Logic-Based Optimal Control for Shipboard Power System Management

A Thesis

Submitted to the Faculty

of

Drexel University

by

Edoe F. Mensah

in partial fulfillment of the

requirements for the degree

of

Doctor of Philosophy

March 2008

Dedications

To the memory of my father and my mother

To my brothers and sisters for their continuous encouragement

To my family

Acknowledgements

It has been a long educational journey, a journey that once seemed improbable but today, in this month of March 2008, the completion of this thesis brings me to my final destination. It is a journey which couldn’t have been possible without the contribution of the following people. I will begin by the person who gave me the opportunity to be part of this research project from the very beginning, my advisor Dr Kwatny to whom I am infinitely grateful. He has been a source of inspiration and a role model for me. It is truly a blessing to have studied under his guidance and I am looking forward to working with him in the future. I would like to thank Dr Niebur, my secondary advisor who had provided me with all her support, she never hesitated to point me to right direction and information. I thank Dr Chang for his encouragements and advice. To Dr Nwankpa, I say thank you for your kind attention and for letting me be part of the CEPE lab. To Dr Bajpai, thank for your support and for making sure that our research project is completed in a timely manner.

For the last three years, I have had the privilege to be part of the Center for Electric Power Engineering (CEPE), I would like to thank all its members with whom I had cordial relationship. I will really miss the CEPE activities.

A special thank goes to Ekrem Gursoy who had helped me numerous times in this thesis formatting and proof reading and thank you for being a friend. I thank Adam Beytin at Techno-sciences, Inc. a member of the project group, who was always able to come up with alternative solution when needed. I thank Ralph Beam for being diligent in designing the SimPower simulation tool of the DDG 1000 IPS and for helping me with some simulation results.

I thank Jean-Etienne Dongmo with whom I had partially shared office for the last three years; your preliminary work on the DAE solver had elevated this research.

I thank Myke Knauff and Sudipta Lahiri for sharing the sharing the Bossone power lab with me.

To my parents who I admire and love, from where you are now, I know you are taking great pride in my accomplishment.

Finally my thanks go to my brothers and sisters, whose educational achievement had set me on right path from my adolescence to now, thank you for believing in me.

Table of Contents

List of Tables

List of Figures

Abstract

Logic-Based Optimal Control for Shipboard Power System Management

Edoe F. Mensah
Advisor: Harry G. Kwatny, Ph.D.

The capability to dynamically reconfigure future naval integrated electric power systems is central to the Navy's vision of the future combat ship. The objective in this thesis is to design, implement and evaluate a Shipboard Power System Management system that will prevent loss of power at critical buses when damage conditions are encountered. The approach that we are proposing is based on a new paradigm for the design of optimal control systems for hybrid systems, i.e., systems composed of continuous dynamics and discrete events. Discrete events may involve external disturbances, the discrete action of protection devices or control systems. The essence of the idea is that the discrete acting subsystems are naturally associated with a set of logical conditions or logical and the continuous system dynamics are usually described by differential equations or differential-algebraic equations. We will introduce a dynamic programming method for hybrid systems that solves dynamic optimization problems involving both binary and real variables. The stability analysis of the hybrid control systems is conducted via bifurcation control analysis. The state feedback controller strategy for the mode switching of the power system is obtained through Mixed Integer Dynamic Programming. It is computed in the form of a lookup table that represents a mapping from combinations of modes, and continuous states to the required switching actions. Simulations results will be analyzed.

CHAPTER 1: INTRODUCTION

1.1 Motivation

The DDG 1000, Littoral Combat Ship (LCS) and CG(X) form the Navy's next-generation surface combatant family of ships. Together these warships will provide the complementary mission capabilities and affordably satisfy the full spectrum of operational requirements demanded well into the 21st century. The DDG 1000 fills an immediate and critical naval-warfare gap for today's warfighter, meeting validated Marine Corps fire support requirements and supporting the joint warfighting doctrine [73].

What is DDG 1000?

