real time. These innovations opened the door for the introduction of new (*intelligent*) functions that was previously done by humans. These functions were added to the process to increase the speed of response to failure, to relieve the operator from mundane tasks, to protect them from hazards. The improved and new functions as a result of the integration between mechanics, electronics and software in mechatronics systems could be summarized as follows [36]:

- Precise speed control for all operating conditions
- Simpler kinematics by decentralized drives
- Control of non-measurable but reconstructed or estimated variables
- Operation in unstable or dangerous regions
- Adaptive damping of oscillations or unbalance
- Optimization of efficiency or pollution
- Supervision with fault diagnosis
- Anti blockage or slip control
- Overall process management

Indeed, the allocation of technologies and the fusion of the different domains into an overall working system require a new design philosophy. In this design philosophy, system thinking, creativity and cost effectiveness are central elements [15,32].

In mechatronics, digital control makes it relatively easy to build devices in which the applied control is any computable function of system measurements. This widens the scope for the system designs that can be implemented, but the additional design freedom brings with it many more design parameters. Consequently, we seek tools for modeling and analyzing the overall system as an integrated unit in order to cope with the newfound freedom. Modeling tools that take into account restrictions caused by the integration of both continuous and discrete components for achieving

optimum system behavior. A task we believe defies any fragmented approach of modeling. If the concurrent design approach is an important element for high-quality mechatronics products, then a unified approach of modeling and simulation is an important element to support such successful concurrent design. We begin here to develop a unified modeling framework that can help us to cope with the newfound freedom.

1.2. Mathematical Actors

In the light of the above, a fundamental characteristic of mechatronics systems is that they contain analog components which make up the *physical system*. This system contains the actuators required for power delivery, the mechanical process, and the instrumentation required for measurements and signal processing and power modulation. All aspects related to controlling, coordinating and managing this physical system as well as communications with the user and other systems are performed by a multilevel decision-making structure implemented by a digital logic device. The logic device could be a mode changer, a tuner, or a supervisor depending on the application. We shall refer to this digital logic device as the *logical system*.

The physical system lives in a continuous or piecewise continuous time world. Mathematically seen, its input, output and internal states are all points in a *continuous-time metric space*. In contrast, the input, output, and internal states of the *logical system* are all points in a *discrete metric space* [48]. Hence, a mechatronics system could then be seen as a hybrid system that mixes together two spaces; a higher level logical system and lower level physical system as shown in Figure 1.6.

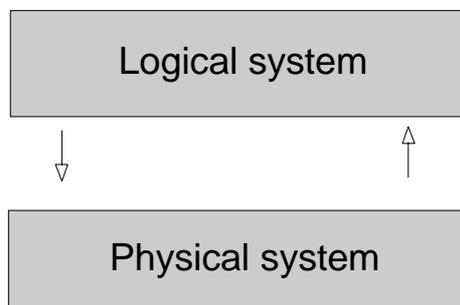


Figure 1.6. Hybrid nature of mechatronics systems

The theory of hybrid systems have recently evolved in order to encounter the many theoretical and technological problems associated with the mixture between the continuous world and the discrete world [1,2,3,4,5,28,29,46].

1.2.1. Mode switching phenomenon

Typically, the logical system is used to sense and control the physical system by issuing discreet directives that will bring the physical system to a desired mode or state.

Directives issued by the logical system may involve, see Figure 1.7:

1. Switching various continuous controller in and out of use such as adaptive control
2. Connecting or disconnecting faulty equipment such as fault detection and recovery.
3. Start or stop an operation such as sequence or logic control.

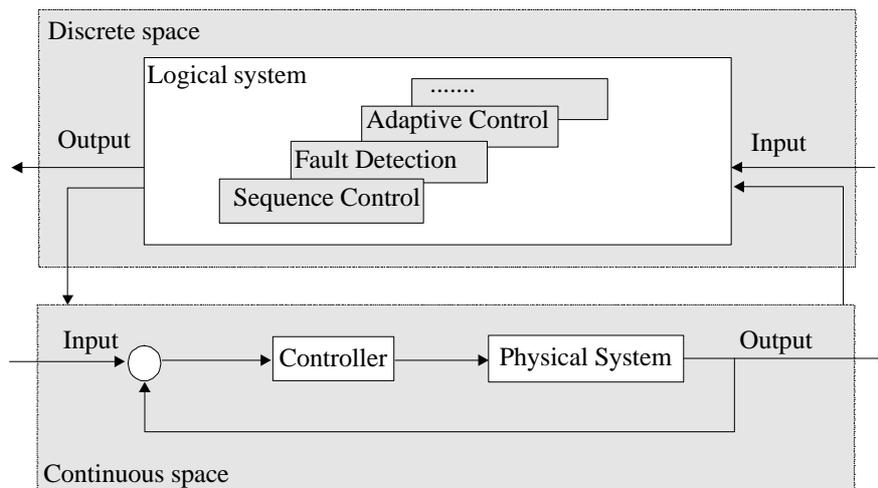


Figure 1.7. Directives from the logical system to the physical system

When to issue these directives is up to the logical system. In turn, the logical system utilizes continuous as well as discrete measurements from the physical system and its environment.

As far as modeling the logical system is concerned, our aim is restricted to formalize the logical relations between input, output and internal states and express this formalization in array terms. The objectives of such model are to verify the logical requirements and to guarantee that the logical system is consistent and complete.

We shall generally refer to the directives issued by the logical system to the physical system as *controlled discrete events*. These controlled events can be generally classified into two classes:

1. Control directives that are intended to alter only the input to the physical system as shown in Figure 1.8. This type of discrete events

will not contribute to altering the dynamical structure of the physical system.

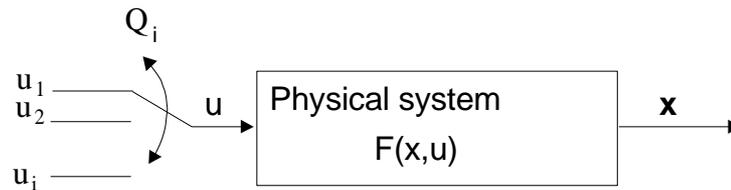


Figure 1.8. Control directives Q_i only alter the input u_i to the physical system

2. Control directives that will impose discontinuity on the behavior of the physical system. Here, the physical system will *switch* or *jump* between different continuous system models. Each one is called mode or a regime with distinct dynamic characteristics as shown in Figure 1.9.

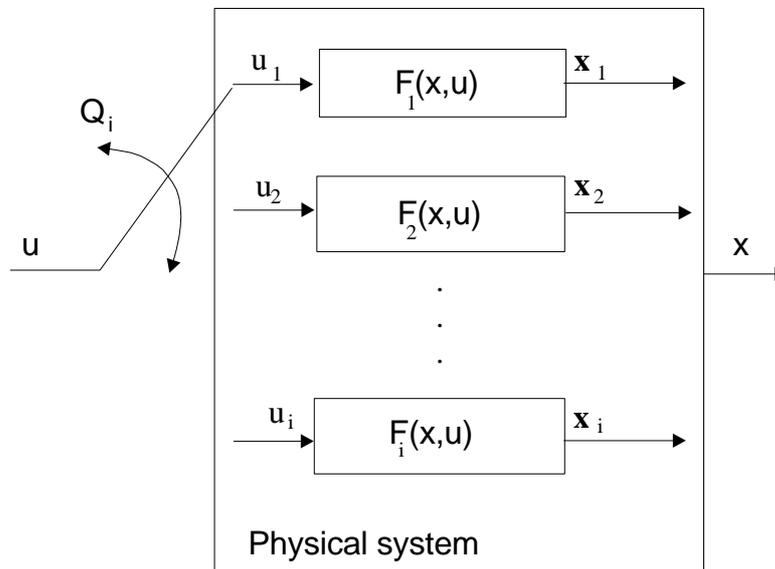


Figure 1.9. Directives issued by the logical system Q_i cause the physical system to switch between multiple modes

In this dissertation, we focus on the switching phenomenon that arises because of the second type of discrete events. Here, the output from the logical system Q_i causes the physical system to switch or jump between multiple modes.

A system that evolves between different modes of operations where each mode has predictable behavior is referred to as *mode switching system*.

Mathematically, mode-switching systems are described by multiple set of equations. Each particular mode is associated with one set of equations describing the continuous behavior at that particular mode. This multiplicity will evidently complicate both modeling and simulation, particularly if we consider the number of models required for describing all modes of operations.

1.2.2. Computational problems associated with the multiplicity of models

Seen from the physical system standpoint, one can single out two main computational problems associated with the multiplicity of models:

1. Initial value problem.
2. Variable state space dimension.

Initial value problem: Each mode model operates as a continuous system described by the usual differential or difference equations. When the logical system switches to the next mode, there is a jump discontinuity from the end of application of the old mode to the beginning of application of the next mode [48]. This will occur, for example when one use logic to switch between different continuous controllers, each with predictable behavior [46].

Variable state space dimension: This problem arises when the length of state vector in one mode has lower rank than the state vector of the previous mode. This will occur when modeling components failure or changes in dynamical description of the continuous system, for example, modeling an automatic gearbox. In this case, some state variables could no longer be observable or controllable from this particular mode. Variable state space dimension is a simulation problem since there is no any numerical integration algorithm that can handle variable state space dimension.

Thus, as far as modeling the physical system is concerned, our objective in this regard is to develop a modeling framework by which we can handle the above two problems and formulate a continuous system model that:

1. Explicitly contains all modes of operations the system undergoes.
2. Preserve state space dimension of the derived explicit model to remain invariant to switching.

In this formulation, the continuous system model contains real as well as logic variables from the binary domain. These logic variables represent the directives issued by the logical system to drive the physical to the desired mode. Formally, the continuous system model will be given by the hybrid equations:

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

Where (x) is continuous state variables, (u) is the input vector to the continuous system, (t) is time and (Q) is the set of discrete logic variables involved in mode switching and generated by the logical system.

By such hybrid formulation of the continuous system model we have guaranteed:

1. Smooth transition (bump-less switching) from one mode to the next mode.
2. State space dimension of the continuous system model will remain invariant to switching and hence persevering aspects like observability and controllability.

1.2.3. Switching elements

Directives issued by the higher level logical system are often executed using explicit switching elements that are intentionally designed to perform switching directives. Examples of this category from physical domains are; electrical switch, mechanical clutch, and hydraulic valves as shown in Figure 1.10.

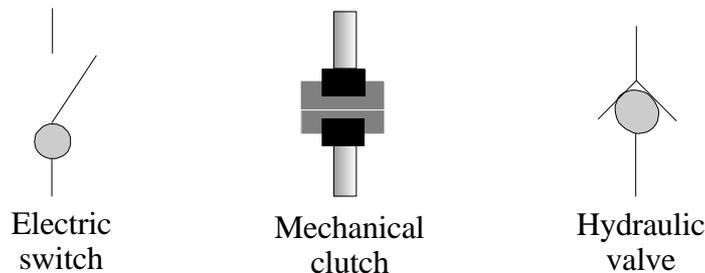


Figure 1.10. Switching elements from multiple physical domains

In other computer-based applications, directives are executed in a digital or electronic medium. In both applications, they all share the same discrete characteristics, they are either active or inactive. Therefore, in the topological structure of the physical system, all directives from logical system shall be represented by a switch element that can assume one of two states [On, Off]. The state of the switch element will be controlled by the higher level logical system. We shall come back to the characteristics of the switch element in chapter 5 and 6. Here we only assume that it is present and represents the interface between the logical and the physical system.

Mathematically, the state of the switch will be represented by a binary variable (S_w) to indicate that the switch is in [open] state.

The compliment of this binary variable will then be labeled (\tilde{S}_w) and will be used to indicate that the switch is in [closed] state.

$$\tilde{S}_w = \text{not}(S_w)$$

Table 1.1 shows a summary for the meaning of the binary variable S_w and its complement \tilde{S}_w in some physical domains.

Binary variable	Electrical	Mechanical	Hydraulic
$S_w = 1$	Switch is open	Clutch is disengaged	Valve is open
$\tilde{S}_w = 1$	Switch is closed	Clutch is engaged	Valve is blocking

Table 1.1. Meaning of the binary variable S_w and its complement \tilde{S}_w in some physical domains

1.3. Thesis Outline

The thesis contains seven chapters in addition to this introduction. In chapter two, we shall start by outlining the fundamental principles for physical and logical system modeling. We shall follow the mathematical foundations developed in the Scandinavian school of systems theory particularly those developed in Norway [13] and in Denmark [23,24,47].

In chapter three, we shall consider the first level of interface between the logical system and the physical system when the objective of the logical system is restricted to switch between different input vectors for the physical system to use. A manufacturing system is used as an application example.

In chapter four, we shall elaborate further on the computational problems associated with the multiplicity of models. That is, when directives from the logical system will switch elements and components into and out of use causing the physical system to switch between different modes or regimes. In chapter four, we shall also explain the role of switching elements in this hybrid environment.

In chapter five, we shall present the first approach to deal with mode switching phenomenon. This approach employs non-ideal switching elements in order to handle the problems arising from the multiplicity of models. An automatic gearbox is used as an application example.

In chapter six, we shall present the core contribution of this research and present a more general approach by utilizing ideal switching elements. The automatic gearbox has also been used as an application example. Finally, chapter seven presents the conclusions and discussions.

As we have mentioned in the abstract, during work on system modeling, we have examined the possibility of using array-based logic for feedback

control. The intention was to extend the role of logical system from mainly a switching controller to include feedback controller. The results of this work are published in [34,35]. We have chosen not to include this part of the research in the dissertation in order to keep the focus on continuous system modeling.

2 PRINCIPLES OF SYSTEM MODELING

The objective of this chapter is to present and outline the main concepts involved in the process of modeling for both the physical as well as the logical system.

2.1. Elements of Modeling

A system is defined as *a set of connected elements* that interact with its surroundings through *sources*. The **elements** carry all properties of the system, they are the basic primitive components of the system. **Connection** reflects how the elements influence each other and it represents the internal constraints between system elements. The **sources** reflect the external constraints between the system and its environment.

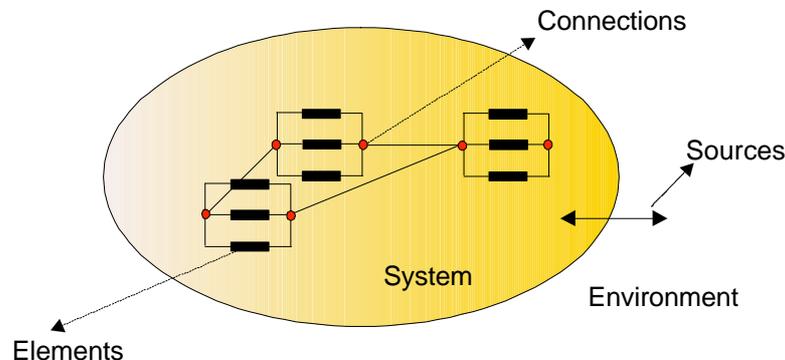


Figure 2.1. Elements, connections and sources in the system model

The above definition will be our reference in modeling the physical and the logical system. The definition is application-neutral and therefore well suited as a foundation for modeling a multi-disciplinary system. Based on our perception of a system and throughout the process of modeling, we shall distinguish between the following concepts, please see Figure 2.2.

Decomposition: The purpose of decomposition is to reduce the complexity of the total system by dividing the system into a hierarchy of subsystems, and primitive elements. The breaking up of the total system is carried out on several stages until the primitive elements that constitute the system are isolated and identified. The hierarchy of systems, subsystems and primitive elements is not absolute, since the most primitive part of a system could be modeled in such detail that it would be a complex subsystem. Along this hierarchy of systems, subsystems and primitive elements, a subsystem could be seen as the system part that can be considered as a system in itself,

with a set of connected elements. On the other hand, a primitive element is the only component that could be treated as a black box without any need to know what causes it to act as it does.

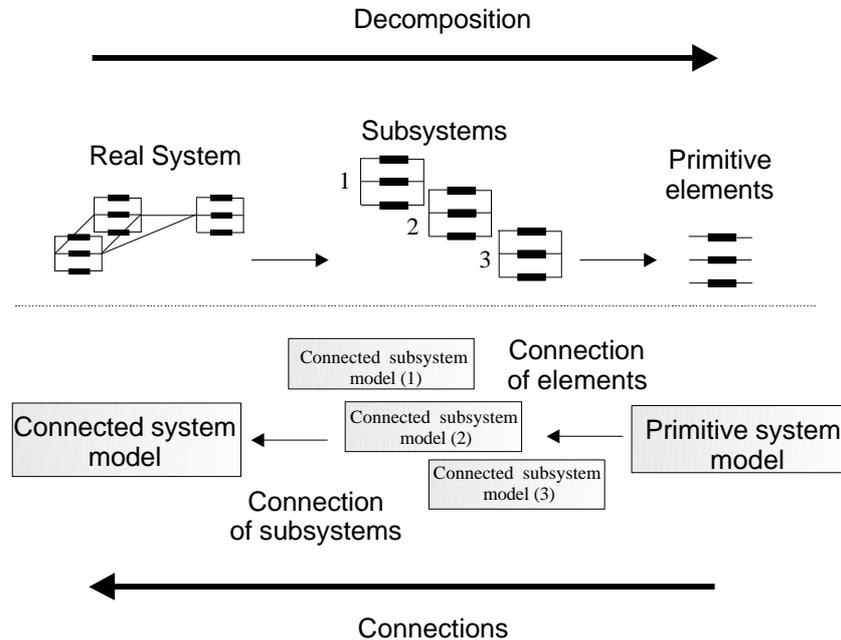


Figure 2.2. Elements of modeling

Primitive system model: This is a mathematical description of the system's elements in the disconnected state. It expresses the relation between the variables in the elements when the bonds between these elements are removed. By this model, we isolate a specific behavior; static, dynamic, logic etc., in each element.

Connections: This is a procedure to be followed in order to transform the model of the primitive system into the model of the connected system, which resemble either a subsystem or the total system. The procedure is generally based on the model of the primitive system and some connection objects. These connection objects identify connectivity constraints between system elements or subsystems. Connectivity constraints expressed by connection objects identify the conditions that the variables in the individual elements should comply with because of connection.

Connected system model: This is a description of the system after applying the process of connection on the model of the primitive system.

2.2. Physical System Modeling

As far as physical system modeling is concerned, our aim is to formalize the differential equations that describe the dynamic behavior of the system in array theoretical terms. The algebraic structure of this modeling process is represented in a computational structure known as Roth's diagram shown in Figure 2.3. This structure illustrates the correspondence between the physical behavior of systems variables and their algebraic structure.

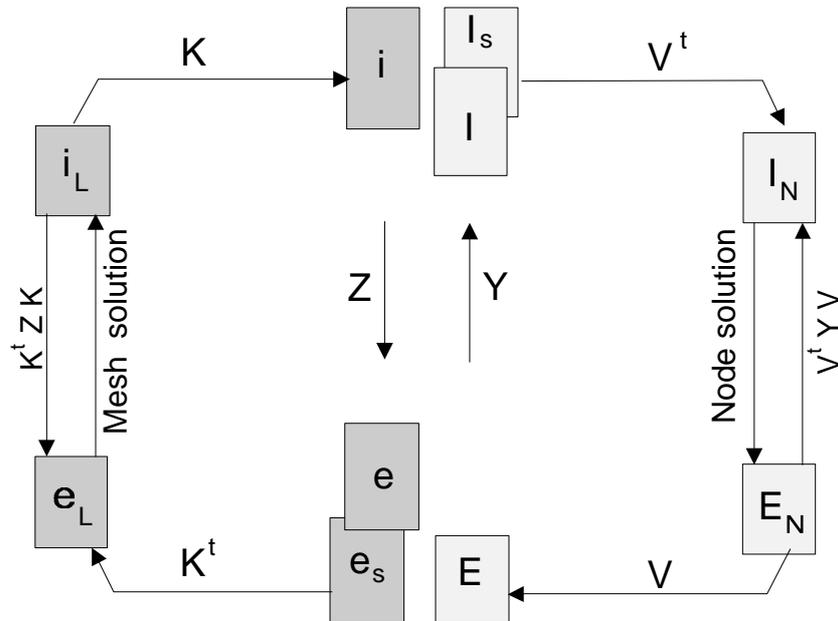


Figure 2.3. Roth's diagram

2.2.1. Physical primitive elements

The building blocks of the physical system are those elements that bind together two state variables *a flow variable* (i, q), and *a potential variable* (e, f). The property of each element (Y) is the quotient between the pair of these two state variables, such as mass, resistance, stiffness, capacitance, etc. This system could accommodate three categories of elements, as shown in Figure 2.4:

1. *Generalized damper*: e.g. electric resistor, mechanical damper, and hydraulic resistor.
2. *Generalized spring*: e.g. electric capacitor, mechanical spring, and hydraulic reservoir.
3. *Generalized mass*: e.g. electric inductor, mechanical mass, and hydraulic inductor.



Figure 2.4. Generalized elements

Primitive system model

Mathematically, the primitive system model Y_{abc} is a 3-object. It defines the property of the system in the state of no connection. The layers are arranged as follows:

Layer 1 contains the properties of those elements that are classified as generalized springs. The governing equations of layer 1 of the primitive system are expressed as follows:

$$f_b = Y_{1bc} x^c \quad (1)$$

Layer 2 contains the properties of elements that are classified as generalized dampers. The governing equations of layer 2 of the primitive system are expressed as follows:

$$f_b = Y_{2bc} \dot{x}^c \quad (2)$$

Layer 3 contains the properties of elements that are classified as generalized mass. The governing equations of layer 3 of the primitive system are expressed as follows:

$$f_b = Y_{3bc} \ddot{x}^c \quad (3)$$

The governing equations of the primitive system model could then be obtained by combining Equations (1), (2), and (3), yielding:

$$f_b = Y_{1bc} x^c + Y_{2bc} \dot{x}^c + Y_{3bc} \ddot{x}^c \quad (4)$$

The set of variables $(x^c, \dot{x}^c, \ddot{x}^c)$ represents generalized displacement, velocity, and acceleration respectively. The other set of variables f_b represents potential variables. These variables live inside the primitive system and are referred to as the *local variables*.

Example 2-1. Mass-spring-damper system

We will illustrate the procedure of setting up the primitive system model by a simple example, take for instance the mass-spring-damper system shown in Figure 2.5. The system contains all three categories of elements that might exist in any physical system.

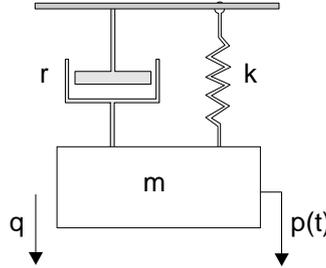


Figure 2.5. Mass-spring-damper system

The set of local displacements x^c is given by the vector $(x^1, x^2, x^3)^T$, the set of local velocities \dot{x}^c will then be given by: $(\dot{x}^1, \dot{x}^2, \dot{x}^3)^T$, and finally the set of local accelerations \ddot{x}^c will then be given by: $(\ddot{x}^1, \ddot{x}^2, \ddot{x}^3)^T$. The local forces in the elements f_b are given by the vector $(f_1, f_2, f_3)^T$.

The governing equations of layer 1 of the primitive system model for the mass-spring-damper system are:

$$\begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} = \begin{bmatrix} k & & \\ & 0 & \\ & & 0 \end{bmatrix} \begin{bmatrix} x^1 \\ x^2 \\ x^3 \end{bmatrix}$$

The governing equations of layer 2 are:

$$\begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} = \begin{bmatrix} 0 & & \\ & r & \\ & & 0 \end{bmatrix} \begin{bmatrix} \dot{x}^1 \\ \dot{x}^2 \\ \dot{x}^3 \end{bmatrix}$$

Finally the governing equations of layer 3 are:

$$\begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} = \begin{bmatrix} 0 & & \\ & 0 & \\ & & m \end{bmatrix} \begin{bmatrix} \ddot{x}^1 \\ \ddot{x}^2 \\ \ddot{x}^3 \end{bmatrix}$$

Collecting the three layers to set up the 3-object Y_{abc} , yields:

$$Y_{abc} = \begin{bmatrix} k \\ 0 \\ 0 \\ r \\ 0 \\ 0 \\ m \end{bmatrix}$$

The governing equations of the primitive system are obtained by applying Equation (4):

$$\begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} = \begin{bmatrix} k \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} x^1 \\ x^2 \\ x^3 \end{bmatrix} + \begin{bmatrix} 0 \\ r \\ 0 \end{bmatrix} \begin{bmatrix} \dot{x}^1 \\ \dot{x}^2 \\ \dot{x}^3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ m \end{bmatrix} \begin{bmatrix} \ddot{x}^1 \\ \ddot{x}^2 \\ \ddot{x}^3 \end{bmatrix} \quad (5)$$

2.2.2. Connections

While the variables $(x^c, \dot{x}^c, \ddot{x}^c)$ and (f_b) live inside the primitive system Y_{abc} , there exists a corresponding set of variables living inside the connected system Y_{ijk} which contains the property of the system when the elements are connected together. They are referred to as *global variables*.

The set $(q^j, \dot{q}^j, \ddot{q}^j)$ representing generalized displacement, velocity, and acceleration respectively, and the set (p_i) representing generalized potential. The transformation from the primitive system to the connected system is given by *Connection Objects*. Connection objects identify how the local variables in the elements are transformed into the global variables in the connected system.

In the linear case, this transformation could be defined directly based on the topological structure of the system. In this case, the connection object V_j^c called the incidence matrix is the only one needed in order to apply this transformation. It identifies incidence between the nodes and the branches in the topology graph.

In this case, the transformation from the set of variables $(x^c, \dot{x}^c, \ddot{x}^c)$ to the set of variables $(q^j, \dot{q}^j, \ddot{q}^j)$ will be given by:

$$x^c = V_j^c q^j \quad (6)$$

$$\dot{x}^c = V_j^c \dot{q}^j \quad (7)$$

$$\ddot{x}^c = V_j^c \ddot{q}^j \quad (8)$$

The transformation from (f_b) to (p_i) will be given by the adjoint transformation:

$$p_i = V_i^b f_b \quad (9)$$

The governing equations of the total connected system are then obtained by substituting the transformation laws given in Equations (6) (7), (8), and (9) into Equation (5), yielding:

$$p_i = (V_i^b Y_{1bc} V_j^c) q^j + (V_i^b Y_{2bc} V_j^c) \dot{q}^j + (V_i^b Y_{3bc} V_j^c) \ddot{q}^j \quad (10)$$

In compact form:

$$p_i = Y_{1ij} q^j + Y_{2ij} \dot{q}^j + Y_{3ij} \ddot{q}^j \quad (11)$$

For the given mass-spring-damper system, the topology graph of the system is shown in Figure 2.6.

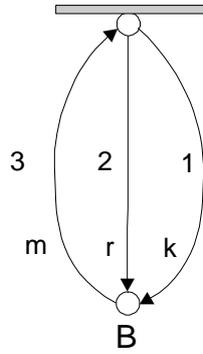


Figure 2.6. Topology graph of the mass-spring-damper system

The topological structure of physical system is based on nominal scale measurements. In that graph, each element is represented by a branch, the orientation of that branch is arbitrary, the branch-head node incidence is considered negative and the branch-tail node incidence is considered positive, the transformation matrix V_j^c could now be obtained:

$$V_j^c = c \begin{matrix} j \\ 1 \left[\begin{matrix} -1 \\ -1 \\ 1 \end{matrix} \right] \\ 2 \\ 3 \end{matrix} \quad (12)$$

And the transformation V_i^b :

$$V_i^b = i \begin{bmatrix} b & & \\ & 1 & 2 & 3 \\ & -1 & -1 & 1 \end{bmatrix} \quad (13)$$

By substituting Equations (5), (12), and (13) into Equation (10), we obtain the governing equation of the connected system for the mass-spring-damper system:

$$p(t) = xq + r\dot{q} + m\ddot{q}$$

In the above example, we have demonstrated a systematic procedure we will follow in order to construct a mathematical model for any physical system. In the following section, we will interpret this procedure geometrically.

Geometrically, the primitive system model is formed on three vector spaces with the variables (x^1, x^2, x^3) , $(\dot{x}^1, \dot{x}^2, \dot{x}^3)$ and $(\ddot{x}^1, \ddot{x}^2, \ddot{x}^3)$ as shown in Figure 2.7. Each vector space is a 3-dimensional Euclidean space. The property of the element is used to define the unit vector along the axis corresponding to the element in that space. The unit vectors along the reference frame is kept orthogonal in order to fulfill the requirement of no influence between elements. A point in the space now represents one state of the primitive system, and all points represent all possible states.

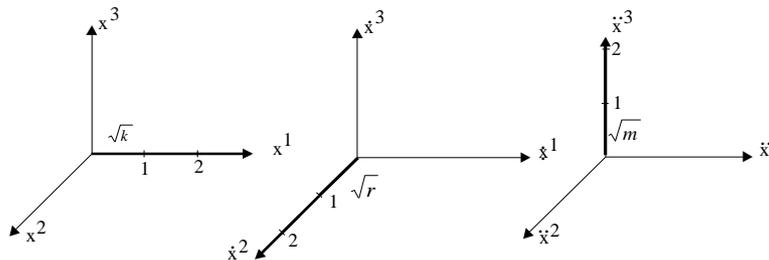


Figure 2.7. Geometrical representation of the primitive system model

The purpose of the connection is to reduce the theoretically infinite amount of possible states in the primitive system, to these states that comply with the constraints given by connection.

The mass-spring-damper system given in the example has only one degree of freedom, ($d = 1$), therefore the model of the connected system is formed on three vector spaces with the variables (q) , (\dot{q}) , and (\ddot{q}) . Each vector space is only 1-dimensional. All points in that space represent all possible states that the connected system may undertake, as shown in Figure 2.8.

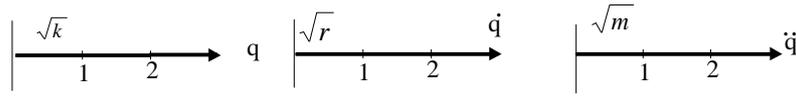


Figure 2.8. Geometrical representation of the connected system model

Connection object V_j^c is a mapping between the two spaces. It defines coordinate transformation between the two spaces. The complexity of this transformation depends on the nature of the surface the connected system is located at (flat, curved) and whether or not the reference coordinates systems are having moving or stationary axes. By solving the system, we isolate one state of all the states the connected system may undertake that comply with the constraints imposed by the sources.

2.3. Logical System Modeling - Array Based Logic

The logical decision-making is essentially a verbal description written in any natural language of the possible input combinations and the desired outputs or states resulting from the appropriate inputs. Such verbal description is supposed to fully describe what system response (output) we should obtain for various (input) combinations.

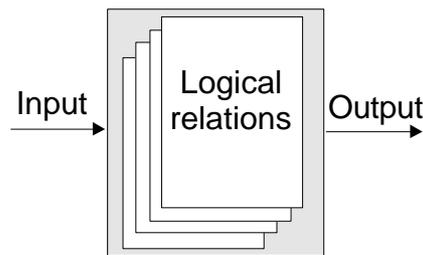


Figure 2.9. Logical system

As we have indicated before, our aim is to formalize these logical properties algebraically in array theoretic terms. Therefore, modeling task here is to synthesize an array-based mathematical model that *fully* captures these logical relationships.

Unlike physical systems, this relationship between input and output is highly irregular and can not be described by differential or difference equations. According to the relationship between input and output combinations, logical systems are generally classified into two basic branches, combinational and sequential.

Combinational logical systems are characterized by the fact, that output is dependent upon the present input and does not depend on the sequence of

the input excitations. Combinational systems represent the simplest form of logical systems.

In **sequential logical systems**, the output is a function of the current input and the present state of the system. This requires that a sequential system has a memory capability, where as a combinational system, being time independent require no memory. With the increasingly complicated operations in a sequential system, time synchronization could become necessary, resulting in a further break down of sequential systems into synchronous and asynchronous.

Asynchronous sequential systems are characterized by that, system outputs are generated as soon as input combinations appear. In other words, they are *event dependent*. In synchronous systems, all logic outputs and transitions occur at specified time intervals according to a global clock. Finally, asynchronous sequential systems, can be either deterministic, where the system always operates with the same sequence of input and output or non-deterministic where the input operates in random pattern. The second type of non-determinism result from small changes in process parameters.

Array based logic deals only with combinational and sequential asynchronous logical systems. Both categories can be grouped into one class of systems referred to as discrete event systems. A discrete event system is characterized by that input, output variables as well as the internal state change in discrete fashion and not continuously with time.

Since the general system model presented above is application neutral, it is therefore applicable to modeling logical systems as well. In logical systems, we are still capable of identifying the three major components in a system; elements, connections, and sources. Apparently, the nature of the elements involved, their connections, and the imposed sources are totally different from those of the physical system since they embody different aspects.

2.3.1. Logical elements

We define a logical element as the primitive part that carries a logical variable. Indeed, it is the logical variable which is of interest to us and not the element itself. This logical variable can in general take a finite set of N -unique states. This set identifies the universe of discourse for the variable and hence the domain of each element in the system model.

The domain [false, true] which corresponds to, for instance, [open, close] is thus a special case of the definition of a logical element with ($N = 2$).

Since the logical system is actually used to sense and control the physical system, then logical elements are in fact a collection of abstract representations on different levels of the physical system and its environment.

On one hand, elements of the logical system could represent an abstraction of some discrete physical elements such as switches, sensors, actuators, users, other systems and so on. For example, a switch in an electrical network could have a logical version in the logical system defined by a logical element with a variable that might take one of two states, [open, closed].

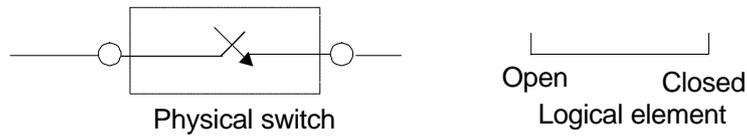


Figure 2.10. Physical switch and its logical version

Logical elements could also represent some important trajectories in the physical system. This higher level abstraction becomes necessary when implementing higher-level control strategy such as fault detection and recovery or adaptive control. In this case, logical variables take a finite set of N -non overlapping states, each state mirrors a region of unique measurements a continuous state variable in the physical system may take. For example, the voltage drop across a resistor in an electrical network could be mirrored to a logical variable called, for instance, *Drop* with the domain $[S_1, S_2, S_3]$ as shown in Figure 2.11.

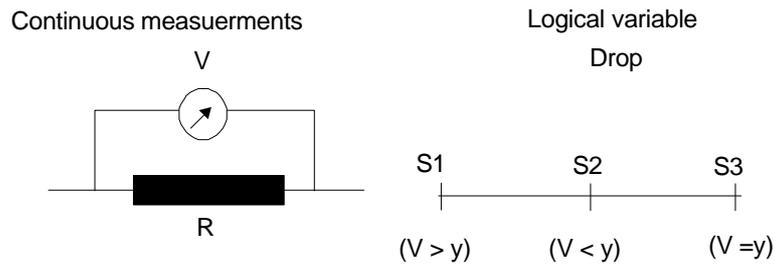


Figure 2.11. Logical variable constructed from continuous measurements

These three non-overlapping states could correspond to the continuous measurements: (the voltage drop is larger than y voltages), (the voltage drop is less than y voltages) and (the voltage drop is equal to y voltages) respectively.

Any logical variable with N -states can actually be reduced to N -variables where each variable can assume only two states [false, true]. For instance, the logical variable *Drop* shown in Figure 2.12, can be replaced by three variables (S_1 , S_2 , and S_3) where each variable can be either false or true.

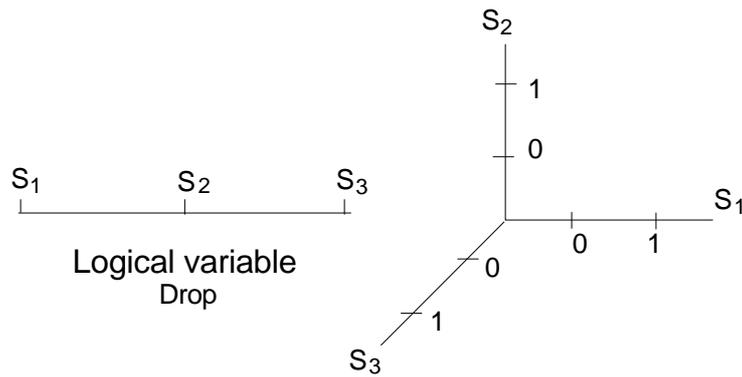


Figure 2.12. Equivalent representation of logical variables with 3-states

Logical variables may also represent abstractions of actions or directives to be applied on the physical system. Directives generated by the logical system may involve, switching continuous controller parameters, disconnecting faulty equipment, changing the set point for the continuous controller, or simply issuing commands to start a sequence of operations. All these directives are based solely on the logical decision making process that takes place in the logical system, which utilizes continuous as well as discrete measurements from the physical system and its environment. For example, the applied force on a mechanical system could have a logical version, called for instance, *Force* with the domain $[S_1, S_2]$. These states could be corresponding to the actions: (apply u_1), (apply u_2) where u_1 and u_2 are sequence of continuous control laws or even any constant real values that will bring the system into a desired mode.

Regardless of the number of states, in array-based logic each state is assigned with a distinctive binary combination. By assigning a binary combination for each state, we give more compact and powerful algebraic description to these logical variables.

2.3.2. Connections

The variables of the logical elements are connected together via logical connectives to form logical relations or premises P_i . The group of basic logical connectives includes three objects:

Connective	Operation	Symbol
Conjunction	AND	\wedge
Disjunction	OR	\vee
Negation	NOT	\neg

All logical operations such as implication and bi-implication that usually classified as basic connectives in classical logic have been excluded since they can be expressed by the above basic connectives.

The connected system model of the logical system P_c is then obtained by aggregating all the logical arguments P_i . In the case of a system where all the arguments must jointly be satisfied, the connected system P_c is found by connecting all the logical arguments using **conjunction** (\wedge):

$$P_c = P_1 \wedge P_2 \dots \wedge P_i$$

Each logical argument P_i is transformed into a multi-dimensional array by simple array operation (outer product). The connected system P_c is thus the truth table of the logical system expressed in a multi-dimensional array form. The number of axes in that array should be equal to the number of elements, therefore any duplicated axes must be eliminated by using the method *colligation*.

The state of each element in the array P_c must be a tautology to insure that the axes of the array are orthogonal, that is, the variables of the individual elements in the primitive system are independent of each other. The state of the connected system P_c must not be a contradiction to ensure that the system is consistent.

In physical systems, the properties of the system's elements and their connections govern the *dynamic behavior* of the overall system. Clearly, this line of thought can be extended to logical systems. In analogy to physical systems, a logical premise is indeed a connected system with its connectivity constraints are given by how the variables are connected via the basic logical connectives. The logical connectives in logical systems are therefore the equivalent to connection objects found in physical systems. They identify the transformation from the primitive system model to the connected system model.

2.3.3. Sources

One important condition we had to satisfy in the model of the connected system was to ensure that all the variables in the system were unbounded (tautologies). In the course of interacting with environment, the state of one or more of the variables in the logical system will be bounded to a particular state. Consequently, the connected system P_c will attain a new state P . This new state of the connected system satisfies both connectivity constraints *and* the external constraints.

This new state is obtained by performing conjunction between the source vector P_s and the unbounded connected system P_c :

$$P = P_c \wedge P_s$$

The state of all logical variables can then be determined by performing deduction on the connected system P , this is what is known as *inference* in classical logic, that is to drive information from the system. The conclusion is the response of the logical system to input source due to interaction with the environment.

Example 2-2. Conveyer system

In order to give a clearer picture of this modeling drama, consider the simple system shown in Figure 2.13. The system consists of a conveyer operated by an electrical motor. The conveyer is used to transport parts from end (a) to end (b). Two proximity sensors A and B are used to detect the presence or absence of parts at end (a) and (b) respectively.

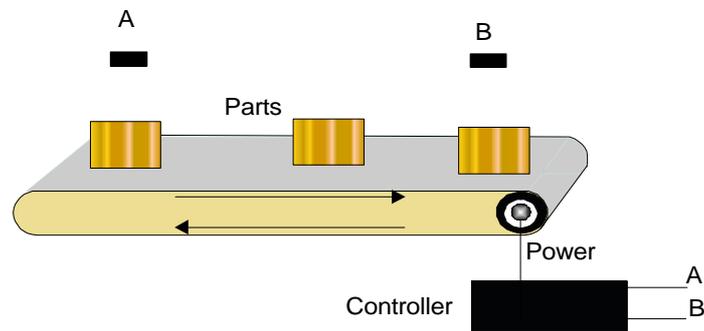


Figure 2.13. Conveyer system

The system works as follows:

Parts are loaded on the conveyer by an operator at end (a) and unloaded from end (b) by another process. When a part is detected at end (a), the power supply to the motor is switched on. The motor will then remain in this state until the part reaches end (b), then power supply will be switched off until the system detects a part at end (a), and then the cycle will be repeated.

Elements: Verbal description stated above describes the sequential relation between logical variables and identifies the functionality of the system.

We shall start by breaking up these verbal descriptions into number of logical elements that cover the entire state space of the logical system. These elements carry all the logical variables involved in the description of the primitive system model. The decomposition of the logical system is shown in Figure 2.14.

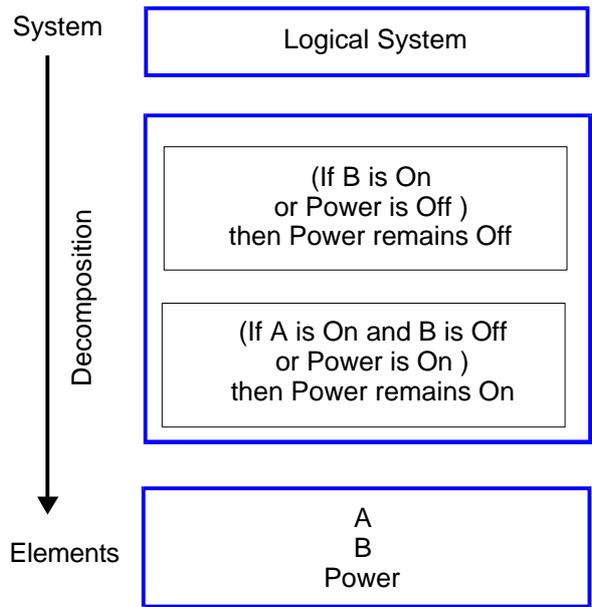


Figure 2.14. Decomposition of the logical system

In the system above, the primitive system mode can be described by the following variables:

Axis number	Variable name	Description of true state	Description of false state	Array representation
1	A (input)	Sensor A is On	Sensor A is Off	0 1
2	B (input)	Sensor B is On	Sensor B is Off	0 1
3	Power (output)	Power supply is On	Power supply is Off	0 1

Table 2.1. Description of the primitive system variables

The model of the primitive system is established by means of three orthogonal axes (A, B, and Power). The axes are orthogonal in order to reflect the fact that these variables are in the state of no connection.

Connections: starting from the lowest layer and moving upwards, each logical premise is transformed into a multi-dimensional array by performing outer product and eliminating repeated variables as shown in Figure 2.15.

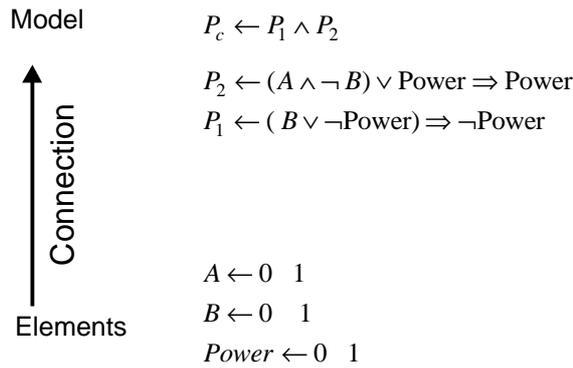


Figure 2.15. Connection of logical variables

Connected system model: The connected system model of the logical system which resemble the logical relation between the state of the power supply (*Power*) and the state of the sensors (*A*) and (*B*) is shown in Figure 2.15.

		Off		On		A
		Off	On	Off	On	Power
$P_c =$	Off	1	1	Off	0	1
	On	1	0	On	1	0
		B	B			

Figure 2.16. Array model of the logical relation between the sensor states and the output to the motor

This model expresses all the possible combinations of states after taking into consideration the logical relations between the input variables (*A* and *B*) and the output variable (*Power*). The model shows that when there are no parts detected at either end, the state of power supply could be either On or Off depending on the previous state of the power supply. Suppose that the current state of the power supply to the motor is On, and both (*A*) and (*B*) are bounded to the On state. The input source vector will then be given by:

$$P_s \leftarrow A \wedge B \wedge \text{Power}$$

As we indicated before, bounded variables should be seen as external sources on the logical system. The new state of the logical system is then obtained by performing outer product on the conjunction between the source array P_s and the model of the logical system:

$P \leftarrow P_c \wedge P_s$, yielding:

		Off		On		A
		Off	On	Off	On	Power
P =	Off	0	0	0	0	
	On	0	0	On	1	0
		B		B		

Figure 2.17. Logical system with bounded input

The new state of the variable (*Power*) is obtained by performing deduction on the first and second axes. The conclusion of the above logical system under the current input is: *Power supply is off*. The result of simulation the system for various input combinations and the resultant array model is shown in Figure 2.18, each input combination, the initial state of the power supply was off.

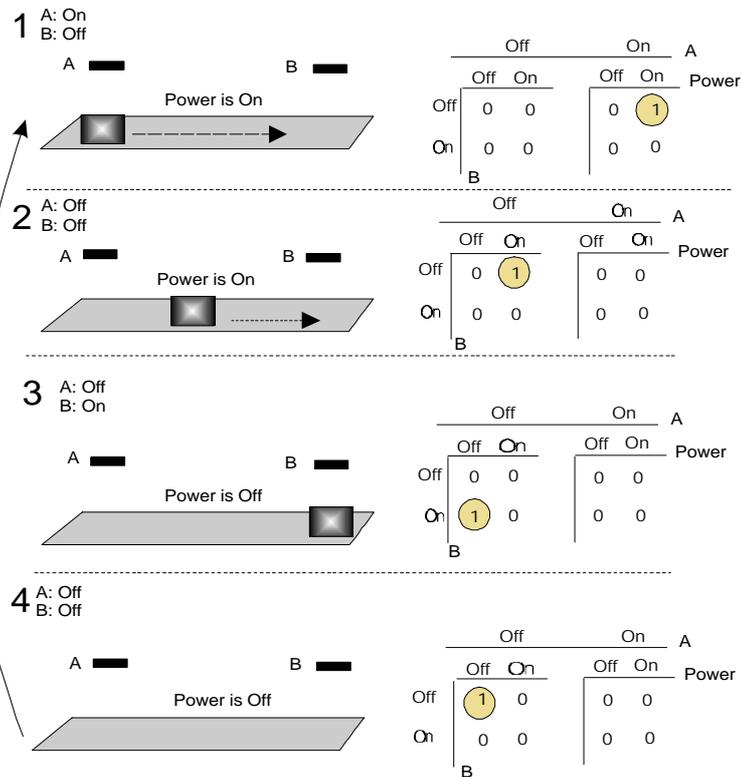


Figure 2.18. System simulation with different input combinations

2.4. Summary

A total system model could be viewed as two-level structure. The lower level is the physical system, where its state variables are evolving according to a set of differential equations. The higher level is the logical system that handles all the deductive reasoning activities in the system whose logical variables are abstract description of the physical system and its environment

A logical system is a verbal description of what we expect a given process or a system to perform. A logical model is an exact mathematical duplication of such verbal description. A supervisory controller is an inference engine that utilizes the mathematical model and input from environment in order to infer what should be done next. Regardless of the means of realizing the controller, it must always do exactly, what is required as reliably as is necessary at the lowest possible cost.