Developed under the DD(X) destroyer program, DDG 1000 Zumwalt is the lead ship of a class of next-generation multi-mission destroyers tailored for land attack and littoral dominance. The DDG 1000 will provide forward presence and deterrence, and operate as an integral part of joint and combined expeditionary forces.

Secretary of the Navy Donald Winter named the lead ship and class in honor of Admiral Elmo R. "Bud" Zumwalt Jr., Chief of Naval Operations from 1970 to 1974.

The DDG 1000 Zumwalt is a key component of the surface combatant 'family of ships' currently being developed by the Navy to serve as the backbone of tomorrow's surface

Fleet. In addition to the DDG 1000 multi-mission destroyer, the family of ships will consist of the focused mission Littoral Combat Ship (LCS) and an air-dominance cruiser, CG(X). These ships, together with in-service Aegis combatants, present an affordable balance between force size and capabilities to meet current and project threats.

The DDG 1000 Zumwalt provides a broad range of capabilities that are vital both to supporting the Global War on Terror and to fighting and winning major combatant operations. Zumwalt's multi-mission warfighting capabilities are designed to counter not only the threats of today, but threats projected over the next decade as well.

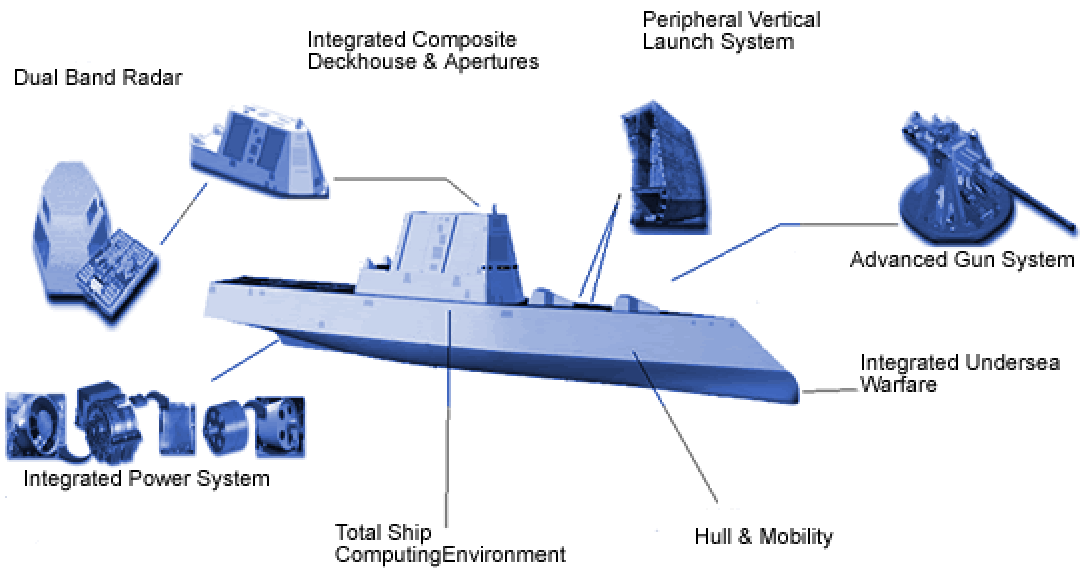

Dual Band Radar
Integrated Composite Deckhouse & Apertures
Peripheral Vertical Launch System
Advanced Gun System
Integrated Undersea Warfare
Integrated Power System
Total Ship ComputingEnvironment
Hull & Mobility

Figure 1.1-1: The DDG 1000 Systems

Integrated Power System (IPS)

An integrated power system (IPS) is an all-electric architecture for future ships, providing electric power to the total ship (propulsion and ship service) with an integrated plant. IPS enables a ship's electrical loads, such as pumps and lighting, to be powered from the same electrical source as the propulsion system (e.g., electric drive), eliminating the need for separate power generation capabilities for these loads. In commercial applications, this is known as the "power station" concept.