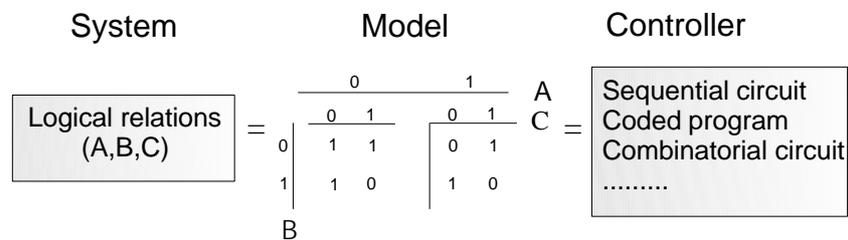


Figure 2.19. System, model and controller

Three fundamental operations must be performed in order to apply the array approach to logic. These operations are:

1. Outer product: used to transform a propositional form expressed in terms of propositions and connectives into a single multidimensional array.
2. Generalized colligation: used to eliminate duplicated variables from the multidimensional array so that the array will only contain independent axes.
3. Deduction: used to drive information from the array by projection on different axes.

3 INTERFACE

In the previous chapter, we have discussed the identifying features of mechatronics systems from a modeling perspective. We have shown that a mechatronics system could be seen as a hybrid system that consists of two basic systems; a higher level logical system which is responsible for all deductive reasoning activities and a lower level physical system which is controlled by the logical system. We have also presented an array-based formalism by which we established the mathematical models that identify these two systems.

We have indicated that the logical system is used to sense and control the physical system by issuing control directives that will bring the physical system to a desired state. For example to start or stop an activity, or to switch analog components into and out of use.

With reference to the general system model defined in the previous chapter, we have distinguished between three different terms:

Elements

Connections

Sources

Elements and connections together defined the property of the overall system. The sources defined the interaction between the system and its environment. If we strictly follow this system definition then the interface between the physical system and the logical system must be classified into two main categories:

1. Interface through sources. In this category, directives issued by the logical system are intended to alter the applied sources to the physical system.
2. Interface through elements and connections. In this category, directives issued by the logical system are intended to switch elements and subsystems into and out of the physical system. We shall denote this type of interface as *system interface*.

Interface through sources appears when the objectives of the logical system is restricted to take logical decisions to simply switch between different input sources for the part of the physical system that will perform a certain activity. In this case, the output from the logical system shall not contribute to altering the underlying dynamical structure of the physical system. Mathematically, this implies that state variables will remain continuous as long as the input is continuous. This category will be presented in this chapter.

System interface arises when the objective of the logical system is to issue directives that are intended to connect or disconnect elements and subsystems to the physical system. This sort of interface yields to altering the dynamical structure of the physical system and impose discontinuity on its behavior. Mathematically, this causes physical state variables to switch or jump between different modes of operations. The computational problems associated with the multiplicity of models will be explained further in chapter four.

3.1. Interface Through Sources

The objectives of the logical system in this case is restricted to take decisions to simply switch between different input sources for the part of the physical system that will perform a certain operation. The evolution of variables in physical system will remain independent of the logical system as long as traffic from the logical system is restricted to issuing commands that affects only the input to the physical system. Actual continuous control is left here to some sort of a continuous feedback controller. The architecture of the total system model with the present functional requirements is shown in Figure 3.1.

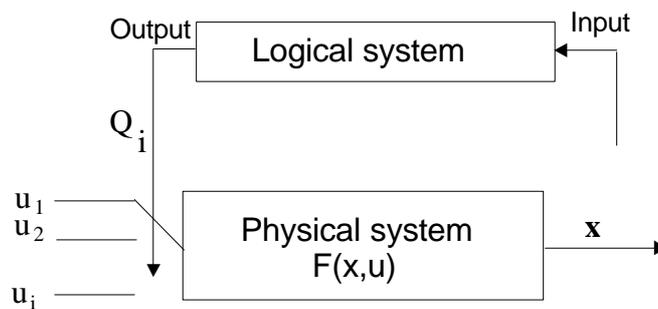


Figure 3.1. Interface through sources

Mathematically seen, this interface assigns each output signal from the logical system (Q_i) with a distinct real valued number (u_i) for the physical system to use.

The basic feature of the adopted array representation is the possibility of treating all objects involved in the description of the system in the same way independent of size and form and shape. This feature allow us to mix together the output from the logical system and the input to the physical system in one single hybrid geometric object that contains real, as well as binary variables.

The hybrid combination of these signals from different domains is what known in array theory as a *strand object*. A strand object is a text object, in each step of simulation process, its corresponding numeric value has to be

computed taking into account the instantaneous numeric values of its real and binary variables.

3.2. Application Example: A Manufacturing System

Given the manufacturing system shown in Figure 3.2. It consists of a workstation, which is a single-axis-boring machine. The workstation consists of a boring spindle operated by a direct current servomotor. The linear motion of the boring spindle is carried out by means of a hydraulic linear actuator. The hydraulic actuator is powered by a constant pressure hydraulic pump and the volumetric flow in the hydraulic circuit is controlled by a servo valve S_t [39].

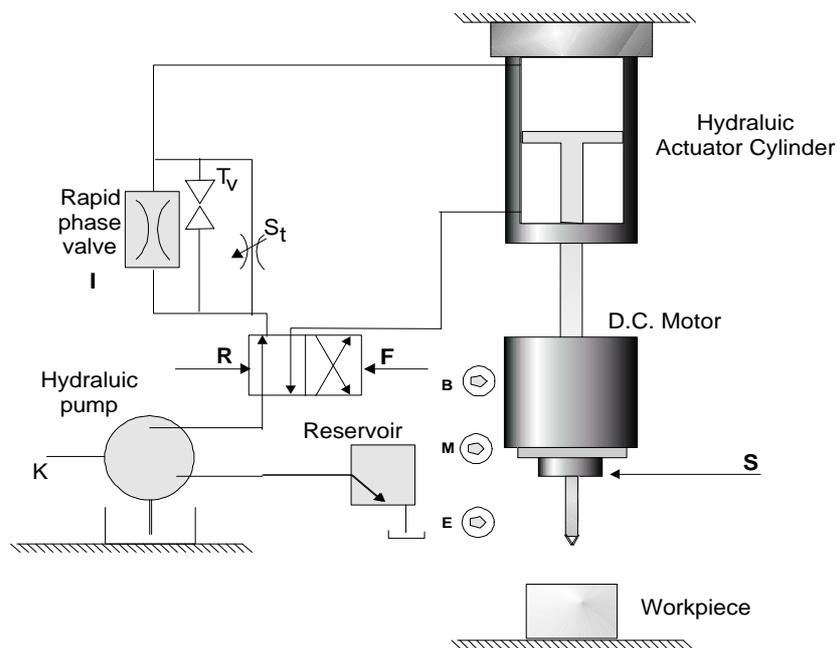


Figure 3.2. Application example: A manufacturing system

The operation of the system is governed by three sensors (micro-switches) B , M , and E which are located in the manufacturing system as displayed in Figure 3.2.

1. Sensor B indicates that the boring spindle is at the rear position.
2. Sensor M indicates that the boring spindle has reached the feeding position.
3. Sensor E indicates that the spindle has reached its final destination, and ready for backwards motion.

Starting from an initial state, the operator switches on the system by a signal (K) which is a very short signal. The boring spindle is at the rear position and the workstation is ready for machining the workpiece. Once the workstation started processing, it can not be interrupted until it completes one whole cycle. The manufacturing system will go between three modes of operations:

1. Starting from this idle state, hydraulic circuit will open rapid phase valve (I), and the workstation will start forward motion by opening the valve (F).
2. At position (M), the rapid phase valve (I) will be switched off in order to start a controlled feed forward motion. This motion is regulated manually by the servo valve S_r . At position (M) the spindle motor will also be switched on by a signal (S).
3. At position (E) the backward motion (R) will begin, simultaneously the rapid phase valve (I) will be switched on.

3.2.1. System modeling

The complete derivation of the logical system and physical system models is enclosed in **Appendix A**. Here we present only the final models and carry out combined simulation in order to verify system requirements. The model of the logical system can be described by the following variables:

Axis number	Variable name	Description of true state	Description of false state
1	K	Start	Stop
2	B	Sensor B is On	Sensor B is Off
3	M	Sensor M is On	Sensor M is Off
4	E	Sensor E is On	Sensor E is Off
5	F	Forward motion is On	-
6	S	Turn on spindle motor	Turn off spindle motor
7	I	Turn on rapid motion	Turn off rapid motion
8	R	Backward motion is On	-

Table 3.1. Variables involved in the description of the logical system

The model of the physical system is described by the following set of second order differential equations, given here as two-dimensional array:

$$\begin{bmatrix} e_f \\ e_a \\ T_L \\ x \\ -F_L \end{bmatrix} = \begin{bmatrix} r_f + L_f s & 0 & 0 & 0 & 0 \\ 0 & r_a + L_a s & k_m & 0 & 0 \\ 0 & -k_e & r_z + (j_z + j_L)s & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{k_x}((g + k_d) + Cs) & \frac{A_p}{k_x} \\ 0 & 0 & 0 & -A_p & (r_p + r_L) + (m_L + m_p)s \end{bmatrix} \begin{bmatrix} i_f \\ i_a \\ \mathbf{w} \\ p \\ v \end{bmatrix}$$

Where:

Symbol	Description	Symbol	Description
e_f	Applied voltage on stator windings	p	Differential pressure inside the actuator cylinder
e_a	Applied voltage on armature winding	v	Linear speed of the actuator cylinder
T_L	Applied torque on the rotor shaft	r_z	Viscous damping of the rotor shaft
x	Linear displacement of the servo valve	r_f	Electrical resistance of stator windings
F_L	Applied force on the cylinder actuator shaft.	r_a	Electrical resistance of rotor windings
i_f	Electric current in stator windings	I_z	Moment of inertia with load inertia
i_a	Electric current in rotor windings	L_f	Inductance of the stator windings
w	Angular rotation of the rotor shaft	L_a	Inductance of the rotor windings
$r_p + r_L$	Resistance to linear acceleration caused by the mass of the piston and the load	g	Leakage constant
$m_p + m_L$	Mass of the load and piston	C	Constant
k_x, k_d	Flow and pressure gradient of the servo valve	A_p	Working area of the actuator piston

As already indicated, the above system contains two independent state spaces. The physical system and the logical system interact indirectly through the sensors (B , M , and E) located in the environment of the physical system. These sensors generate discrete signals for the logical system to use. The logical system then generates discrete output signals (F , R , I and S) that will alter the set points for the physical system and bring it to desired phase of operation.

According to the combination of these signals, the input source to the physical system (I_s) will switch between four set points as shown in Figure

3.3. Each set point will bring the physical system to a specific mode of operation.

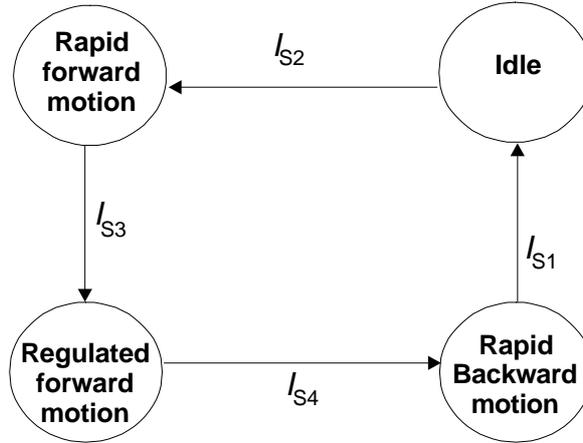


Figure 3.3. Set points for the physical system

The switching between these set points is performed according to the following combinations of the output signals from the logical system:

$$I_S = \begin{cases} I_{s1} & \text{if } \neg(F \vee R) \\ I_{s2} & \text{if } (F \wedge I) \\ I_{s3} & \text{if } (F \wedge S) \\ I_{s4} & \text{if } (R \wedge I) \end{cases} \quad F, R, S, I \in [0 \ 1]$$

Where:

$$I_{s1} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad I_{s2} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ x_1 \\ -F_L \end{bmatrix} \quad I_{s3} = \begin{bmatrix} e_f \\ e_a \\ T_L \\ x_2 \\ -F_L \end{bmatrix} \quad I_{s4} = \begin{bmatrix} e_f \\ e_a \\ T_L \\ x_3 \\ -F_L \end{bmatrix}$$

These four set points can be easily combined in one single strand object ($I_S D$) which contains real variables as well as the binary variables from the logical system:

$$I_S D = \begin{bmatrix} e_f \times S \\ e_a \times S \\ T_L \times S \\ x = ((x_1 \times F) + (x_2 \times (F \wedge S)) + (x_3 \times R)) \\ -F_L \times (F \wedge R) \end{bmatrix} \quad S, F, R \in [0,1]$$

The first three variables in the array represent applied input to the spindle motor, evidently they will be activated if and only if the logical system turns on the signal S .

The signals (F , R , and S) are combined with the displacement of the servo valve (x) so that the physical system will apply the right amount of flow at each mode of operation. This yields to:

$$x = ((x_1 \times (I \wedge F)) + (x_2 \times (F \wedge S)) + (x_3 \times R))$$

Where:

x_1 is the amount of flow through the hydraulic circuit during rapid phase forward motion.

x_2 is the amount of flow through the hydraulic circuit during regulated phase forward motion.

x_3 is the amount of flow through the hydraulic circuit during backwards motion.

Finally, eventual disturbances on the hydraulic system will only take place if the system is either in the backward or forward mode:

$$F_L = F_L \times (F \vee R)$$

In order to investigate the overall behavior of the manufacturing system, simulation of the manufacturing system will be consisting of two simulation loops as shown in Figure 3.4. The first loop is for solving the logical system, that is, to determine the response of the logical system to the input from sensors. The output from the logical system is then accessed to the second loop. In the second loop, the input source vector $I_s D$ will be updated and its numerical value will be computed. Simulation of the physical system is then carried out by solving numerically the differential equations representing the dynamic behavior of the manufacturing system.

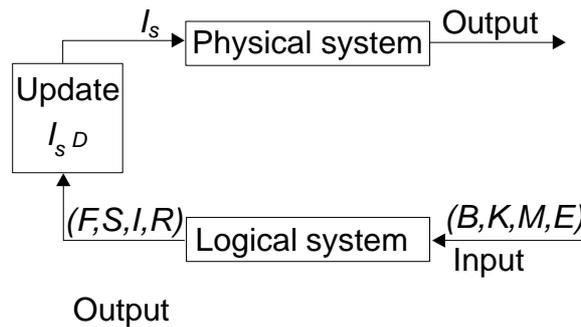


Figure 3.4. Simulation loops for the manufacturing system

3.2.2. Simulation

As indicated in the previous section, the manufacturing system will switch between four modes of operation during one complete cycle of operation. We shall start simulation by considering each mode separately and then simulation for the whole cycle will be carried out.

Rapid phase feed forward motion

Starting from the idle mode when the whole system is at rest and the boring spindle is at the rear position and the user has just pressed start button. The state of input signals to the logical system will be bounded to the following states:

K	B	M	E
On	Off	Off	Off

Table 3.2. State of input signals to the logical system

The state of the output signals are determined by applying deduction on the multidimensional array (P), this will give us the following output signals:

F	I	R	S
On	On	Off	Off

Table 3.3. State of output signals: rapid phase feed forward motion

In the second loop, this combination of output signals will be used to compute the numerical value of the input source vector to the physical system model:

$$I_s = I_{S2} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ x_1 \\ -F_L \end{bmatrix}$$

The result for simulating the physical system in the second loop at this mode of operation for the linear velocity (v) and the differential pressure (p) inside the hydraulic actuator is shown in Figure 3.5.

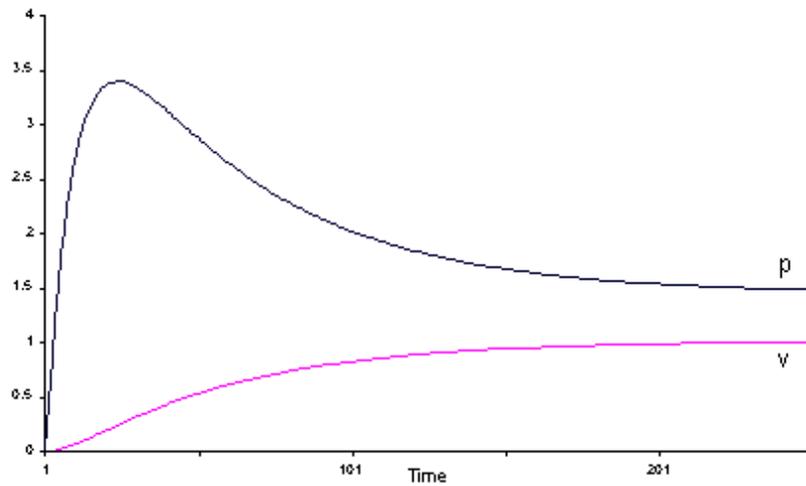


Figure 3.5. Linear velocity (v) and differential pressure (p) in the actuator cylinder during rapid phase feed forward motion

Controlled feed forward motion

In this mode, the workstation will switch from a rapid phase motion to a regulated feed forward motion. This regulation is controlled by the servo valve S_r . Accordingly, the state of input signals to the logical system will be as follows:

B	M	E
Off	On	Off

Table 3.4. State of input signals for controlled feed forward motion

The states of the output signals are determined by consulting the logical system. The logical system will respond by issuing the following output signals:

F	I	R	S
On	Off	Off	On

Table 3.5. State of output signals: controlled feed forward motion

In the second loop, this combination of output signals will cause the physical system to switch to the following set of input sources:

$$I_s = I_{s3} = \begin{bmatrix} e_f \\ e_a \\ T_L \\ x_2 \\ -F_L \end{bmatrix}$$

Simulation of the linear velocity, the differential during the controlled feed forward motion is shown in Figure 3.6.

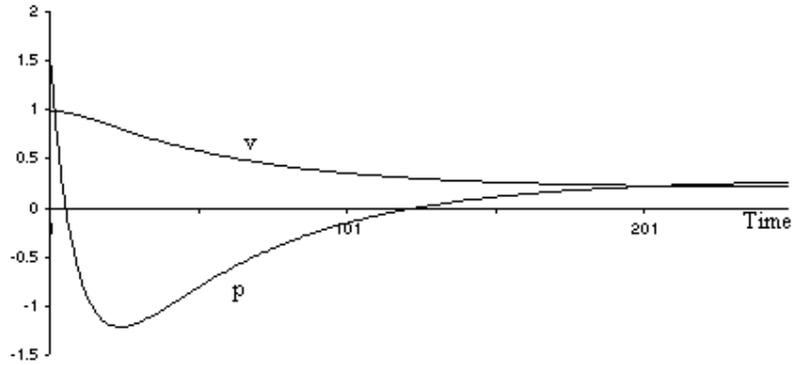


Figure 3.6. Linear velocity (v) and differential pressure (p) in the actuator cylinder during controlled feed forward motion

Backward motion

Provided that the boring spindle has finished boring the work piece. Accordingly, the state of input signals to the logical system will be as follows:

B	M	E
Off	Off	On

Table 3.6. State of input signals: Backward motion

The state of the output signals from the logical system:

F	I	R	S
Off	On	On	On

Table 3.7. State of output signals: Backward motion

In the second loop, this combination of output signals will cause the physical system to switch to the following set of input sources:

$$I_s = I_{s3} = \begin{bmatrix} e_f \\ e_a \\ T_L \\ x_3 \\ -F_L \end{bmatrix}$$

In this phase of motion, the rapid phase valve will be opened and the backward motion will start as shown in Figure 3.7.

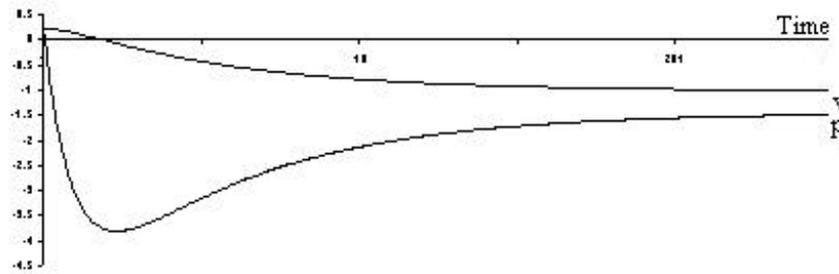


Figure 3.7. Linear velocity (v) and differential pressure (p) during backward motion

Simulation of the whole cycle for the differential pressure and linear velocity of the actuator cylinder is shown in Figure 3.8.

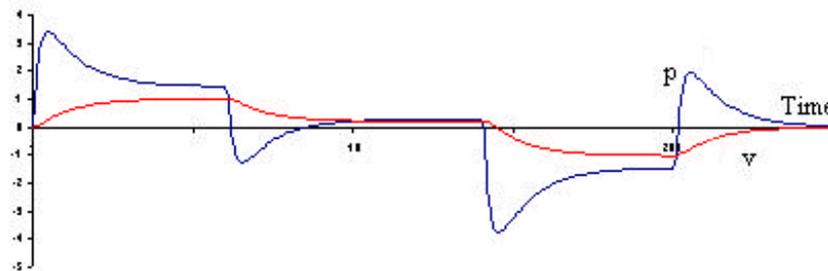


Figure 3.8. Differential pressure (p) and linear velocity (v) of the actuator cylinder through one cycle of operation

We can summarize the most visible advantages of this combined simulation by the following:

1. Verification of the overall system requirements, that include dynamic as well as logical requirements.

2. Determination of the minimum parameters (x_2 and e_a) which would assure satisfactory continuous control (relatively quick adjustment to reference angular velocity for the spindle motor or adjustment to reference linear velocity for the hydraulic cylinder.)
3. Adjusting physical system parameters or the location of micro breakers in order to avoid switching from one mode to another during transient periods.

3.3. Summary

The practical advantage of using multidimensional arrays to describe the physical as well as the logical properties is that combined simulation of both systems can be carried out concurrently using any array oriented simulation environment. In presenting the interface through sources, we saw that the output from the logical system affects only the input to the physical system. Logical system output does not contribute to altering the underlying dynamical structure of the physical system. The logical system merely switches *on* or *off* the power supply to the part of the physical system that will perform an activity. The actual continuous control is then left to some sort of a feedback controller.

4 SYSTEM INTERFACE

System interface arises when the objective of the logical system is to issue directives that will connect or disconnect elements and subsystems into and out of the physical system. This sort of interface yields altering the dynamical structure of the physical system and impose discontinuity on its behavior. Mathematically, this causes the physical system to switch between different modes of operations possibly accompanied by jumps in its state variables. This explains why the term mode-switching systems is often used in this context.

Mathematically, mode-switching systems are described by multiple set of equations. Each particular mode is associated with one set of equations describing the continuous behavior at that particular mode as shown in Figure 4.1.

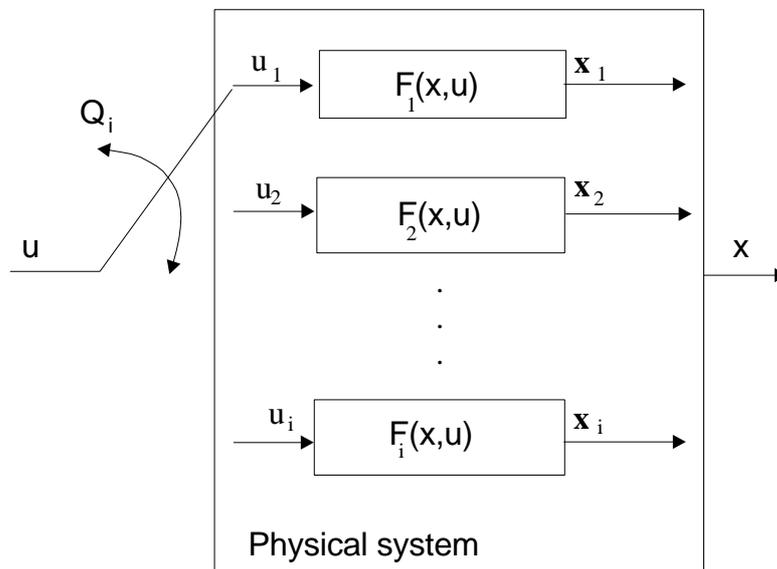


Figure 4.1. Physical system with multiple mode models

This multiplicity will evidently complicate both modeling and simulation, particularly if we consider the number of models required for describing all modes of operations. From computational viewpoint, there are two main problems associated with the multiplicity of models:

1. The initial value problem
2. The variable state space dimension

These two problems are explained further in the following sections.

4.1. Initial Value Problem

Each mode model operates as a continuous system described by the usual differential or difference equations. When the logical system switch to the next mode model there is a jump discontinuity from the end of application of the old mode to the beginning of application of the next mode. In order to guarantee smooth transition (bump-less switching) from one mode to the next mode, the state of the continuous system prior switching must probably be transferred to the continuous system after switching [9]. We shall illustrate what we have in mind by the following example.

Example 4-1. Mass element on a friction-less surface

Consider the second order dynamic system shown in Figure 4.2, which consists of mass element (m) moving along a friction-less surface. Suppose that the input applied to the mass element is a continuous function of time (t) and given by:

$$f = \cos\left(\frac{p}{2}t\right)$$



Figure 4.2 A dynamic system consisting of mass element moving on a friction-less surface

Suppose further that a higher level logical system is used to switch between two continuous controllers at ($t=10$) to change it from

$$y_{c1} = r_1s + k \quad \text{to} \quad y_{c2} = r_2s + k$$

Where r_1, r_2 are damping gain constants and k is stiffness gain constant. Accordingly, the dynamical system will at ($t=10$) switch between two models:

$$f = ms^2 + r_1s + k \quad \text{and} \quad f = ms^2 + r_2s + k$$

If the state of the second model was set to zero at the instant of switching, then the bumpy output of Figure 4.3 will result.

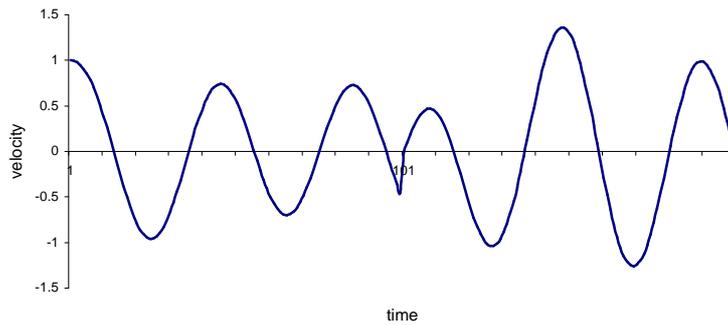


Figure 4.3. There is a discontinuity in the output when we switch the control law without setting the state

The modeling framework we propose here, solve this problem and ensure smooth transition from one mode to another without having to setting the state of the system at each transition. This is achieved by formulating a hybrid model that explicitly contains all modes of operations the system undergoes.

4.2. Variable State Space Dimension

Note that in the previous example, state-space dimension of the dynamic system remained invariant to switching. In some applications, switching may cause the state vector at one mode to jump to a higher or lower order. This implies that some state variables could no longer be observable or controllable from that mode. Variable state space dimension is also a simulation problem since there is no numerical integration algorithm that can handle variable state space dimension. We shall illustrate what we have in mind by the following example.

Example 4-2. Electrical network with a switch

An electrical network is shown in Figure 4.4. The network is driven by a single voltage source. The network consists of four elements and one ideal switch (S_w). The switch can be toggled, by a higher process, between two states: [open, close].

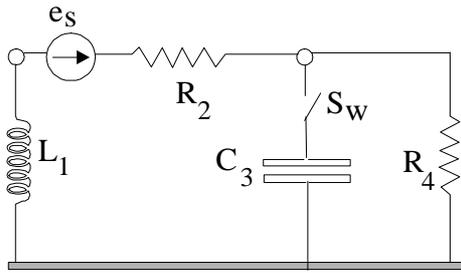


Figure 4.4. Electrical network with a switch

The system with the present configuration will *jump* between two distinctive state space models. Each model describes the dynamic behavior of the network for each switch state. The first mode model (M_1) is activated when the switch is in the open state. The property graph for this mode is shown in Figure 4.5.

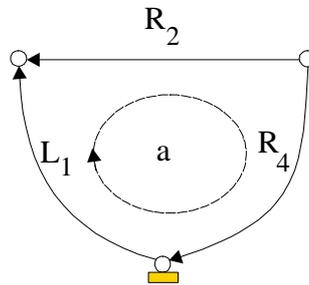


Figure 4.5. Property graph of the circuit when the switch is open

In this mode, the circuit consists only of three elements L_1, R_2, R_4 forming the closed path (a). If we select element 2 and element 4 to form a tree in the graph, then the corresponding mesh connection matrix of the circuit would be given by:

$$K = \begin{matrix} a \\ 2 \\ 4 \end{matrix} \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix}$$

The primitive system model of the circuit in the present mode is given by property matrix Z :

$$Z = \begin{bmatrix} L_1 s & & \\ & R_2 s & \\ & & R_4 s \end{bmatrix}$$

Where $s = \frac{d}{dt}$ is a differential operator.

The connected system model of this mode of operation is obtained by applying the transformation:

$$Z_L = K^t Z K$$

This gives us the second order differential equation:

$$R_2 e_s = L_1 \ddot{q}_a + (R_2 + R_4) \dot{q}_a$$

The state space model of this mode of operation can be obtained by transforming the above second order differential equation into two first order differential equations:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{R_2 + R_4}{L_1} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \frac{R_2 e_s}{L_1} \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

The components of the state vector \mathbf{x} that describe the behavior of the network at this mode represent:

- x_1 : is the generalized displacement (charge in electrical networks) generated in the closed path (a) formed by the elements L_1, R_2 and R_4
- x_2 is the generalized flow (current in electrical networks) generated in the closed path (a) formed by the elements L_1, R_2 and R_4

The second mode of operation (M_2) is activated when the switch is in the closed state. In this mode, the circuit will be consisting of two closed paths as shown in Figure 4.6. Closed path (b) formed by elements L_1, R_2 and C_3 and closed path (c) formed by elements C_3 and R_4

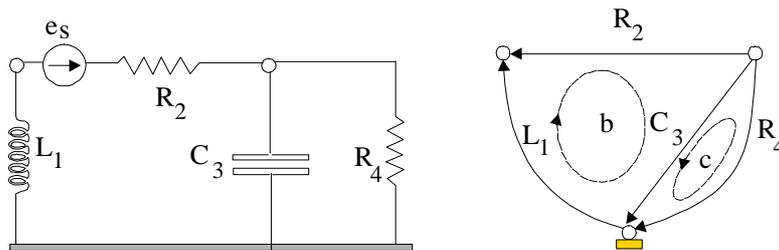


Figure 4.6. Circuit layout and its property graph when the switch is closed

If we select element R_2 and element R_4 to form a tree in the graph, then the corresponding mesh connection matrix of the graph would be given by:

$$K_2 = \begin{matrix} & \text{b} & \text{c} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ -1 & -1 \end{bmatrix} \end{matrix}$$

Following the same procedure as we did to derive the model of the first mode of operation, we obtain the following set of equations that describe the behavior of the network when the switch is in closed state:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ -\frac{1}{L_2 C_3} & -\frac{R_2 + R_4}{L_2} & \frac{1}{L_2 C_3} \\ \frac{1}{C_3 R_4} & 0 & -\frac{1}{C_3 R_5} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + e_s \frac{1}{L_2} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

In the second mode of operation, when the switch is closed, the components of the state space vector \mathbf{x} represent:

- x_1 : is the generalized displacement generated in the closed path (b) formed by the elements L_1, R_2 and C_3
- x_2 : is the generalized flow generated in the closed path formed by the elements L_1, R_2 and C_3
- x_3 : is the generalized displacement generated in the closed path formed by the elements C_3 and R_4

Providing that the switch was initially closed then it is easy to observe that toggling the state of the switch to open state will contribute to a rank reduction of the state vector. The length of the state variable vector \mathbf{x} will be reduced to two variables, eliminating x_3 from the mathematical model, although the variable itself did not actually die out, it is just disappeared from the state space model.

The problem will be magnified when a transition from (M_1) to (M_2) takes place. If such transition occurs, the eliminated state variable must be reinitialized in such manner that the energetic interaction within the structure remains preserved.

One side of the problem is the initial value problem discussed in Example 4-1. The other side of the problem is related to observability and controllability of the network. For instance, state variable x_3 is neither observable nor controllable from the first mode of operation. As we have indicated before, this is also a simulation problem since there is no numerical integration algorithm than can handle variable state space dimension.

4.3. Switching Elements

The previous examples show that directives from the logical system are either realized by physical switching elements as we have seen in Example 4-2. Examples of this category from physical domains are; electrical switch, mechanical clutch and hydraulic valves as shown in Figure 4.7.

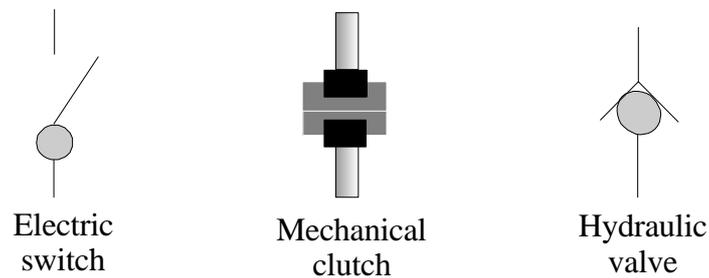


Figure 4.7. Switching elements from multiple physical domains

These directives could just as well be performed in a digital medium, as we have seen in Example 4-1 without having to use physical switching elements. What is common in both cases is indeed the concept of presence or absence of these logical directives and not the medium that executed switching. Therefore, in the modeling framework we present here, every logical directive shall be represented by a switch element that can assume one of two states, [On or Off]. In turn, the state of the switch element is controlled by the higher level logical system.

4.4. Model Formulation

Researchers have proposed two possible approaches to fix the problems arising from multiplicity of models. First, by employing ideal switches with variable circuit topology. The conditions for transition from one mode to another as well as initialization rules associated with each transition are then included in a global discrete structure [22,52]. Although this approach offers a great deal of flexibility, its main disadvantage is that all modes of operations the continuous system could undergo as well as initialization rules for each transition must be identified in advance.

The other approach is to employ non-ideal switches with constant circuit topology, this approach actually solves the initial value problem and system's state space dimension remains invariant to switching. However, it fails to represent systems with ideal switching case.

Our objective is to present a modeling framework that automatically will generate a hybrid continuous system model. This model explicitly contains all modes of operations the physical system undergoes. In this formulation, the continuous model contains real as well as logic variables from the binary domain. These logic variables represent the directives issued by the logical system to drive the physical to the desired mode. Topologically, these logical directives will be substituted by explicit switching elements in the physical system. We begin in chapter 5 to present the first approach which uses non-ideal switching elements.

5 NON-IDEAL SWITCHING ELEMENTS

In order to encounter the problems arising from multiplicity of models, one possible approach to solve this problem is to employ non-ideal switching elements with artificial resistance. In this representation, the switch in its closed position is represented by a very small but not vanishingly small resistance and in its open position by a very large resistance.

In section 1.2.3, we introduced the concept of switching elements to represent the interface between the physical and logical system. Mathematically, the state of the switch was represented by a binary variable (S_w) to indicate that the switch is in the [open] state. The complement of this binary variable was then be labeled (\tilde{S}_w) and was used to indicate that the switch is in the [closed] state.

$$\tilde{S}_w = \text{not}(S_w)$$

By using this binary variable, we could include the switch element as a part of the primitive system model with the following property:

$$y = (R_c \tilde{S}_w) + (R_o S_w)$$

When the switch is closed, we get ($\tilde{S}_w = 1$) and we obtain the following property:

$$y = R_c \cong 0$$

This is similar to having an element connected between two nodes with very small resistance. This resistance element will act as a short circuit between the two nodes and cause the potential of the two nodes to be identical. When the switch is open, we get ($\tilde{S}_w = 0$) and we obtain the following property:

$$y = R \cong \infty$$

This is similar to having an element connected between two nodes with very high resistance. This corresponds to having an open circuit between the two nodes causing the inflow and outflow to be equal to zero.

Example 5-1. Mechanical system with non-ideal clutch

In order to illustrate the process of modeling with non-ideal switching elements, consider the mechanical system shown in Figure 5.1. It consists of two axial shafts that are connected via a mechanical clutch. The corresponding property graph of the system is shown to the left of the mechanical system. The clutch element in this example is replaced by a damper with an artificial damping.

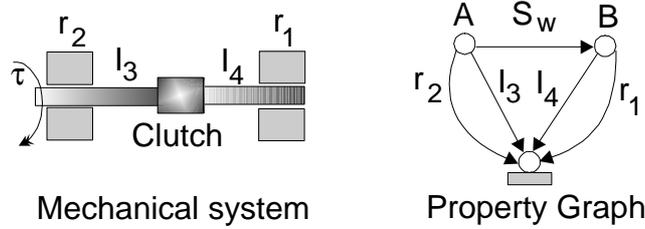


Figure 5.1. Mechanical system with non ideal clutch

The primitive system model of the mechanical system is given by the property matrix Y :

$$Y = \begin{bmatrix} r_1 s & & & & \\ & r_2 s & & & \\ & & I_3 s^2 & & \\ & & & I_4 s^2 & \\ & & & & (\tilde{S}_w r_c s) + (S_w r_o s) \end{bmatrix}$$

From the property graph, the corresponding connection matrix is given by:

$$V = \begin{matrix} & \begin{matrix} A & B \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} & \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ -1 & 1 \end{bmatrix} \end{matrix}$$

The corresponding set of equations that describe the system is obtained by applying the transformation ($Y_N = V^t Y V$) yielding the set of second order differential equations:

$$\begin{bmatrix} \mathbf{t}_A \\ 0 \end{bmatrix} = \begin{bmatrix} (r_2 s + I_3 s^2) + r_c s \tilde{S}_w + r_o s S_w & -r_c s \tilde{S}_w + r_o s S_w \\ -r_c s \tilde{S}_w + r_o s S_w & (r_1 s + I_4 s^2) + r_c s \tilde{S}_w + r_o s S_w \end{bmatrix} \begin{bmatrix} \mathbf{q}_A \\ \mathbf{q}_B \end{bmatrix}$$

When the clutch is engaged, we get ($\tilde{S}_w = 1$), and the differential equations of the system become:

$$\begin{bmatrix} \dot{\mathbf{t}}_A \\ 0 \end{bmatrix} = \begin{bmatrix} (r_2s + I_3s^2) + r_c s & -r_c s \\ -r_c s & (r_1s + I_4s^2) + r_c s \end{bmatrix} \begin{bmatrix} \mathbf{q}_A \\ \mathbf{q}_B \end{bmatrix}$$

Where: $r_c \cong \infty$

When the clutch is disengaged, we get ($\tilde{S}_w = 0$), and the differential equations describing the system become:

$$\begin{bmatrix} \dot{\mathbf{t}}_A \\ 0 \end{bmatrix} = \begin{bmatrix} (r_2s + I_3s^2) + r_o s & -r_o s \\ -r_o s & (r_1s + I_4s^2) + r_o s \end{bmatrix} \begin{bmatrix} \mathbf{q}_A \\ \mathbf{q}_B \end{bmatrix}$$

Where $r_o \cong 0$

The logical conditions for switching are embedded in a higher level logical system. The logical conditions for switching is assumed to be time-related condition in which the clutch will change its state according to the following condition:

$$S_w = \begin{cases} \text{disengaged} & (0 \leq t \leq 0.6) \text{ Or } (1.2 \leq t \leq 1.7) \\ \text{engaged} & \text{otherwise} \end{cases}$$

The result of the combined simulation for the angular rotation and angular velocity of the two axial shafts is shown in Figure 5.2 and Figure 5.3 respectively.

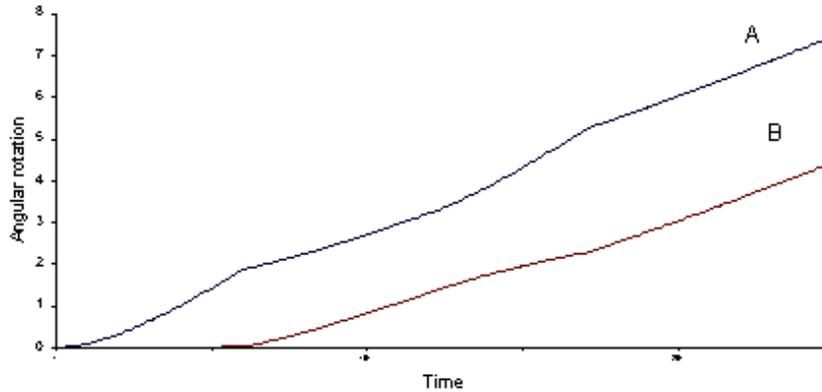


Figure 5.2. Angular rotation of the two axial shafts (node A and node B)

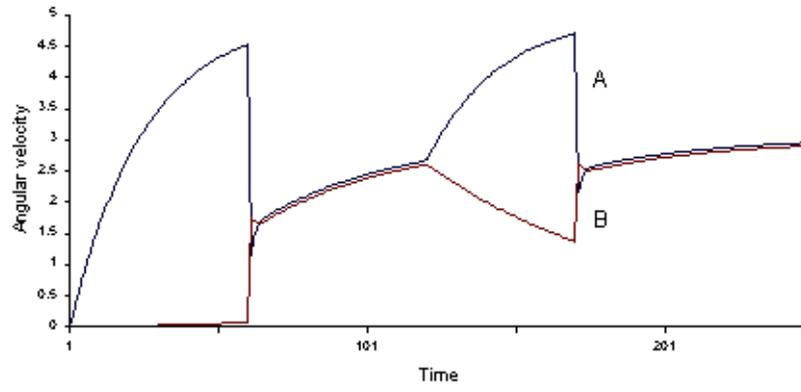


Figure 5.3. Angular velocity of the two axial shafts (node A and node B)

5.1. Application Example: Gearbox

The gearbox described here is presented briefly in [22]. The gearbox is constructed to prevent freewheeling when changing gears. The gearbox is controlled by changing the forces pushing the plates in the clutches towards each other. The gearbox has a forward gear, a backward gear, and a number of different gearing for both directions. The load is representing the weight of the vehicle. In the current example, we only use two clutches for two different gearing in the forward direction. The forward gear is assumed to be already engaged. A simplified schematic diagram of this part of the gearbox is shown in Figure 5.4.

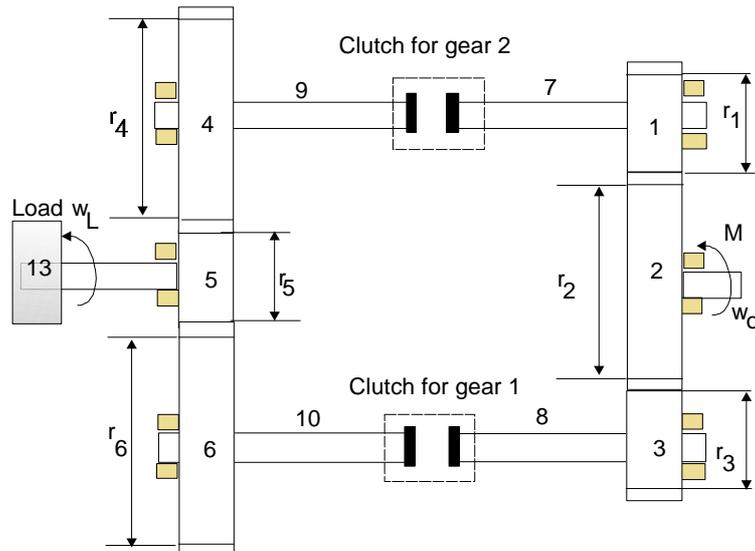


Figure 5.4. Simplified diagram for the gearbox

The gearbox could be seen as two dual transformer systems T_1 and T_2 . The first dual transformer system T_1 consists of three cogwheels with the characteristic parameters r_1, r_2 and r_3 . The three cogwheels have mass moment of inertia I_1, I_2 and I_3 respectively.

The second transformer T_2 also consists of three cogwheels with the characteristic parameters r_4, r_5 and r_6 , and with mass moment of inertia I_4, I_5 and I_6 respectively.

First clutch, labeled (S_{w1}) is connected between T_1 and T_2 via axles (8,10) with mass moment of inertia I_8 and I_{10} . Second clutch, labeled (S_{w2}) is connected between the two transformers via axles (7,9) with mass moment

of inertia I_7 and I_9 . The corresponding property graph of the gearbox is shown in Figure 5.5.

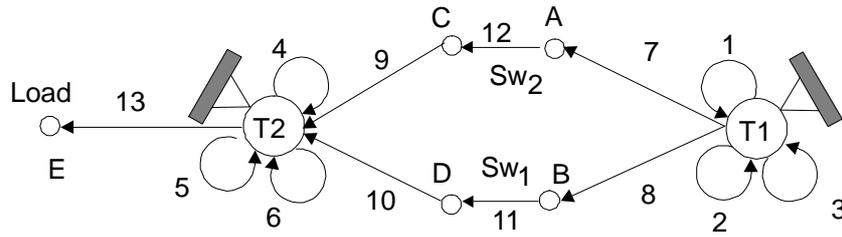


Figure 5.5. Property graph of the gear box

In the property graph, node A and C represent the plates of the second clutch. Node B and D represent the plates of the first clutch. Since the gearbox is constructed to prevent freewheeling, a new gear has to be slightly engaged, before the previous gear is disengaged. This is done by a proper control law for the normal forces applied on the clutch plates as shown in Figure 5.6.

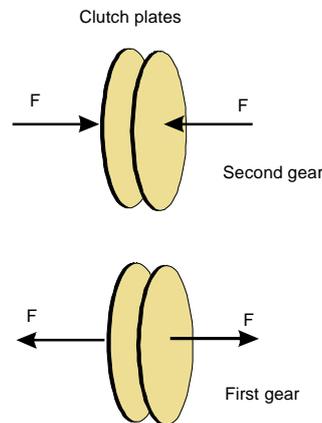


Figure 5.6. Normal forces applied on the clutch's plates

These normal forces are the control variables used to execute gear changes. With zero normal force on the first clutch, the clutch will be disengaged. The degree of engagement will increase with increased normal force until the plates are fixed on each other. In the given reference [22], there were no data given for the control law of these normal forces applied on the plates of the clutches due to secrecy. Therefore, we shall substitute these controlled normal forces applied on the clutch's plates by changing gradually the viscous friction between clutch plates as the clutch plates are sliding towards each other.

In the light of the above, we can conclude that the gearbox in the current example has three modes of operation as shown in Figure 5.7.

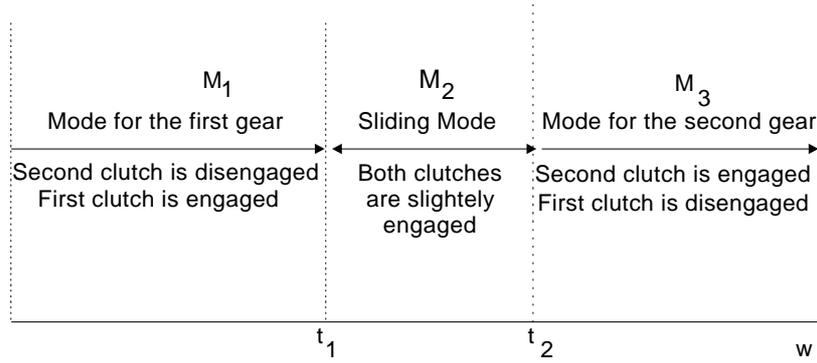


Figure 5.7 Modes of operations for the gearbox

The gearbox runs in its first mode when the first clutch is totally engaged and the second clutch is totally disengaged. The second mode of operation is the sliding mode, this mode is activated during transition from the first gear to the second gear when both clutches are slightly engaged to avoid freewheeling. Finally, the gearbox runs in its third mode when the second clutch is totally engaged and the first clutch is totally disengaged.