Anticipated benefits of IPS include: Fewer prime movers: Usually allows for a reduction from a total of seven to a total of five prime movers in the traditional gas-turbine surface combatant. Reduced costs of ownership: Results in significant fuel savings (15-19% in a typical gas-turbine combatant). Fewer engines installed results in less maintenance and manning. Naval architectural flexibility: Provides flexibility in locating prime movers, allowing space previously used for uptakes to be put to better use. Improved survivability and stealth: Quiet propulsion motors can better meet current acoustic requirements. Smaller main machinery spaces allow for improved damage control. Improved warfighting: Integrated power makes large amounts of power available throughout the life of the ship. This power can be reallocated to accommodate future combat systems. Advances in power conversion are making it possible to provide uninterrupted power, advanced fault isolation, and "fight through" capabilities beyond what is currently available.

In a typical mechanical drive propulsion system, the propulsion prime movers are connected to long shafts running through the ship to large reduction gears that rotate the ship's propellers. With electric drive, the prime movers rotate electric generators that are connected through cabling to motor drives and electric motors that rotate a ship's propellers. Electricity is the medium for transmitting the energy of the prime mover. It enables "cross connecting" of any available prime mover/generator combination by breaking the physical link between the power generation and power utilization components. IPS provides for all of a ship's electrical needs, including propulsion and ship service loads. Electric drive only provides for propulsion. It does not include power for ship service loads.

The Navy has used electric drives in many ships, including early aircraft carriers, a number of ships during World War II, and many of the current inventory of smaller auxiliary ships. In fact, the Navy is leveraging as much as possible what is happening in the cruise ship industry, where nearly all new ships are integrated electric. What is new and significant is the application of these concepts in a fully electrically integrated (no mechanical takeoffs for power) power system on a surface combatant. These ships have higher speed and lower noise requirements than any of the other ships, as well as large combat systems to support. Commercial systems would be too big and too noisy for a surface combatant, and do not have a power system architecture to let them survive damage and continue to fight [72].

1.2 Objective

The objective of this thesis is to develop a new approach to the design of ship power management systems. The goal of a Power Management System (PMS) controller is to prevent loss of power at critical buses of the IPS when damage conditions are encountered. This system includes discrete actions, such as load shedding intended to preserve network integrity during battle conditions. The approach that we propose will optimally shed load and/or activate an uninterruptible power supply, switch vital load feeder or the shift power supply distribution between generators following the occurrence of disruptive events.

Our approach is based on a new paradigm for the design of optimal control systems for hybrid systems, i.e., systems composed of continuous dynamics and discrete events. Discrete events may involve external disturbances, the discrete action of protection devices or control systems. The essence of the idea is that the discrete acting subsystems are naturally associated with a set of logical conditions or logical specifications and the continuous system dynamics are usually described by differential equations or differential-algebraic equations. The key idea in our approach is to symbolically transform the logical specification that describes the discrete subsystem to a set of inequalities in binary-valued variables or mixed binary real-valued variables. The resulting inequalities are called integer programming formulas, or simply IP formulas or IP forms. We also develop a dynamic programming method tailored to hybrid systems that solves dynamic optimization problems involving both binary and real variables [1], [6]. The design of a state feedback controller for the mode switching of the power system is obtained through mixed-integer dynamic programming. The controller is computed in the form of a lookup table that represents a mapping from combinations of modes, and continuous states to the optimal switching actions. Two examples will be studied: 1) a 3-bus power system with Uninterruptable Power Supply (UPS) and 2) a Shipboard Integrated Power System (IPS) based on a notional configuration of the U.S. Navy's DDG 1000 destroyer. We will conduct a stability analysis via bifurcation control on the

3-bus power system as well as a complete modeling, design and simulation of the PMS controller. We will make survivability and reconfigurability important functionality of the PSM. While in the 3-bus power system it is relatively easy to exactly solve the algebraic part of the Differential-Algebraic Equation (DAE) model, in the case of the DDG 1000 IPS there is no way to avoid working directly with the DAE model, therefore a discrete numerical algorithm will be developed and integrated into the dynamic programming algorithm.

1.3 Contribution of this research

It is now being recognized that the modeling and design of complex systems will required a global hierarchical controller design that acknowledges the importance of discrete as well as continuous actions. For such complex systems, a hybrid optimal control is a viable approach [2], [63]. The overall contribution of this thesis is the working progress toward the development of new computational tools integrating continuous and discrete event dynamics for the modeling and design of control systems for power system management. In that regard several specific contributions had been made, they are described in the subsections below.