5.1.1. Logical conditions for switching

Provided that the gearbox is initially in the first mode M_1 , the gearbox will remain in this mode as long as the angular speed of the load is less than the reference speed for switching:

$$v_L < w$$

In this mode of operation, since the second clutch is completely disengaged, then the property of the second clutch (y_{12}) will be given by:

$$y_{12} = R_0 S_{w2}$$

Where $R_0 \cong 0$

The property of the first clutch will be given by:

$$y_{11} = R_c \tilde{S}_{w1}$$

Where $R_c \cong \infty$

The gearbox enters its second mode of operation when the condition for switching is detected that is, when $v_L = v$, in this mode, the viscous friction in the first clutch will decrease gradually to make it slide until it is

completely disengaged. Concurrently, to increase the friction gradually in the second clutch, until the second clutch is completely engaged as shown in Figure 5.8.

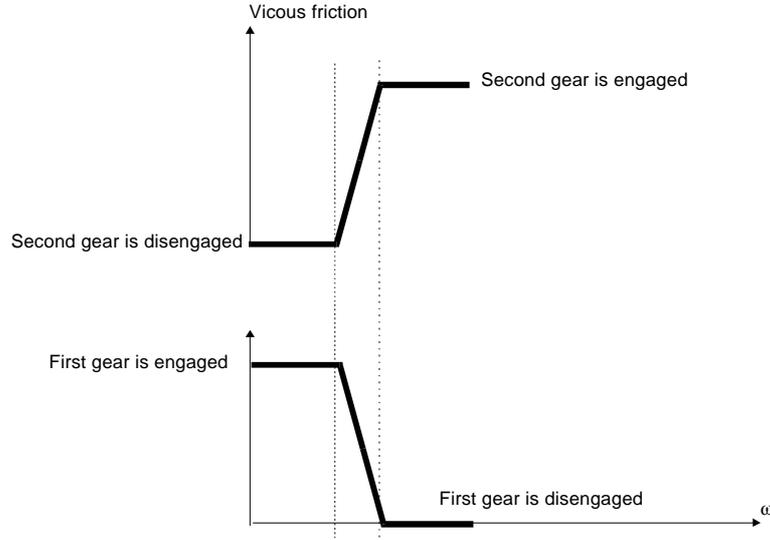


Figure 5.8. Gearbox in its second mode (sliding mode)

Provided that the gearbox enter this mode at $(t = t_1)$ and exits at $(t = t_2)$, then the property of the first clutch at this mode of operation could be given by the linear function:

$$y_{12} = R_0 \times (t_2 - t_1) S_{w2}$$

The property of the first clutch at this mode of operation is given by:

$$y_{11} = R_c \times 1 - (t_2 - t_1) \tilde{S}_{w1}$$

The gearbox enters its third mode of operation when the second clutch becomes totally engaged and the first gear become totally disengaged. The corresponding property of the clutches at this mode will be given by:

$$y_{12} = R_c \tilde{S}_{w2}$$

$$y_{11} = R_0 S_{w1}$$

Now we can combine the property of the second clutch at all modes of operation in one array hybrid equation:

$$y_{12} = R_0 \times S_{w2} + (R_c \times (t_2 - t_1) \times S_{w2}) + (R_c \times \tilde{S}_{w2})$$

$$V = \begin{matrix} & \text{T1} & \text{T2} & \text{A} & \text{B} & \text{C} & \text{D} & \text{E} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \\ 12 \\ 13 \end{matrix} & \begin{bmatrix} -1/r_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/r_2 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1/r_3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1/r_4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/r_5 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1/r_6 & 0 & 0 & 0 & 0 & 0 \\ -1/r_1 & 0 & -1 & 0 & 0 & 0 & 0 \\ -1/r_3 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1/r_4 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1/r_6 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 & 0 \\ 0 & 1/r_5 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \end{matrix}$$

5.1.3. Simulation

For simulation purpose, we assumed that time span from t_1 to t_2 is one second. This is the time from the instant the system detects that the condition for switching is fulfilled until it accomplishes switching from the second mode to the third mode. We shall start simulation under the assumption that the central cogwheel in (T1) is subjected to a torque given by this step function: $I_{S2} = 1$

Figure 5.9 shows the angular velocity of the load (element 13) and transformer system T1 and T2, during all modes of operations that are clearly visited as shown by simulation.

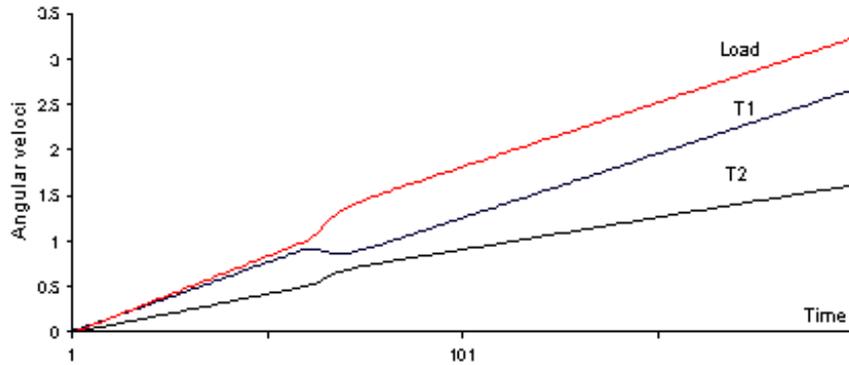


Figure 5.9. Output angular velocity of the load and transformer systems T1 and T2

Figure 5.10 shows the angular velocity of the second clutch plates (node A and C). The figure shows that after switching the angular velocity of the plates is approximately identical. The offset between the plates speed is due to the use of non-ideal switching element.

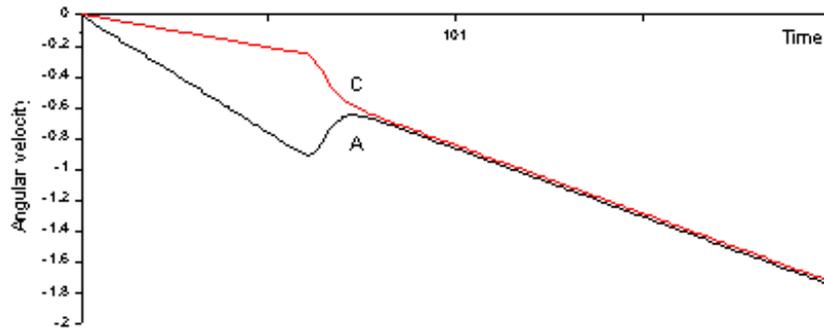


Figure 5.10. Angular velocity of the second clutch plates (node A and node C)

Figure 5.11 shows the angular velocity of the plates of the first clutch, the figure also shows that the plates-speed was approximately equal prior switching during the first mode.

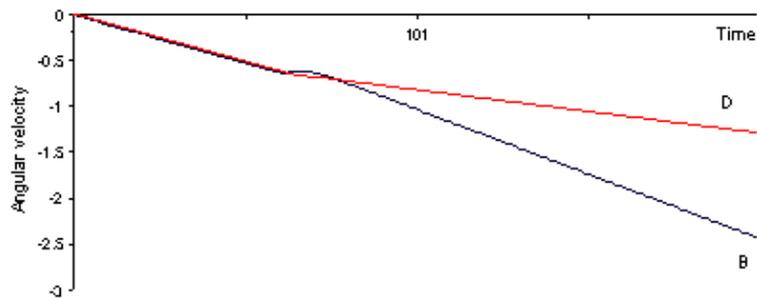


Figure 5.11. Angular velocity the first clutch plates (node B and node D)

5.2. Summary

Control directives issued by the logical system could alter the underlying equations of the physical system. The overall system behavior in this case is hybrid in nature, exhibiting both continuous and significant discrete characteristics. In order to handle the computational problems arising from the presence of discrete and continuous variables, one possible approach is to fix the topology of the physical system to remain unaffected by the discrete variables. This was accomplished by representing these discrete variables by non-ideal switching elements with artificial resistance. From a mathematical point of view, by such formulation we accomplished the following advantages:

1. We obtained an explicit mathematical model that contains all modes of operations. All variables were kept observable at all modes and the system's state space dimension remained invariant to switching.
2. Combined simulation of both discrete and continuous system has become possible.

The price of using non-ideal switching element is evidently the accuracy of the results. For instance, in the last example, we could notice a slight offset between the angular velocity of both plates, which is not theoretically accurate since after the engaging the clutch, both plates must have identical angular velocity.

6 IDEAL SWITCHING ELEMENTS

In the ideal case, a switch element distinguishes itself from all other types of physical elements in the way its variables are related. In an ideal resistor, resistor's state variables are related via the property *Resistance* which identifies how many units of potential variables we get from one unit of flow variables flowing through the resistor:

$$e = Ri$$

In an ideal switch, there is no static relation that tells us how the transformation from flow variables (i_{sw}) to potential variables (e_{sw}) can be performed.

Let us investigate what types of relation exist between flow and potential variables in an ideal switch. First, when the switch is closed, a short circuit will be established between the terminal of the switch and consequently the potential drop across the switch will become equal to zero. At the same time, the flow through it becomes different than zero as shown in Figure 6.1.

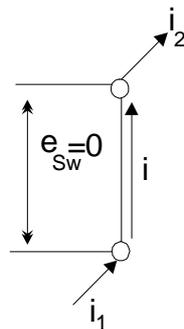


Figure 6.1. Switch is closed, flow variable is different than zero and potential drop is equal to zero

When the switch is open, open circuit is established and this cause the flow through the switch to become equal to zero and the potential difference between the boundaries of the switch becomes different than zero as shown in Figure 6.2.

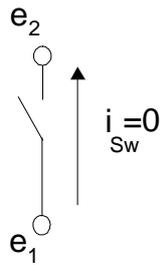


Figure 6.2. Switch is open, flow variable is equal to zero and potential variable is different than zero

The amount of flow through the switch is determined only by the amount of flow through the elements connected to the switch, which in turn depends on the state of the switch. Therefore, the important aspect in this regard is the notion of the presence or absence of switch variables and not their magnitude.

The notion of *presence* or *absence* implies that switch variables can only take values from the binary domain in contrast to the variables in the basic primitive elements that can take values from the real number domain. If the switch is closed, then in binary terms this corresponds to setting $(i_{Sw} = 1), (e_{Sw} = 0)$. On the other hand, opening the switch corresponds to setting $(e_{Sw} = 1)$ and $(i_{Sw} = 0)$ as shown in Figure 6.3.

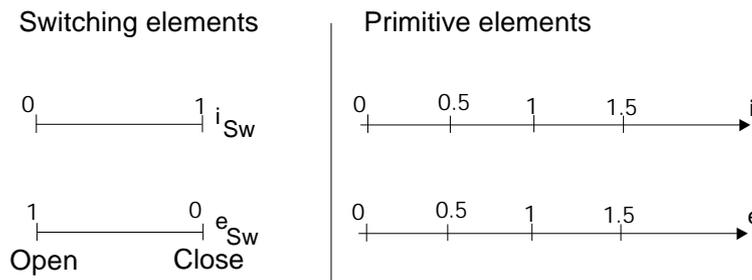


Figure 6.3. Domain of variables in switching and primitive elements

The above discussion shows that:

- The concept of property of an ideal switch element can only be defined in binary terms. It either exists or does not exist. Thus, it can not be measured and therefore considering the ideal switch, as a part of the primitive system must be ruled out
- The only variable that could be modeled explicitly in systems differential equations is the absence or presence of flow or potential variables in the switch.

In summary, an ideal switch element could not be treated, as a primitive element since no algebraic relation can be established between its state variables. The question which we shall elaborate further is what is a switch element and where does it belongs on our system model.

We shall use *node connection procedure* as a reference for model formulation. Node connection procedure is located at the right hand side of Roth diagram. It is used to set up the model of the total system based on the model of the primitive system (Y), the connection matrix (V) and a set of node sources (I_N) and element sources (I_s). The algebraic structure of this procedure is shown in Figure 6.4.

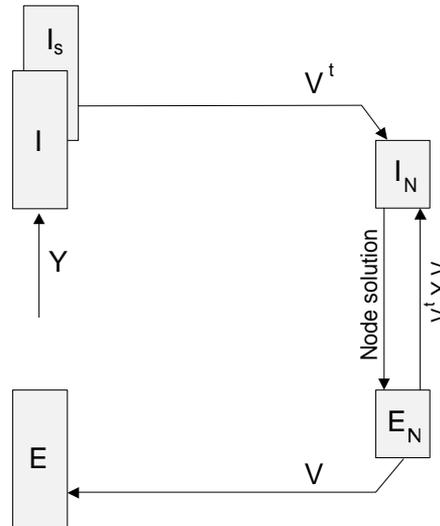


Figure 6.4. Node connection procedure

6.1. The Switch As a Connection Element

As a connection element, the switch adds one more connectivity constraint to the elements connected to its boundary nodes. In its closed state, it forces the potential drop at its boundary nodes to be identical ($e_1 = e_2$) as shown in Figure 6.5.

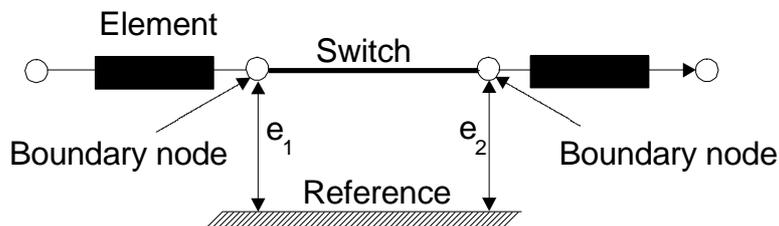


Figure 6.5. Switch as a connection element- closed state

How to take this extra connectivity constraint into consideration in order to generate a compact and explicit model. At the same time to keep the dimension of the state space model invariant to these additional constraints is what we shall discuss in the following sections.

6.2. Types of Switch Connection

With reference to the general system model described in section 2.1, we could single out two unique types of switch connection in the topology graph:

1. The switch is used to connect or disconnect an element as shown in Figure 6.7. In this case, altering the state of the switch will not alter the state space dimension of the continuous system. It will cause the physical system to switch between two different models that share the same dimension.

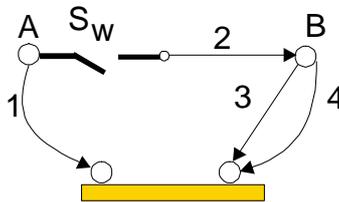


Figure 6.6. Switch is used to connect or disconnect elements

2. The switch is connected between two free nodes as shown in Figure 6.7. In this case, altering the state of the switch will cause state variables to jump between two different modes where each mode has different state space dimension.

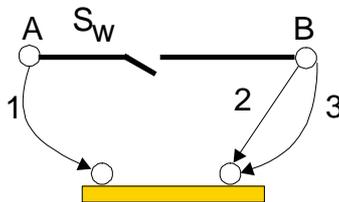


Figure 6.7. Switch is used to connect or disconnect nodes

We will show that toggling the state of the switch from one state to another cause connection objects to change by a certain pattern. This implies that once we identify connection objects at one mode, connection objects in the other modes can be found by simple array operation. These implications simply reduce the amount of work needed to identify and capture all possible combinations of mode models. In addition, it offers us the freedom of automating model formulation.

We will now present in detail the topological and algebraic consequences as well as a modeling procedure for each case.

6.2.1. Switch is connected in series with an element

In the first case, the switch element is connected in series with an element between two nodes as shown in Figure 6.8.

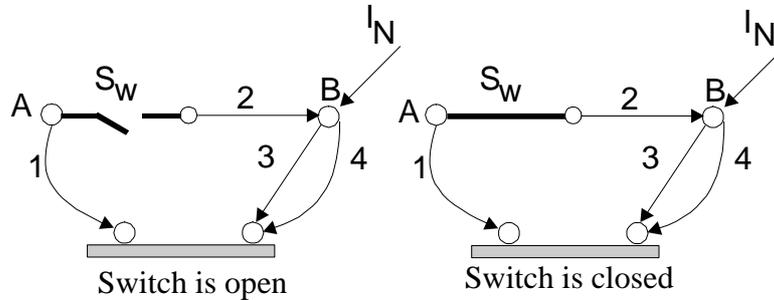


Figure 6.8. Switch is connected in series with an element

Topologically, this type of connection will not lead to any reduction or increase in the number of nodes in the property graph and therefore the state space dimension will remain invariant to switching.

In physical terms, closing the switch corresponds to allow a flow variable to pass through the element. Hence, element transformation will take place if and only if the switch is closed. Mathematically this can be expressed by the following hybrid equation:

$$f = y.i.\tilde{S}_w$$

Where:

(f) is the force across the element.

(i) is the flow passing through the element.

(\tilde{S}_w) is switch state.

(y) is element property.

The state of the switch can be combined with the property of the element, since the flow through the element is only allowed if the switch is closed:

$$f = (y.\tilde{S}_w)i$$

$$f = y'.i$$

When the switch is closed, \tilde{S}_w will be equal to one and the property of the element (y') will be equal to (y). When the switch is open, \tilde{S}_w will become equal to zero and therefore the property of the element will become equal to zero.

Accordingly, the primitive system model could be now given by the hybrid strand object:

$$Y_D = \begin{bmatrix} y_1 & & & \\ & y_2 \tilde{S}_w & & \\ & & y_3 & \\ & & & y_4 \end{bmatrix}$$

The connection matrix of the system will remain invariant to the state of the switch element because such connection does not affect the topological structure of the property graph.

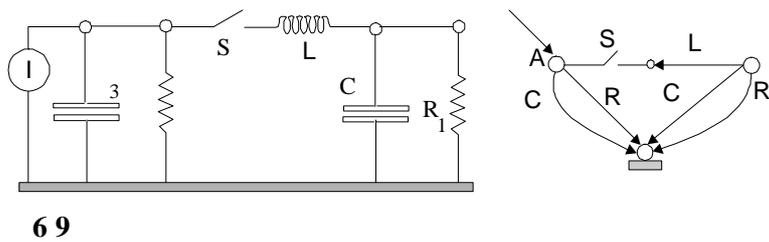
$$V = \begin{matrix} & \begin{matrix} A & B \end{matrix} \\ \begin{matrix} 1 \\ 1 \\ 0 \\ 0 \end{matrix} & \begin{bmatrix} 1 & 0 \\ 0 & -1 \\ 1 & 1 \\ 0 & 1 \end{bmatrix} \end{matrix}$$

Connected system model is then obtained as usual by applying the transformation:

$$Y_N = (V^t Y V)$$

Example 6-1. Switching circuit

Consider the electrical circuit shown in Figure 6.9. The corresponding property graph of the circuit is shown to the left of the figure.



First, we shall start as usual by identifying the primitive system model of L_5 then the

element when the switch is closed. The model of the primitive system will be given by:

$$Y_D = \begin{bmatrix} (1/R_1) & & & & \\ & (1/R_2) & & & \\ & & C_3 s & & \\ & & & C_4 s & \\ & & & & \tilde{S}_w \frac{(1/L_5)}{s} \end{bmatrix}$$

From the property graph, the corresponding connection matrix will be given by:

$$V = \begin{matrix} & \text{A} & \text{B} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} & \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 1 & -1 \end{bmatrix} \end{matrix}$$

The corresponding set of equations that describe the system are obtained by applying the transformation $(V^t Y V)$ yielding to the set of second order differential equations:

$$\begin{bmatrix} I_{NA} \\ 0 \end{bmatrix} = \begin{bmatrix} (1/R_2) + C_3 s + \frac{(1/L_5)}{s} \tilde{S}_w & -\frac{(1/L_5)}{s} \tilde{S}_w \\ -\frac{(1/L_5)}{s} \tilde{S}_w & (1/R_1) + C_4 s + \frac{(1/L_5)}{s} \tilde{S}_w \end{bmatrix} \begin{bmatrix} e_{NA} \\ e_{NB} \end{bmatrix}$$

When the switch is connected, we get $\tilde{S}_w = 1$, and the differential equations of the system yields:

$$\begin{bmatrix} I_{NA} \\ 0 \end{bmatrix} = \begin{bmatrix} (1/R_2) + C_3 s + \frac{(1/L_5)}{s} & -\frac{(1/L_5)}{s} \\ -\frac{(1/L_5)}{s} & (1/R_1) + C_4 s + \frac{(1/L_5)}{s} \end{bmatrix} \begin{bmatrix} e_{NA} \\ e_{NB} \end{bmatrix}$$

When the switch is open we get ($\tilde{S}_w = 0$) we obtain:

$$\begin{bmatrix} I_{NA} \\ 0 \end{bmatrix} = \begin{bmatrix} (1/R_2) + C_3 s & 0 \\ 0 & (1/R_1) + C_4 s \end{bmatrix} \begin{bmatrix} e_{NA} \\ e_{NB} \end{bmatrix}$$

As expected, this is a trivial case, since the transition from one mode to another will not cause any rank reduction of state space. It will affect only

the numerical values of the connected system model. The system in the example was also simulated using the same set of logical conditions for switching in Example 5-1. Figure 6.10 shows the result for simulation for the voltage drop of both nodes (e_{NA} , e_{NB}).

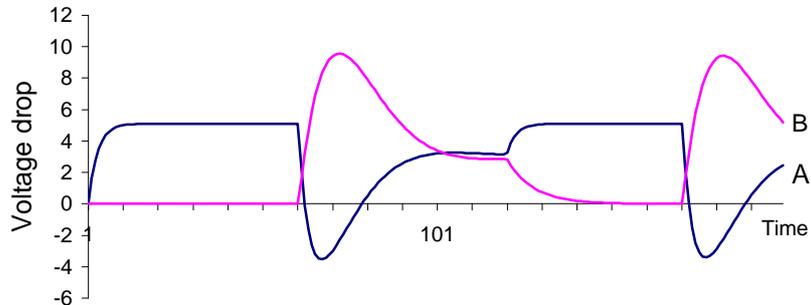


Figure 6.10. Voltage drop of node A and B

Let us now turn the attention to the problem of variable structure control presented in Example 4-1, which is presented for convenience in Figure 6.11. Note that, in this problem there are no real physical switches to perform switching, therefore they have to be made and added to the connection model.



Figure 6.11. A dynamic system consisting of mass element (m) moving on a frictionless surface under the applied force (f)

In the example, it was given that the dynamical system is supposed to switch between two control laws at ($t = 10$), from:

$$(y_{c1} = r_1 s + k) \quad \text{to} \quad (y_{c2} = r_2 s + k)$$

By examining these continuous control laws, it was easy to observe that switching implies to connect (r_2) and concurrently disconnect (r_1) at ($t = 10$). The spring constant in both control laws was not affected by switching. Now we can reconstruct the topology of the dynamical system to include one ideal switch connected to the rest of the system as shown in Figure 6.12.

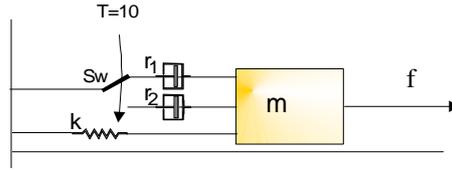


Figure 6.12. Dynamic system with ideal switching element

The logical condition for switching is then given by the following condition:

$$S_w = \begin{cases} 1 & \text{for } (t < 10) \\ 0 & \text{for } (t \geq 10) \end{cases}$$

With respect to the discussion outlined above, switching does not cause any jumps or reduction to the order of dynamical system. Therefore, the state of the switch can be combined with the primitive system model:

$$Y_D = \begin{bmatrix} ms^2 & & & \\ & (r_1 s) S_w & & \\ & & (r_2 s) \tilde{S}_w & \\ & & & k \end{bmatrix}$$

Connection object V is obtained from the topology of the dynamical system:

$$V = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

The connected system model is then obtained by applying the usual transformation, yielding the following hybrid equation:

$$f = ms^2 + (r_1 S_w + r_2 \tilde{S}_w) s + k$$

When the system runs in its first mode, we get ($S_w = 1$) and the differential equation of this mode becomes:

$$f = ms^2 + r_1 s + k$$

When the system runs in its second mode, we get ($S_w = 0$) and the differential equation of this mode becomes:

$$f = ms^2 + r_2 s + k$$

The hybrid model ensures smooth transition from one mode to another without having to worry about setting the state of the system at each transition as shown in Figure 6.13.

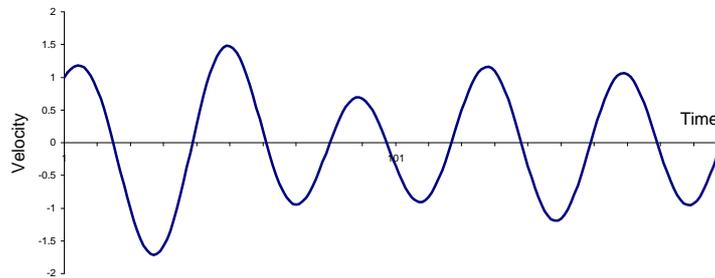


Figure 6.13. Pump-less switching obtained by a hybrid formulation of continuous system model

6.2.2. Switch is connected between two free nodes

In this case, the switch is connected between two free nodes, closing the switch will lead to a stiff connection between the boundary nodes of the switch as shown Figure 6.14.

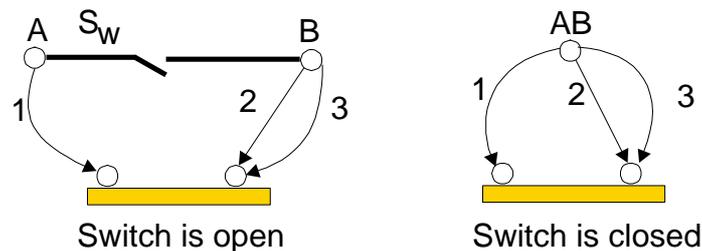


Figure 6.14. Stiff connection as a result of closing the switch

The consequence of such topological fusing is the reduction of the number of differential equations required for describing system behavior. The dimension has been reduced from two equations to only one equation since each node in the property graph represents one degree of freedom the system can have. Therefore, fusing two nodes leads to a state space reduction and some variables will no longer be observable in this mode.

In order to avoid state space reduction as a result of switching, a switch element in the closed state will be represented in the property graph as a bold line connected between the two free nodes without any orientation as shown in Figure 6.15.

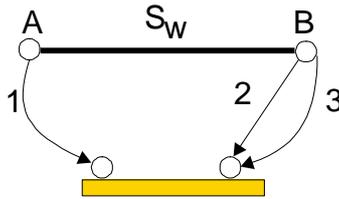


Figure 6.15. Switch representation in the property graph

Using this bold line as a traffic way, the elements connected to node B could be moved along the line and linked to node A. Similarly, the elements connected to node A could be moved along the line and linked to node B, as shown Figure 6.16.

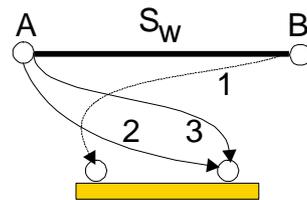


Figure 6.16. Swapping of elements between node A and node B when the switch is closed

Therefore, a switch element in the property graph can be seen as a highway that permits the movement of variables from one node to another. This imaginary highway will help us to generate automatically connection objects which are valid for each switch state.

In Figure 6.17, we have drawn the property graph prior and after closing the switch. Provided that the switch initially is open. The connection matrix at this mode can be set up.

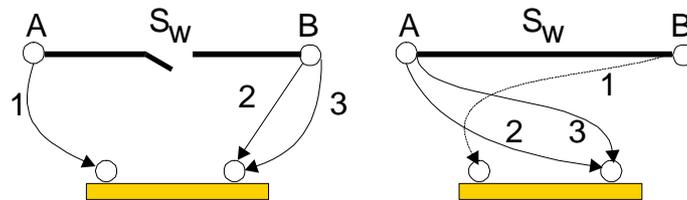


Figure 6.17. Property graphs for both modes

To the left of Figure 6.17, the compatibility condition at node A is given by:

$$e_1 = e_{NA}$$

Where (e_1) is the local displacement of element one, and (e_{NA}) is the displacement of node A. Similarly, compatibility condition at node B is given by two equations:

$$e_2 = e_{NB}$$

$$e_3 = e_{NB}$$

Where (e_2) and (e_3) are the local displacements of element two and three, and (e_{NB}) is the displacement of node B. The corresponding connection matrix representing these three constraints labeled V_1 will be given by:

$$V_1 = \begin{matrix} & \text{A} & \text{B} \\ \begin{matrix} 1 \\ 2 \\ 3 \end{matrix} & \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix} \end{matrix}$$

Physically, closing the switch corresponds to *forcing* the displacement of both nodes forming the boundary of the switch to be identical, that is:

$$e_{NA} = e_{NB}$$

This simply means that the compatibility conditions at each node can now be transmitted to the other node through the free-way established by closing the switch, that is from node A to node B and vice versa. By this transmission, we did not violate any physical rule, we just looked at the compatibility condition from two different spots. Hence, with reference to the right side of Figure 6.17, the compatibility conditions at the two nodes can now be re-written as:

$$e_1 = e_{NB}$$

$$e_2 = e_{NA}$$

$$e_3 = e_{NA}$$

Thus, the corresponding connection matrix of the system which reflects this connection is given by V_2 :

$$V_2 = \begin{matrix} & \text{A} & \text{B} \\ \begin{matrix} 1 \\ 2 \\ 3 \end{matrix} & \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 1 & 0 \end{bmatrix} \end{matrix}$$

If we compare V_1 with V_2 we could observe that they could be deduced from each other by simply swapping the columns of the nodes that are connected to the switch:

$$V_2[:, A] = V_1[:, B]$$

$$V_2[:, B] = V_1[:, A]$$

Providing that the switch normally is open, then the model of the connected system could be obtained by applying the transformation:

$$Y_N = (V_1^T Y V_1) + (V_2^T Y V_2) \tilde{S}_w$$

When the switch is closed, \tilde{S}_w will be equal to one. This gives us the following set of equations:

$$Y_N = (V_1^T Y V_1) + (V_2^T Y V_2)$$

When the switch is open, \tilde{S}_w will be equal to zero and we obtain:

$$Y_N = (V_1^T Y V_1)$$

By this formulation, we kept the dimension of the system, invariant to switching and at the same time included the extra constraint caused by the presence of the switch. All modes of operations the system undergoes are modeled explicitly in systems differential equations. The following examples will be used to illustrate model formulation under different circumstances.

Example 6-2. Mechanical system with ideal clutch

Consider the same mechanical system that was used in Example 5-1. It contains two axial shafts that are connected via an **ideal clutch**. The corresponding property graph of the system is shown to the left of the mechanical system in Figure 6.18.

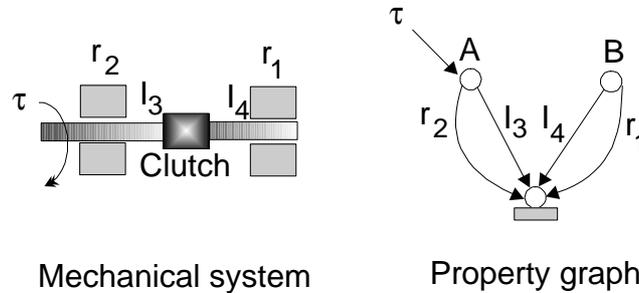


Figure 6.18. Mechanical system with an ideal switching element

First, we shall start as usual by identifying the primitive system model of the mechanical system given by the property matrix Y :

$$Y = \begin{bmatrix} r_1 s & & & \\ & r_2 s & & \\ & & I_3 s^2 & \\ & & & I_4 s^2 \end{bmatrix}$$

In Figure 6.19, We have re-drawn property graph for the two modes of operations the system can take.

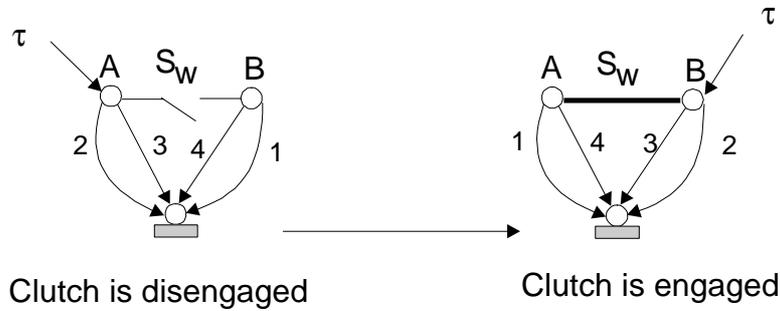


Figure 6.19. Property graphs of the mechanical system in both modes of operations

When the clutch is disengaged, the corresponding connection matrix will be given by:

$$V_1 = \begin{matrix} & \begin{matrix} A & B \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \end{matrix}$$

The connection matrix for the system when the clutch is engaged labeled V_2 is obtained by swapping the location of the entire column A and column B in the connection matrix:

$$\begin{aligned} V_2[:, A] &= V_1[:, B] \\ V_2[:, B] &= V_1[:, A] \end{aligned}$$

$$V_2 = \begin{matrix} & \begin{matrix} A & B \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \end{matrix}$$

The differential equations of the system can now be generated by the transformation:

$$Y_N = (V_1^T Y V_1) + (V_2^T Y V_2) \tilde{S}_w$$

This transformation gives the following set of hybrid equations:

$$\begin{bmatrix} \dot{\mathbf{t}}_A \\ \tilde{S}_w \dot{\mathbf{t}}_A \end{bmatrix} = \begin{bmatrix} (r_2 s + I_3 s^2) + (I_4 s^2 + r_1 s) \tilde{S}_w & 0 \\ 0 & (r_1 s + I_4 s^2) + (r_2 s + I_3 s^2) \tilde{S}_w \end{bmatrix} \begin{bmatrix} \mathbf{q}_A \\ \mathbf{q}_B \end{bmatrix}$$

If the clutch is disengaged, the binary variable \tilde{S}_w will be equal to zero and we obtain the following set of equations:

$$\begin{bmatrix} \dot{\mathbf{t}}_A \\ 0 \end{bmatrix} = \begin{bmatrix} (r_2 s + I_3 s^2) & 0 \\ 0 & (r_1 s + I_4 s^2) \end{bmatrix} \begin{bmatrix} \mathbf{q}_A \\ \mathbf{q}_B \end{bmatrix}$$

In the second mode when the clutch engaged, \tilde{S}_w becomes equal to one, and we get the following set of equations:

$$\begin{bmatrix} \dot{\mathbf{t}}_A \\ \dot{\mathbf{t}}_A \end{bmatrix} = \begin{bmatrix} (r_2 s + I_3 s^2) + (I_4 s^2 + r_1 s) & 0 \\ 0 & (r_1 s + I_4 s^2) + (r_2 s + I_3 s^2) \end{bmatrix} \begin{bmatrix} \mathbf{q}_A \\ \mathbf{q}_B \end{bmatrix}$$

The logical conditions for switching was assumed to be time-related events in which the clutch will change its state according to the following conditions:

$$S_w = \begin{cases} \text{disengaged} & (0 \leq t \leq 0.6) \text{ Or } (1.2 \leq t \leq 1.7) \\ \text{engaged} & \text{otherwise} \end{cases}$$

Figure 6.20, shows the angular velocity of the two axial shafts (node A and node B) in response to a step input torque around the prime mover shaft, that is node A in the property graph.

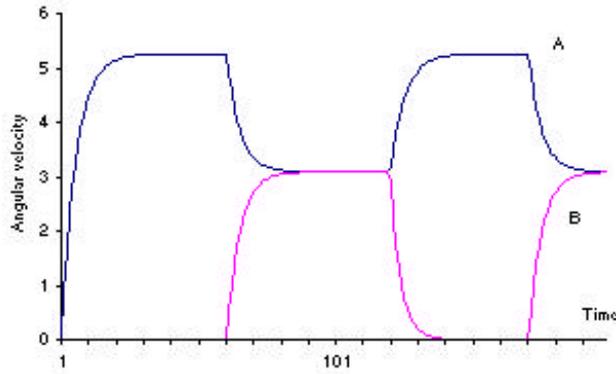


Figure 6.20. Angular velocity of node A and node B (ideal switching case)

The next figure shows the angular rotation and angular velocity of node B in response to the same input torque. The figure shows clearly that all variables were observable from all modes of operation and the conditions for transition from one mode to another were implicitly contained in the mathematical model.

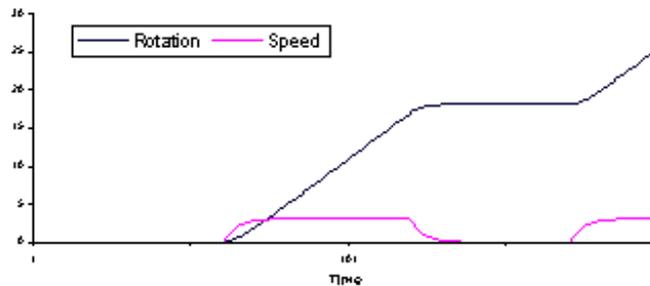


Figure 6.21. Angular velocity and angular rotation of Node B (ideal switching case)

Example 6-3. Property graph with ideal switch

Consider the property graph shown in Figure 6.22. An ideal switch is connected between node A and element 2. In this example, the tail node of element 2 is subjected to input source I_N .

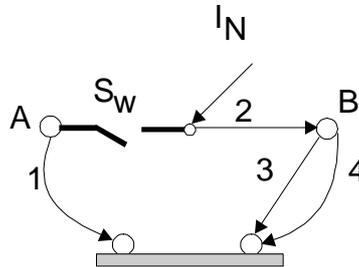


Figure 6.22. Property graph with an ideal switching element

Structurally, the property graph of the system is identical to the property graph in the controlled switching case presented in section 6.2.1. However, since the tail node of element 2 is subjected to input source, it becomes impossible to apply the procedure of section 6.2.1. The procedure used in section 6.2.1 is valid if and only if there are no sources applied at the point that connects the switch to the element.

In this case, the property graph has to be re-drawn to reflect the fact that the tail node of element 2 must be seen as an independent node that carries a global variable as shown in Figure 6.23.

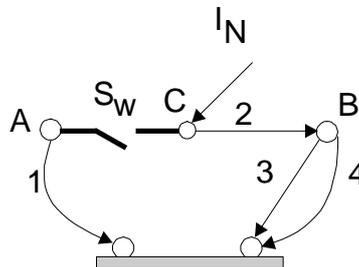


Figure 6.23. Reconstructed graph

The graph shown in Figure 6.23 structurally obeys the procedure outlined in section 6.2.2. From the property graph shown to the left in Figure 6.24, when the switch is open, the corresponding connection matrix will be given by:

$$V_1 = \begin{matrix} & \begin{matrix} A & B & C \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix} \end{matrix}$$

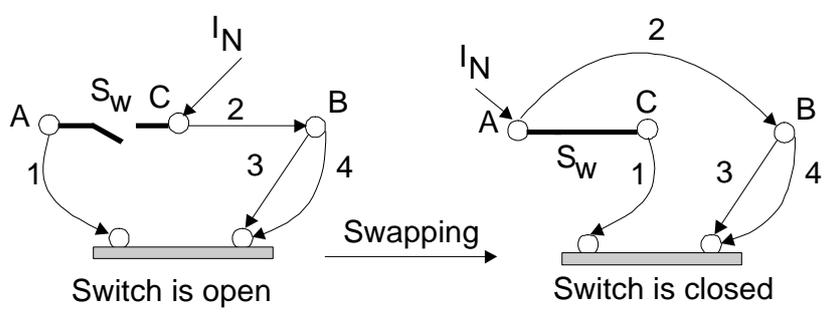


Figure 6.24. Property graph before and after switching

Connection matrix for the system when the switch is closed (shown to the right of Figure 6.24) is obtained by swapping only column A and column C. column B in this case remains unaffected by the switching.

$$V_2 = \begin{matrix} & \begin{matrix} A & B & C \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{bmatrix} 0 & 0 & 1 \\ 1 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix} \end{matrix}$$

The differential equations of the system can now be generated by the transformation:

$$Y_N = (V_1' Y V_1) + (V_2' Y V_2) \tilde{S}_w$$

6.3. Application Example: Gearbox – Ideal Case

Now it is tempting to try to model the gearbox presented in the previous chapter but with ideal clutches this time. The gearbox is again shown in Figure 6.25.

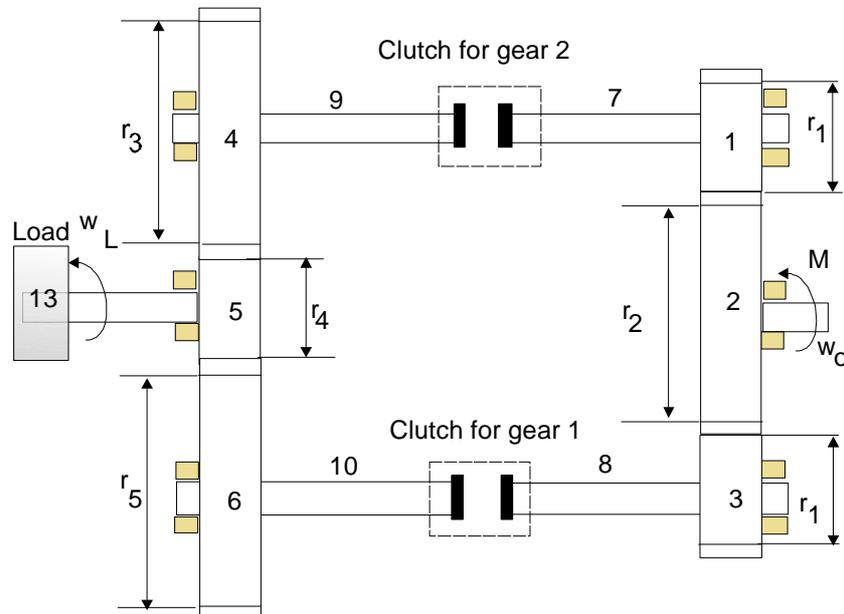


Figure 6.25. Simplified diagram for the gearbox

We have pointed out that the gearbox is configured to switch between three different modes of operations shown in Figure 6.26.

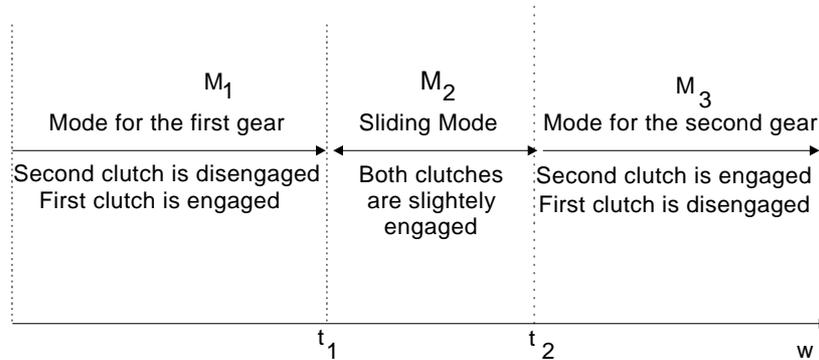


Figure 6.26. Modes of operations for the gearbox

In order to consider the three modes of operation, each clutch shall be represented by two ideal switches and one damper. The damper represents

viscous friction between clutch plates during sliding mode. The label Sw_{11} , Sw_{12} , and y_{11} will be used to represent the first clutch. The labels Sw_{21} , Sw_{22} , and y_{12} will be used to represent the second clutch as shown in the property graph of the gearbox in the ideal case in Figure 6.27.

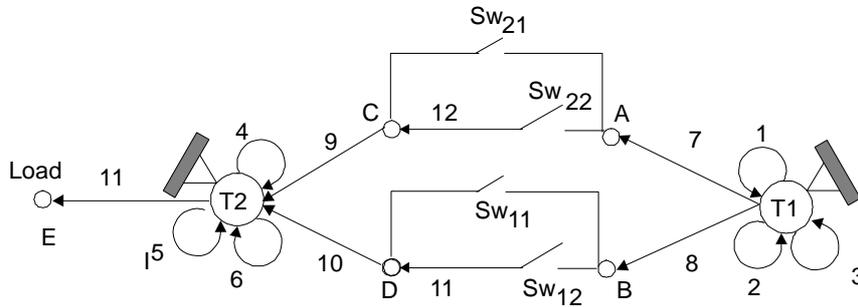


Figure 6.27. Property graph of the gearbox- ideal case

It could be observed that the model of the gearbox with ideal clutches could be seen as a combination of the first case discussed in section 6.2.1 and the second case discussed in section 6.2.2.

6.3.1. System modeling

We shall start the analysis from the second mode since it contains the maximum number of elements and the model has the highest state space dimension. Figure 6.28 shows the state of the switches in the second mode of operation. In this mode, both Sw_{11} and Sw_{21} are open while Sw_{12} and Sw_{22} are closed. This will cause the damper y_{11} to be connected between node B and node D, and the damper y_{12} to be connected between node A and C. As we have indicated, both dampers represent the viscous friction in the clutches during sliding mode.

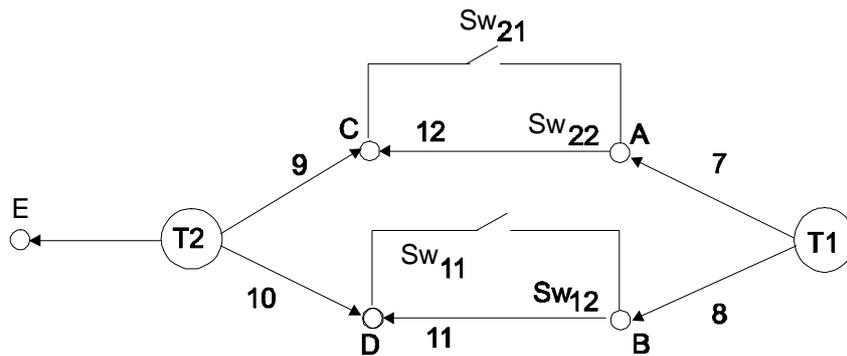


Figure 6.28. State of the switches during sliding mode

$$V_2 = \begin{matrix} & \text{T1} & \text{T2} & \text{A} & \text{B} & \text{C} & \text{D} & \text{E} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \\ 12 \\ 13 \end{matrix} & \begin{bmatrix} -1/r_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/r_2 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1/r_3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1/r_4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/r_5 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1/r_6 & 0 & 0 & 0 & 0 & 0 \\ -1/r_1 & 0 & -1 & 0 & 0 & 0 & 0 \\ -1/r_3 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 1/r_4 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1/r_6 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 & 0 \\ 0 & 1/r_5 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \end{matrix}$$

Finally, Figure 6.30, shows the property graph of the gearbox in the third mode of operation. In this mode, all the switches are in open state except Sw_{21} , this will cause the angular velocity of both node A and node C to be identical as long as Sw_{21} is closed.

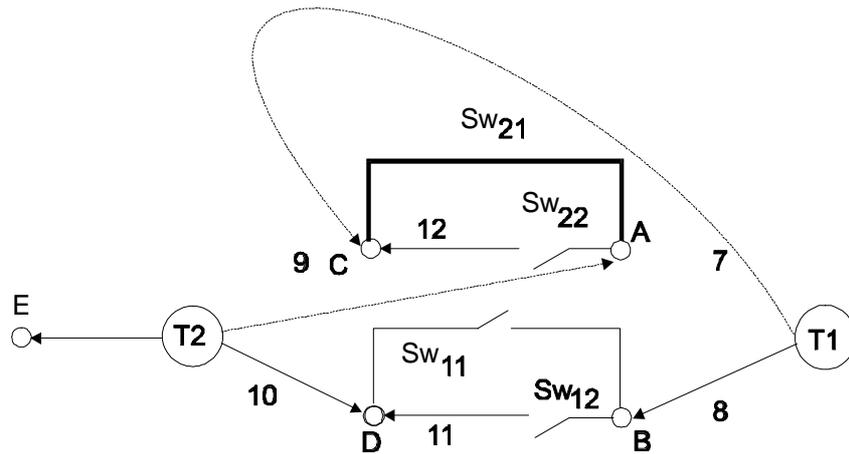


Figure 6.30. State of the switches at third mode of operation

The corresponding connection matrix for the gearbox in this mode labeled V_3 , is obtained by swapping columns A and column C in the connection matrix V_1 , yielding:

$$V_3 = \begin{matrix} & \text{T1} & \text{T2} & \text{A} & \text{B} & \text{C} & \text{D} & \text{E} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \\ 12 \\ 13 \end{matrix} & \begin{bmatrix} -1/r_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/r_2 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1/r_3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1/r_4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/r_5 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1/r_6 & 0 & 0 & 0 & 0 & 0 \\ -1/r_1 & 0 & 0 & 0 & -1 & 0 & 0 \\ -1/r_3 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1/r_4 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1/r_6 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & 1/r_5 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \end{matrix}$$

Now the model of the connected system could be obtained by applying the transformation:

$$Y_N = (V_1^T Y V_1) + (V_2^T Y V_2) \tilde{S} w_{11} + (V_3^T Y V_3) \tilde{S} w_{21}$$

6.3.2. Simulation

Figure 6.31 shows that in the first mode the angular velocity of first clutch plates are identical, since the clutch in this case is totally engaged.

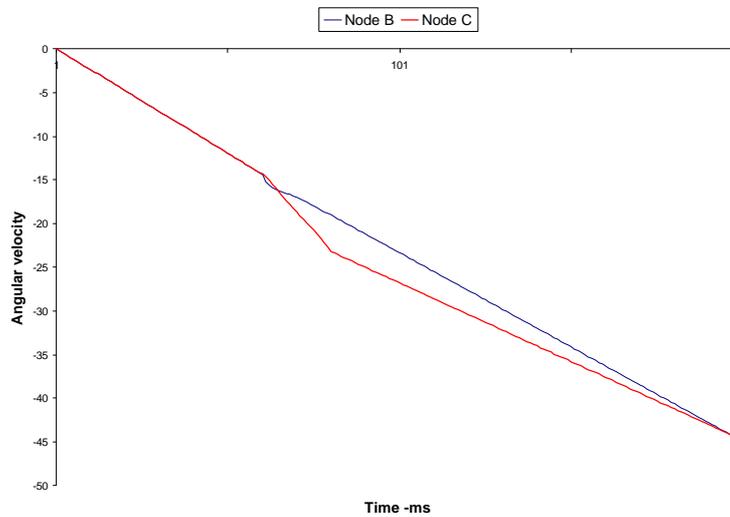


Figure 6.31. Angular velocity of the first clutch plates (node B and D) during all modes of operations

When the gearbox enters its second mode, the plates starts to slide away from each other but still slightly engaged. In the third mode, clutch plates are totally disengaged and the gearbox enters its third mode.

Figure 6.32, shows the angular velocity of the second clutch plates, in the first mode both plates are rotating with different speeds since they are totally disengaged. As the gearbox enters its third mode, the angular velocity of the plates start to approach each other until they become identical when the gearbox enters its third mode.

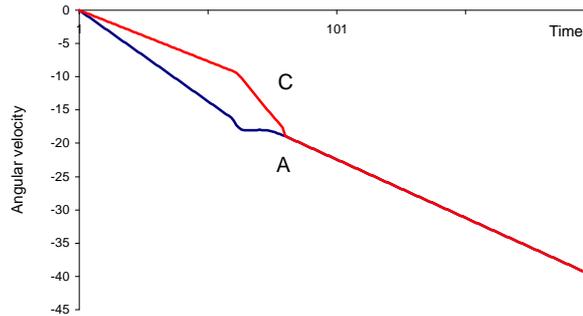


Figure 6.32. Angular velocity of the second clutch plates (node A and C) during all modes of operations

Finally, Figure 6.33, shows the relation between the angular velocities of the transformer system T1 and the load during all modes of operations.

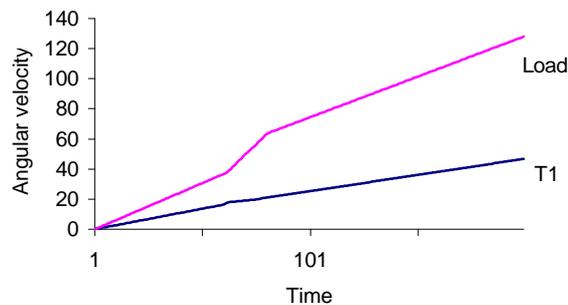


Figure 6.33. Angular velocity of transformer system T1 and the load during the three modes of operations the gearbox undergoes-ideal case

6.4. Summary

Depending on the location of the switch element in the topology graph, we have presented two algorithms for model formulation.

The first algorithm is applied when the switch is connected in series to an element. From computational point of view, toggling the switch in this case does not alter the state space dimension of the continuous system. An explicit mathematical model that contains all modes of operations is obtained by including the state of the switch as a part of the primitive system model.

The second algorithm is applied when the switch is connected between two free nodes. From computational point of view, the state of the switch affects the state space dimension of the continuous system model. In order to avoid state space reduction and to keep all variables observable from all modes we fixed the dimension of connection objects so that they remain invariant to switching. We have also shown that switching the state of switching elements from one state to another cause connection objects to change by a certain *pattern*. This implies that once we identified connection objects at one mode, connection objects in the other modes can be found by simply swapping the columns corresponding to the boundary nodes of the switch. These implications simply reduce the amount of work needed to identify and capture all possible combinations of mode models. It also allows automatic generation of models.

7 CONCLUSIONS

Mechatronics systems are hybrid systems with interacting discrete and continuous components. The continuous structure represents *the physical system* which includes the actuators, the target system and the measurement system. The discrete structure represents *the logical system* which could be seen as a mode changer, a supervisor, a state machine or a tuner depending on the application. The logical system senses and tries to control the physical system by issuing discrete logical directives at event times that will bring the physical system to a desired mode or state.

As far as modeling the logical system is concerned, our aim was restricted to formalize the logical relations between input, output and internal states and to express this formalization in array terms. The objectives of this model are to verify the logical requirements and to reveal potential conflicts and inconsistency in the overall system.

The interface between the physical system and the logical system is classified into two main categories:

3. Interface through sources. In this category, directives issued by the logical system are intended to alter the applied sources to the physical system.
4. System interface through elements and connections. In this category, directives issued by the logical system are intended to switch elements and subsystems into and out of the physical system.

Interface through sources appears when the objectives of the logical system is restricted to take logical decisions to simply switch between different input sources for the part of the physical system that will perform a certain activity. In this case, the output from the logical system shall not contribute to altering the underlying dynamical structure of the physical system. Mathematically, this implies that state variables will remain continuous as long as the input is continuous.

System interface arises when the objective of the logical system is to issue directives that are intended to connect or disconnect analog components into and out of the physical system. This sort of interface yields to altering the dynamical structure of the physical system and impose discontinuity on its behavior. The overall system then evolve in a piecewise continuous manner, where governing equation changes at event time possibly accompanied by jumps in state variables. Mathematically, this causes physical state variables to switch or jump between different modes of operations. We have focused primarily on the computational problems that arise due to this switching phenomenon. The presented modeling

framework is designed to handle these computational problems, in particular the problems of variable state space dimension and initial value problem. The practical advantages of our modeling framework could be summarized as follows:

1. Model formulation could be automated
2. Simulation is kept continuous through all mode transitions

We should point out that, the modeling framework presented here is only valid for linear systems. In linear systems, connection in the system is defined by one connection object, the incidence matrix- V . In non-linear systems, we have more complicated situation, since the connection in the system is defined by three connection objects, the displacement object, the velocity object and the acceleration object. Future work in the application of manufacturing systems theory to mode switching systems should consider systems with non-linear connection.

As far as the logical system is concerned, we have only considered the logical aspects. In real world applications, the concept of time and duration must be taken into account in order to ensure that the system will deliver at the right time in addition to be logically correct. Future work must take into account the real-time aspects.

REFERENCES

References marked with (*) are not cited in the manuscript. They are provided as further readings on the subject.

- [1] Alur, Rajeef et al, editors. *Hybrid systems III: verification and control. Vol.1066, Lectures notes in computer science.* Berlin, Springer 1996.
- [2] Anderson, Mats. *Object oriented modeling and simulation of hybrid systems.* Ph.D. dissertation, Lund Institute of Technology 1994.
- [3] Antsaklis, Panos et al, editors. *Hybrid systems II. Vol.999, Lectures notes in computer science.* Berlin, Springer 1995.
- [4] Antsaklis, Panos et al, editors. *Hybrid systems IV. Vol. 1273, Lectures notes in computer science.* Berlin, Springer 1997.
- [5] Antsaklis, Stiver, and Lemmon. *Hybrid systems modeling and autonomous control systems.* In Grossman, et al, [28], pages 366-392.
- [6] Antsaklis, Stiver, and Lemmon. *Event identification and intelligent hybrid control.* In Grossman, et al.[28], pages 268-296.
- [7] Auslander, David M. *Mechatronics: mechanical system interface.* Upper Saddle River, N.J., Prentice Hall 1996 *.
- [8] Auslander, David M. *What is Mechatronics.* IEEE/ASME Transactions on Mechatronics, Vol.1, No.1, March 1996.
- [9] Arehart, A.B., Wolovich, W.A. *Bumpless switching in hybrid systems.* In Antsaklis et al. [4], pages 1-17.
- [10] Bellman, Richard and Kalaba, Robert. *Selected papers on mathematical trends in control theory.* New York, Dover Publications 1964 *.
- [11] BEOLOGIC. *Array inference toolbox, user's manual.* Bang&Olufsen 1990*.
- [12] Berkeley, Edmund C. *Symbolic logic and intelligent machines.* New York, Reinhold 1959 *.
- [13] Bjørke, Øyvind. *Manufacturing systems theory - A geometric approach to connection.* Trondheim, Tapir 1995.
- [14] Bjørke Øyvind, Hussein, Bassam A. *A theory of manufacturing systems.* The First World Congress on

- Intelligent Manufacturing, Processes & Systems. Mayaguez, Puerto Rico, February 1995.
- [15] Buur, Jacob. *A theoretical approach to mechatronics design*. Ph.D. dissertation Institute for engineering design. Technical University of Denmark 1990.
- [16] Branicky, M.S. *General hybrid dynamical systems: modeling, analysis, and control*. In Alur, et al. [1], pages 186-200.
- [17] Branicky, M.S., Mattsson, S.E. *Simulation of hybrid systems*. In Antsaklis, et al. [4], pages 31-56.
- [18] Cellier, Francios E. *Continuos system modeling*. Springer-Verlag 1993.
- [19] Discovery Channel. Program titled *Ancient inventions*
- [20] Dorf. R.C. *Modern control systems*. Addison-Wesley Publishing Company 1980.
- [21] Dromey, Geoff. *Program derivation: The development of program forms specifications*. Addison Wesley 1989 *.
- [22] Edström, Krister. *Simulation of mode switching systems– using switched bond graphs*. Linköping University, Sweden 1996.
- [23] Franksen, Ole I. *Are data structures geometrical objects? Part I, II, III, IV. System analysis-modeling - simulation*. Vol. 1, No. 1, 2, 3, 4. 1984.
- [24] Franksen, Ole I. *The geometry of logic, from truth tables to nested arrays*. The fourth international symposium on systems analysis and simulations. Berlin 1992.
- [25] Friedman, Sander B. *Logical design of automation systems*. Englewood Cliffs, Prentice-Hall 1990.
- [26] Gabbay, et al, editors. *What is a logical system?* Oxford, Clarendon Press 1994 *.
- [27] Governor, *Encyclopædia Britannica Online*. <<http://search.eb.com/bol/topic?eu=38304&sctn=1>> [Accessed 30 June 1999].
- [28] Grossman, Robert L. et al, editors. *Hybrid systems, Vol.736, Lectures notes in computer science*. Berlin, Springer 1991.
- [29] Henzinger, et al, editors. *Hybrid systems: computation and control: first international workshop, HSCC'98*. Berkeley, California, USA. Berlin, Springer 1998.
- [30] Harashima, et al. *Mechatronics-What is it, why, and how?* IEEE/ASME Transactions on Mechatronics, Vol.1, No.1, March 1996.
- [31] Hewit, J.R. et al. *Mechatronics*. Wien, Springer 1993.

- [32] Hildre, Peter et al. *mekatronikk metodikk*. NTNU/SINTEF report (in Norwegian), 1996.
- [33] Hussein, Bassam A. *On modeling mechatronics systems*. Proceedings of the sixth UK Mechatronics forum international conference, pages 849-854. Skovde-Sweden 9-11 September 1998.
- [34] Hussein, Bassam A. *A systems approach for modeling mechatronics systems*. CIRP Annals 1998. CIRP Volume 1, pages 431-436, Switzerland 1998.
- [35] Hussein, Bassam A. *On modeling mechatronics systems- A geometrical approach*. accepted and to be published in Mechatronics-international journal- 1999
- [36] Isermann, Rolf. *Modeling and design methodology for mechatronics system*. IEEE/ASME Transactions on Mechatronics, Vol.1, No., March 1996.
- [37] Kron, Gabriel. *Non-Riemannian dynamics of rotating electrical machinery*. Journal of mathematics and physics, 1934, Vol. 1, pages, 103-194
- [38] Kron, Gabriel. *The application of tensors to the analysis of rotating electrical machinery*. General electric review, 1935, 1936, 1937
- [39] Lien, Terje K. *Digital styring for mekatronikk*, Tapir Forlag 1995.
- [40] Manugistics Inc. *APL is easy*. Manugistics 1991*.
- [41] Matko, Drago. *Simulation and modeling of continuous systems: a case-study approach*. New York, Prentice Hall 1992*.
- [42] Maxwell, J.C. *On governors*. Proceedings of the royal society of London. In Bellman et al [10], pages 70-83
- [43] McLean, M. et al. *Mechatronics, development in Japan and Europe*. Technova, Frances Pinter, London, 1983.
- [44] Merritt, Herbert E. *Hydraulic control systems*. John Wiley & Sons, Inc. 1967.
- [45] Miu, Denny K. *Mechatronics electromechanics and countermechanics*. Springer-Verlag 1996 *.
- [46] Morse, Stephen A. et al. *Control using logic based switching*. London: Springer, 1997
- [47] Møller, Gert L. *On the technology of array based logic*. Ph.D. dissertation Electric power engineering department. Technical University of Denmark 1995.
- [48] Nerode, Kohon. *Models for hybrid systems: automata topologies, stability*. In Grossman, et al. [28], pages 317-356.
- [49] Pedersen, Allan. *Digraph representation in array based logic*. Ph.D. dissertation. Electric power

- engineering department. Technical University of Denmark 1992 *.
- [50] Ruberti, A. et al. *Systems sciences and modeling*. Dordrecht, Reidel 1984.
- [51] Slotine, Jean-Jacques E. *Applied nonlinear control*. Englewood Cliffs, N.J., Prentice-Hall 1991 *.
- [52] Strömberg, Jan-Erik *A mode switching modeling philosophy*. Ph.D. dissertation. Dept. of Electrical Engineering. Linköping University 1994.
- [53] Tomkinson, Donald. *Mechatronics engineering*. New York, McGraw Hill 1996 *.
- [54] Toumi, Kamal Youcef. *Modeling, design, and control integration: A necessary step in mechatronics*. IEEE/ASME Transactions on Mechatronics, Vol.1, No., March 1996.
- [55] Tzafestas, et al. *Applied control: current trends and modern methodologies*. New York 1993.
- [56] Zhou, MengChu. *Petri net synthesis for discrete event systems*. Kluwer Academic Publishers 1993 *.
- [57] Warwick, Kevin. *Adaptive control*. In Tzafestas, et al [55], pages 253-271.

APPENDIX A: MANUFACTURING SYSTEM

A.1. Physical System Model

At the top level of the hierarchy, the workstation could be decomposed into two subsystems, a hydraulic subsystem and the boring spindle subsystem and the conveyer. The boring spindle subsystem could be further decomposed into two subsystems as shown in Figure A.1.

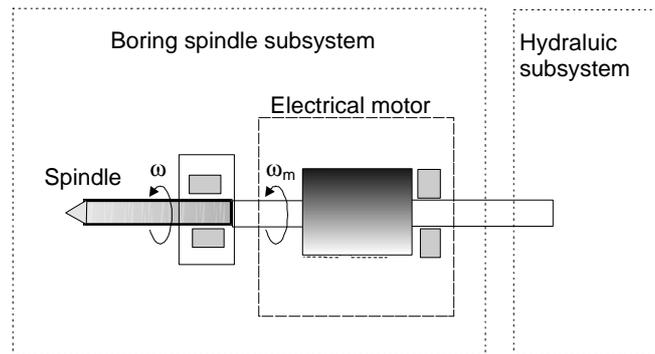


Figure A.1. Boring spindle subsystem

These subsystems are:

1. Direct current motor
2. Spindle and the load

The hydraulic subsystem shown in Figure A.2, could be decomposed into the following subsystems:

1. Cylinder actuator
2. Hydraulic pump
3. Hydraulic circuit
4. Servo valve

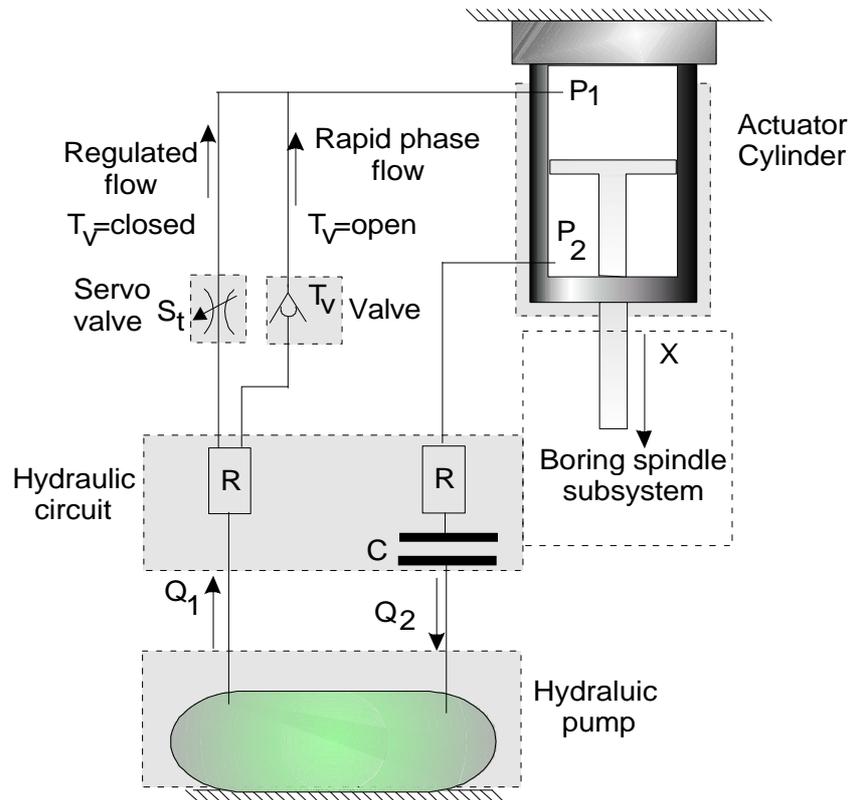


Figure A.2. Hydraulic subsystem

The hydraulic circuitry represents the paths formed by the connection of tubes and reservoir. The connection between circuit elements represents a direct connection between the components of the hydraulic subsystem. In this circuit, the volumetric flow is subjected to resistance to flow and resistance to acceleration by the reservoir. Both the hydraulic cylinder and the pump resemble a sort of connection analogous in terms to the electrical motor. They connect variables from two different domains.

The decomposition of the continuous system is shown in Figure A.3.

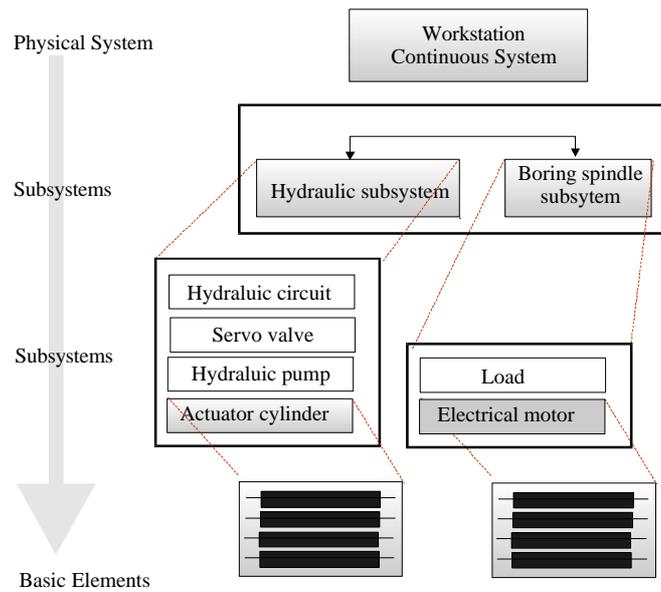


Figure A.3. Decomposition of the workstation

Breaking up the physical system into subsystems and down to basic elements will allow us to isolate a certain phenomenon we intend to model. This will provide us with a sharp insight about the evolution of the physical quantities within each subsystem yielding to better understanding of the modes and the states that each subsystem would attain. The advantages of having such insight will become visible during the design phase of a local continuous control system. The connection process is a bottom up process, we connect elements to form models of subsystems, and these models could be simulated and refined for optimal system performance. In connection, we start at the bottom level of this hierarchy and move upwards. Algebraically seen, that implies to propagate from a primitive system model to a connected system model on multiple stages by using connection objects.

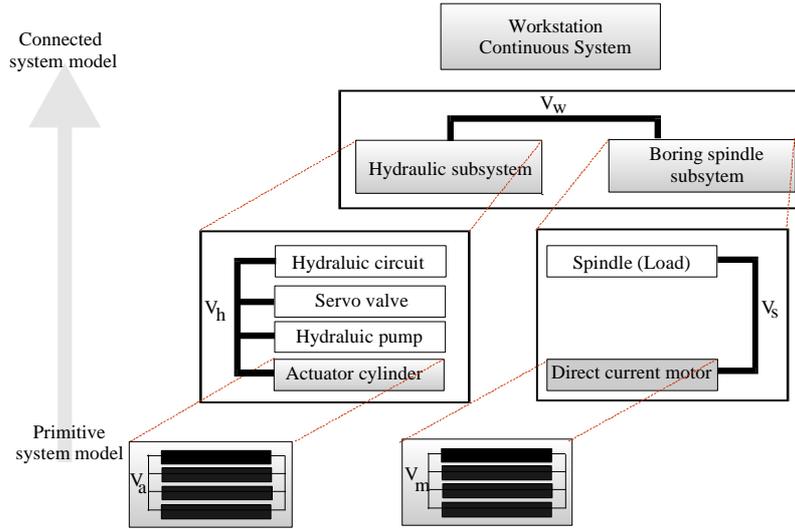


Figure A.4. Connection process

When we move up to the next level in the hierarchy, we start again with a description of a primitive system model and propagate to a connected system model using another set of connection objects. Fortunately, we do not have to repeat the above process for each single component because physical systems often contain standard components like, dc motors, step motors, sensors and so on. A model definition for such standard components could be done once and then reused for different products. Connections that resemble the internal constraints within the boundaries of each subsystem define the transformation from the primitive system model to a connected system model in the upper level. For example, in Figure A.4, connection object V_m resembles the transformation from the primitive system model of the motor to the connected system version of the motor.

A.1.1. Modeling boring spindle subsystem

The differential equations describing the connected system model of the direct current motor is given by [14]:

$$\begin{matrix}
 & f & a & z \\
 \begin{matrix} f \\ a \\ z \end{matrix} \begin{bmatrix} e_f \\ e_a \\ 0 \end{bmatrix} = \begin{bmatrix} r_f + L_f s & 0 & 0 \\ 0 & r_a + L_a s & k_m \\ 0 & -k_e & r_z + j_z s \end{bmatrix} \begin{bmatrix} i_f \\ i_a \\ w_z \end{bmatrix}
 \end{matrix}$$

where (f, a, z) refers to the axes of field windings, armature windings, and rotor shaft of the direct current motor respectively. (i) is the electrical current, (w_z) is the angular speed of the rotor shaft, (e) is the applied voltage. $(i_a k_e)$ is the generated motor torque to balance the inertia and

friction of the rotor shaft, (r_z) is the viscous damping. (r) is the electrical resistance. (k_m , k_e) are mechanical torque and back emf constants respectively. (j_z) is the moment of inertia of the rotor shaft. (L) is the electrical inductance. ($s = d/dt$) is a differential operator

The load seen by the rotor shaft is the spindle itself. Assuming a stiff spindle then the spindle can be considered as mass element with inertia property to resist angular acceleration. The dynamic equation that describe the behavior of the spindle:

$$T_L = j_L \dot{w}_L$$

Where T_L is the applied load torque on the spindle, (j_L) is the inertia of the spindle and w_L is the angular speed of the spindle

The primitive system model of the boring spindle subsystem is set up by aggregating diagonally the connected system models of the D.C. motor and the load:

$$\begin{matrix} & \text{f} & \text{a} & \text{z} & \text{L} \\ \begin{bmatrix} e_f \\ e_a \\ 0 \\ T_L \end{bmatrix} & = & \begin{matrix} \text{f} \\ \text{a} \\ \text{z} \\ \text{L} \end{matrix} \begin{bmatrix} r_f + L_f s & & & \\ & r_a + L_a s & k_m & \\ & -k_e & r_z + j_z s & \\ & & & j_L s \end{bmatrix} & \begin{bmatrix} i_f \\ i_a \\ w_z \\ w_L \end{bmatrix} \end{matrix}$$

Assuming a stiff and direct connection between the spindle and the rotor shaft then globally seen, the boring spindle subsystem will have one angular velocity (w) which is equal in magnitude and direction to the local angular velocity of the rotor shaft (w_z) and the angular velocity of the spindle (w_L). This relation is defined by the connection object V_s :

$$V_s = \begin{matrix} i_f & i_a & w \\ i_f & & \\ i_a & & 1 \\ w_z & & & 1 \\ w_L & & & & 1 \end{matrix}$$

The connected system model of the boring spindle subsystem is obtained by applying the transformation ($V_s^t \cdot Y \cdot V_s$), yielding:

$$\begin{matrix} f & a & z \\ f \begin{bmatrix} e_f \\ e_a \\ T_L \end{bmatrix} & = & \begin{matrix} f \\ a \\ z \end{matrix} \begin{bmatrix} r_f + L_f s & 0 & 0 \\ 0 & r_a + L_a s & k_m \\ 0 & -k_e & r_z + (j_L + j_z)s \end{bmatrix} \begin{bmatrix} i_f \\ i_a \\ \mathbf{w} \end{bmatrix}
 \end{matrix}$$

A.1.2. Modeling hydraulic subsystem

For the advantage of simplicity we will ignore the dynamics of the hydraulic pump because it produces a constant pressure and the actual control of the fluid flow into the actuator cylinder is done by varying the servo valve position. The dynamic response of the actuator cylinder and the load is much slower than the dynamic response of the spool valve. Therefore, the dynamics of the servo valve can also be neglected.

There are two features that must be considered when analyzing the hydraulic cylinder. The first feature appears because of the tendency of the movable elements in the hydraulic cylinder such as the piston to resist motion and acceleration due to the generated forces by the pressure inside the cylinder. In that case, forces applied and generated must be at balance, and the hydraulic piston must satisfy equilibrium condition. The second feature appears because of the characteristics of the fluid and the cylinder. In this, the fluid inside the hydraulic circuit must always satisfy the continuity principle.

Equation of force: The piston inside the actuator cylinder has two properties, resistance to a linear velocity, this property is given by viscous damping caused by the motion of the piston (r_p) and resistance to linear acceleration caused by the mass of the piston (m_p). The dynamic model of the piston is shown in Figure A.5.

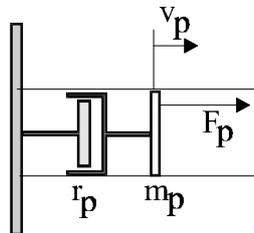


Figure A.5. Actuator cylinder model

The dynamic equation that describes the connected system model of the piston inside the cylinder which resembles the spring-mass characteristics of the hydraulic system is given by:

$$[F_p] = [r_p + m_p s][v_p]$$

The force generated by the actuator cylinder F_p is simply the differential pressure p multiplied by the working area of the actuator piston A_p :

$$F_p = p \cdot A_p$$

Continuity equation: In order to complete the analysis for the actuator cylinder, we must consider that the medium inside the hydraulic cylinder is not a vacuum. The cylinder is filled with a fluid that has specific physical characteristics that must be considered when deriving the dynamic model of the hydraulic cylinder. When volumetric flow takes place inside the cylinder. The cylinder will act like a hydraulic reservoir (capacitor) accumulating stored potential energy (pressure). Thus, the cylinder has a capacitor like property given by:

$$Q = C \cdot sp$$

Where (C) is a constant that could be calculated by the equation: $C = \frac{V_t}{4B}$

Where (V_t) is the total cylinder volume and (B) is the bulk module of the fluid. What we have presented here as the capacitance property of the cylinder is known as compressibility constant in classical hydraulic control theory.

Since the cylinder is not perfectly sealed, some of fluid volume will leak through the sealing to the outside or between the two sides of the piston inside the cylinder representing dissipated or wasted potential energy. This lost potential energy will reduce the total differential pressure inside the cylinder. Thus, the cylinder has a conductance like property given by:

$$Q = g \cdot p$$

Where (g) is a constant and known as the leakage constant in classical hydraulic theory. For actuator cylinder the leakage constant could always be ignored.

The amount of volumetric flow inside the piston is generated by the applied differential pressure on the piston area

$$Q = A_p \cdot p$$

The complete equation of continuity which describe the balance of fluid flow inside the actuator cylinder is obtained by combining the above equations:

$$Q = A_p v_p + g \cdot p + C \cdot sp$$

By combining the force equation and continuity equation, we obtain the connected system model of the hydraulic cylinder:

$$\begin{bmatrix} Q \\ 0 \end{bmatrix} = \begin{bmatrix} (g + Cs) & A_p \\ -A_p & (r_p + m_p s) \end{bmatrix} \begin{bmatrix} p \\ v_p \end{bmatrix}$$

Servo valve model: With constant pressure supply P_s , the volumetric flow rate Q_0 is proportional to the displacement of the spool valve x .

$$Q_0 = k_x x$$

The droop effect is the shift in spool valve position as a result of the load. Droop causes a slight drop in the volumetric flow rate (Q_d).

$$Q_d = k_d p$$

Net volumetric flow to the piston will be given by:

$$Q = Q_0 - Q_d = k_x x - k_d p$$

Where k_x and k_d are flow and pressure gradient of the servo valve.

By combining the equations describing the actuator cylinder and the servo valve, we obtain the connected system model of the hydraulic cylinder:

$$\begin{bmatrix} k_x x \\ 0 \end{bmatrix} = \begin{bmatrix} (g + k_d) + Cs & A_p \\ -A_p & (r_p + m_p s) \end{bmatrix} \begin{bmatrix} p \\ v_p \end{bmatrix}$$

The load seen by the hydraulic actuator is the boring spindle subsystem. The boring spindle seen as a mechanical load has two properties; resistance to linear velocity given by a viscous damping of the table the boring spindle is moving on (r_L) and resistance to linear acceleration given by the total mass of the boring spindle system (m_L). The dynamic model of the load is shown in Figure A.6.

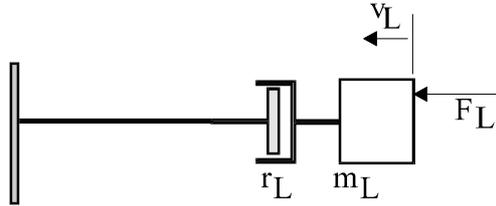


Figure A.6. Load model

The connected system model describing the load subsystem seen by the hydraulic cylinder:

$$[F_L] = [r_L + m_L s][v_L]$$

The primitive system model of the total hydraulic subsystem is set up by aggregating diagonally the connected system models of the hydraulic circuit and the load:

$$\begin{bmatrix} k_x x \\ 0 \\ F_L \end{bmatrix} = \begin{bmatrix} (g+k_d)+Cs & A_p & 0 \\ -A_p & (r_p+m_p)s & 0 \\ 0 & 0 & (r_L+m_L)s \end{bmatrix} \begin{bmatrix} p \\ v_p \\ v_L \end{bmatrix}$$

Assuming a stiff and direct connection between the load and the piston, then globally seen, the hydraulic subsystem will have one linear velocity (v) as shown in Figure A.7

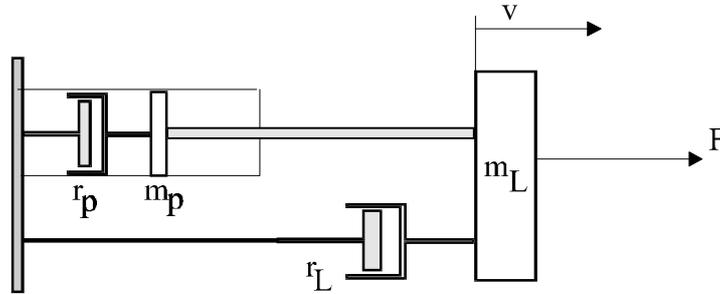


Figure A.7. Hydraulic subsystem

According to Figure A.7, the connection object between the load and the actuator cylinder will be given by:

$$V_h = v_p \begin{bmatrix} p & v \\ 1 & 0 \\ 0 & 1 \\ 0 & -1 \end{bmatrix}$$

The complete connected system model of the hydraulic subsystem is obtained by applying the transformation ($V_h^t \cdot Y \cdot V_h$):

$$\begin{bmatrix} k_x x \\ -F_L \end{bmatrix} = \begin{bmatrix} (g+k_d)+Cs & A_p \\ -A_p & (r_p+r_L)+(m_L+m_p)s \end{bmatrix} \begin{bmatrix} p \\ v \end{bmatrix}$$

Finally in more compacted form, the model of the hydraulic subsystem could be expressed as follows:

$$\begin{bmatrix} x \\ -F_L \end{bmatrix} = \begin{bmatrix} \frac{1}{k_x}((g+k_d)+Cs) & \frac{A_p}{k_x} \\ -A_p & (r_p+r_L)+(m_L+m_p)s \end{bmatrix} \begin{bmatrix} p \\ v \end{bmatrix}$$

Notice the similarity between the connected system model of the hydraulic actuator and the connected system model of the electrical actuator. They differ from each other only by the magnitude of model parameters and not in the main model structure.

A.1.3. Total physical system model

The model of the physical system is set up by aggregating diagonally the connected system models of the hydraulic subsystem and the boring spindle subsystem and applying the transformation $(V_M^i \cdot Y \cdot V_M)$. The connection matrix of the manufacturing system is unity in this case. Then the connected system model of the manufacturing system will be given by:

$$\begin{bmatrix} e_f \\ e_a \\ T_L \\ x \\ -F_L \end{bmatrix} = \begin{bmatrix} r_f + L_f s & 0 & 0 & 0 & 0 \\ 0 & r_a + L_a s & k_m & 0 & 0 \\ 0 & -k_e & r_z + (j_z + j_L)s & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{k_x}((g + k_d) + Cs) & \frac{A_p}{k_x} \\ 0 & 0 & 0 & -A_p & (r_p + r_L) + (m_L + m_p)s \end{bmatrix} \begin{bmatrix} i_f \\ i_a \\ \mathbf{w} \\ p \\ v \end{bmatrix}$$

A.2. Logical System Modeling

The behavior of the logic controller can be described by the following variables that represents the local variables in the primitive system model of the process controller. The axis number define which axis is assigned to which variable.

Axis Number	Variable Name	Description of True State	Description of False state
1	F (output)	Forward motion is On	-
2	K (input)	Start	Stop
3	E (input)	Breaker E is On	Breaker E is Off
4	S (output)	Spindle motor is On	Spindle motor is Off
5	M (input)	Breaker M is On	Breaker M is Off
6	I (output)	Rapid motion is On	Rapid motion is Off
7	R (output)	Backward motion is On	-
8	B (input)	Breaker B is On	Breaker B is Off

Table A.1. Variables of primitive system in the process controller

Connected System: The variables in the primitive system are connected together via the basic logical connectors (AND, OR and NOT) to form premises that defines the functions of the controller. These rules [39] tell simply what we expect the process controller to do:

Linguistic description	Symbolic description
R ₁ : Precede the forward motion if and only if start signal is On or the boring spindle is already moving forward and it did not reach the breaker E	$R_1: F = (K \text{ OR } F) \text{ AND } (\text{NOT } E)$
R ₂ : Open rapid motion valve I if and only if the boring spindle is moving forward and it has not yet reached the breaker M	$R_2: I = (F \text{ AND NOT } M)$
R ₃ : Start the spindle motor if and only if the boring spindle is moving forward or it reached the breaker M	$R_3: S = (F \text{ OR } M)$
R ₄ : Start the backward motion if and only if the breaker B is off and the feed forward is off.	$R_4: R = (\text{NOT } F) \text{ AND } (\text{NOT } B)$

Table A.2. Functions of the logic controller

It is implied by rule 2 that the rapid motion valve will be closed and regulated feed forward motion will start when breaker *M* is *On*. It is also implied by the first rule and the last rule that the backward motion will start when the breaker *E* is *On*. It is also implied by rule 3 that the spindle motor will continue rotating even in the backward motion this will allow a smooth depart from the work piece.

By propagating to the next level in, the above rules are aggregated using the conjunction AND. This aggregation will form the connected system model for the controller.

The multidimensional array which describe the model of the logical system is given here:

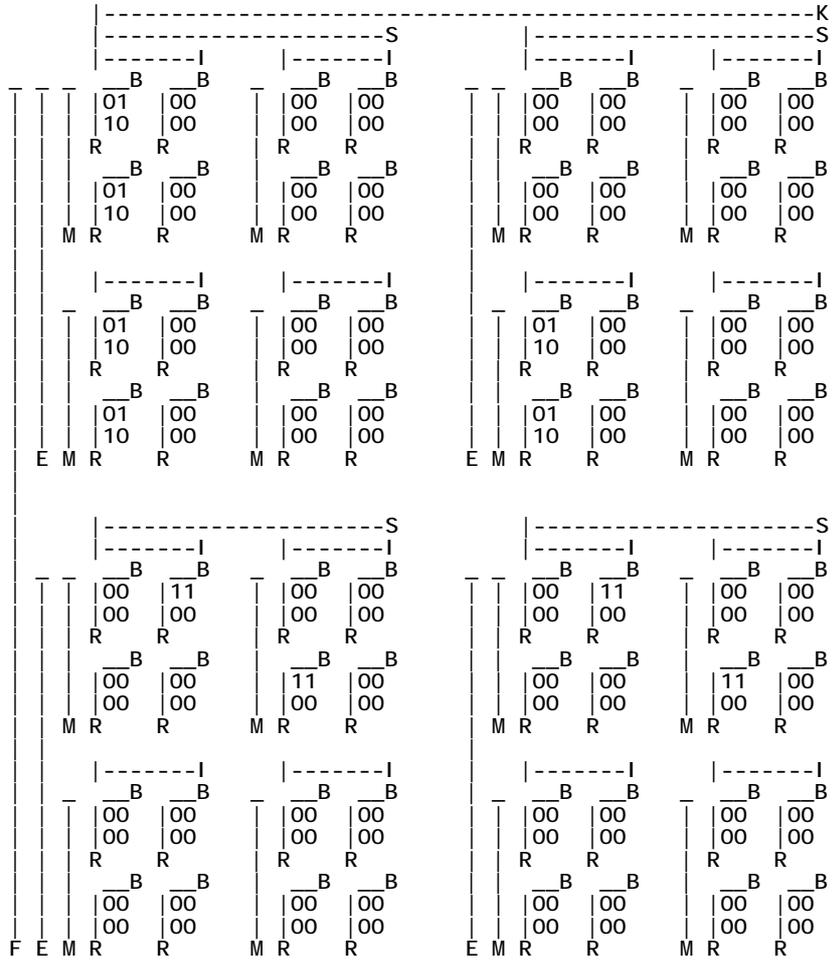


Figure A.8. Array model of the workstation

The multidimensional array P_c has 8 axes, each axis corresponds to one variable.

LIST OF FIGURES

Figure 1.1. Fundamental activities in a controlled process -----	10
Figure 1.2. Gain scheduler in an adaptive system -----	11
Figure 1.3. Examples of early machines -----	12
Figure 1.4. Flying-ball governor -----	13
Figure 1.5. Multidisciplinary nature of mechatronics -----	14
Figure 1.6. Hybrid nature of mechatronics systems -----	16
Figure 1.7. Directives from the logical system to the physical system -----	17
Figure 1.8. Control directives Q_i only alter the input u_i to the physical system ----	18
Figure 1.9. Directives issued by the logical system Q_i cause the physical system to switch between multiple modes -----	18
Figure 1.10. Switching elements from multiple physical domains -----	20
Figure 2.1. Elements, connections and sources in the system model -----	23
Figure 2.2. Elements of modeling -----	24
Figure 2.3. Roth's diagram -----	25
Figure 2.4. Generalized elements -----	26
Figure 2.5. Mass-spring-damper system -----	27
Figure 2.6. Topology graph of the mass-spring-damper system -----	29
Figure 2.7. Geometrical representation of the primitive system model -----	30
Figure 2.8. Geometrical representation of the connected system model -----	31
Figure 2.9. Logical system -----	31
Figure 2.10. Physical switch and its logical version -----	33
Figure 2.11. Logical variable constructed from continuous measurements -----	33
Figure 2.12. Equivalent representation of logical variables with 3-states -----	34
Figure 2.13. Conveyer system -----	36
Figure 2.14. Decomposition of the logical system -----	37
Figure 2.15. Connection of logical variables -----	38
Figure 2.16. Array model of the logical relation between the sensor states and the output to the motor -----	38
Figure 2.17. Logical system with bounded input -----	39
Figure 2.18. System simulation with different input combinations -----	39
Figure 2.19. System, model and controller -----	40
Figure 3.1. Interface through sources -----	42
Figure 3.2. Application example: A manufacturing system -----	43
Figure 3.3. Set points for the physical system -----	46
Figure 3.4. Simulation loops for the manufacturing system -----	47
Figure 3.5. Linear velocity (v) and differential pressure (p) in the actuator cylinder during rapid phase feed forward motion -----	49
Figure 3.6. Linear velocity (v) and differential pressure (p) in the actuator cylinder during controlled feed forward motion -----	50
Figure 3.7. Linear velocity (v) and differential pressure (p) during backward motion -----	51
Figure 3.8. Differential pressure (p) and linear velocity (v) of the actuator cylinder through one cycle of operation -----	51
Figure 4.1. Physical system with multiple mode models -----	53

Figure 4.2 A dynamic system consisting of mass element moving on a friction-less surface -----	54
Figure 4.3. There is a discontinuity in the output when we switch the control law without setting the state -----	55
Figure 4.4. Electrical network with a switch -----	56
Figure 4.5. Property graph of the circuit when the switch is open -----	56
Figure 4.6. Circuit layout and its property graph when the switch is closed -----	57
Figure 4.7. Switching elements from multiple physical domains -----	59
Figure 5.1. Mechanical system with non ideal clutch -----	62
Figure 5.2. Angular rotation of the two axial shafts (node A and node B) -----	63
Figure 5.3. Angular velocity of the two axial shafts (node A and node B) -----	64
Figure 5.4. Simplified diagram for the gearbox -----	65
Figure 5.5. Property graph of the gear box -----	66
Figure 5.6. Normal forces applied on the clutch's plates -----	66
Figure 5.7 Modes of operations for the gearbox -----	67
Figure 5.8. Gearbox in its second mode (sliding mode) -----	68
Figure 5.9. Output angular velocity of the load and transformer systems T1 and T2 -----	70
Figure 5.10. Angular velocity of the second clutch plates (node A and node C) ---	71
Figure 5.11. Angular velocity the first clutch plates (node B and node D) -----	71
Figure 6.1. Switch is closed, flow variable is different than zero and potential drop is equal to zero -----	73
Figure 6.2. Switch is open, flow variable is equal to zero and potential variable is different than zero -----	74
Figure 6.3. Domain of variables in switching and primitive elements -----	74
Figure 6.4. Node connection procedure -----	75
Figure 6.5. Switch as a connection element- closed state -----	75
Figure 6.6. Switch is used to connect or disconnect elements -----	76
Figure 6.7. Switch is used to connect or disconnect nodes -----	76
Figure 6.8. Switch is connected in series with an element -----	77
Figure 6.9. Switching circuit -----	78
Figure 6.10. Voltage drop of node A and B -----	80
Figure 6.11. A dynamic system consisting of mass element (m) moving on a frictionless surface under the applied force (f) -----	80
Figure 6.12. Dynamic system with ideal switching element -----	81
Figure 6.13. Pump-less switching obtained by a hybrid formulation of continuous system model -----	82
Figure 6.14. Stiff connection as a result of closing the switch -----	82
Figure 6.15. Switch representation in the property graph -----	83
Figure 6.16. Swapping of elements between node A and node B when the switch is closed -----	83
Figure 6.17. Property graphs for both modes -----	83
Figure 6.18. Mechanical system with an ideal switching element -----	85
Figure 6.19. Property graphs of the mechanical system in both modes of operations -----	86
Figure 6.20. Angular velocity of node A and node B (ideal switching case) -----	88
Figure 6.21. Angular velocity and angular rotation of Node B (ideal switching case) -----	88
Figure 6.22. Property graph with an ideal switching element -----	89
Figure 6.23. Reconstructed graph -----	89

Figure 6.24. Property graph before and after switching -----	90
Figure 6.25. Simplified diagram for the gearbox -----	91
Figure 6.26. Modes of operations for the gearbox-----	91
Figure 6.27. Property graph of the gear box- ideal case -----	92
Figure 6.28. State of the switches during sliding mode -----	92
Figure 6.29. State of the switches during first mode -----	94
Figure 6.30. State of the switches at third mode of operation-----	95
Figure 6.31. Angular velocity of the first clutch plates (node B and D) during all modes of operations -----	96
Figure 6.32. Angular velocity of the second clutch plates (node A and C) during all modes of operations -----	97
Figure 6.33. Angular velocity of transformer system T1 and the load during the three modes of operations the gearbox undergoes-ideal case -----	97
Figure A.1. Boring spindle subsystem -----	105
Figure A.2. Hydraulic subsystem-----	106
Figure A.3. Decomposition of the workstation -----	107
Figure A.4. Connection process -----	108
Figure A.5. Actuator cylinder model-----	110
Figure A.6. Load model -----	112
Figure A.7. Hydraulic subsystem-----	113
Figure A.8. Array model of the workstation -----	116

LIST OF TABLES

Table 1.1. Meaning of the binary variable S_w and its complement \tilde{S}_w in some physical domains.....	21
Table 2.1. Description of the primitive system variables.....	37
Table 3.1. Variables involved in the description of the logical system.....	44
Table 3.2. State of input signals to the logical system.....	48
Table 3.3. State of output signals: rapid phase feed forward motion.....	48
Table 3.4. State of input signals for controlled feed forward motion.....	49
Table 3.5. State of output signals: controlled feed forward motion.....	49
Table 3.6. State of input signals: Backward motion.....	50
Table 3.7. State of output signals: Backward motion.....	50
Table A.1. Variables of primitive system in the process controller.....	114
Table A.2. Functions of the logic controller.....	115

LIST OF EXAMPLES

Example 2-1. Mass-spring-damper system.....	27
Example 2-2. Conveyer system.....	36
Example 4-1. Mass element on a friction-less surface.....	54
Example 4-2. Electrical network with a switch.....	55
Example 5-1. Mechanical system with non-ideal clutch.....	62
Example 6-1. Switching circuit.....	78
Example 6-2. Mechanical system with ideal clutch.....	85
Example 6-3. Property graph with ideal switch.....	89