1.3.1 Discrete Event Transition Dynamics as Logical Specifications

Discrete event transition dynamic is traditionally obtained using partial function, e.g. $f(x,e) = y$ where x is state and upon the occurrence of an event e the system transitions to the state y. For a partial function not all transitions are defined for every event [74]. A equivalent transition can be written as $(x \wedge e) \Rightarrow y$, which signifies that if the system is in state *x* and the event *e* occurs then system will transition to state *y*. As we will see in the rest of this thesis, a transition dynamic written as logical specification can be transformed to inequality constraints. The interpretation of a transition diagram as logical specification has an important computational advantage that can be exploited in optimization routines.

1.3.2 Converting Logical Specifications to IP Formulas

A very efficient Mathematica package which generates integer programming inequalities is called GenIP [1]. Its efficiency comes from the fact that we eliminate the lengthy geometric computation of convex hull described in the Hybrid System Description Language (Hysdel). The GenIP package was initiated by a group of researchers whose desire was to introduce logic in optimization of complex decision problems.

Several additions had been made to the GenIP package.

- Conjunctive Normal Form (CNF) and Disjunctive Normal Form (DNF).

They can be used to reduce the number of inequalities, by providing tighter bounds for the inequalities.

- Exclusive Or and Exactly predicates

The Exclusive Or (XOR) is a bin-ary operator between two prepositions. Due to the fact that our model of hybrid automaton is a sequential machine rather than a concurrent state machine, the Exclusive Or could be used to impose a constraint on a system in order to guarantee mutual exclusive property necessary in the hybrid system formulation.

Notation: let q_1 and q_2 be two literals we have $q_1 \otimes q_2 = exactly(1,\{q_1, q_2\})$ expresses the fact that no matter the situation, exactly one of the two literals is true and the other is false.

The predicate exactly can constitute the generalization of the Exclusive Or. A generalization can be written as $q_1 \otimes q_2 \otimes ... \otimes q_n = exactly(1,\{q_1, q_2, ..., q_n\})$. The importance of this generalization will be exploited in hybrid system modeling to guarantee that one and only state is valid during the evolution of the system trajectory.

1.3.3 Mixed-Integer Dynamic Programming

Transition diagram is the basic tool for modeling any discrete event or hybrid system. Since we have selected a logic-based modeling of transition diagrams, after conversion of the logical specification, we obtained a set of inequalities constraints called IP formulas.

The mixed-integer aspect of optimization comes from the fact that the IP formulas are sometimes a mix of integer values and real values constraints. Continuous-time optimal control problem involving constraints and dynamics are solved using a special purpose mixed-integer dynamic programming algorithm. Two forms of mixed-integer dynamic programming were implemented: one form dealing with differential equations and the other form dealing with the differential-algebraic equations.

1.3.4 Applications

- **DC-DC power conditioning**: An optimal switching feedback controller is synthesized. The results obtained from simulation agree with our expectations. This case study constitutes the first application of our hybrid system design. Any detail information on this case study can be found in [49].

- **A 3-bus power system example**: A more challenging power system composed of

a generator and a transmission line feeding an aggregate induction motor load. The system is equipped with a UPS providing backup power. The network algebraic equations were solved via formal and informal quantifier elimination [6-10].

Bifurcation control

Bifurcation control deals with designing methods to prevent the occurrence of bifurcation point. In the 3-bus power system example the algebraic equation is relatively simple therefore an (informal) quantifier elimination method was used to derive the network characteristic equation, its interaction with the load characteristic equation yield system's operating equilibrium. The goal of bifurcation control is to use discrete parameter to access the stability of the power system before and after the occurrence of bifurcation point.

Feedback controller via lookup table and Stateflow

The state feedback controller strategy for mode switching of the power system is obtained through Mixed Integer Dynamic Programming. It is computed in form of a lookup table that presents a mapping from combinations of modes, events, generator angle, slip conditions and battery state to switching actions. Scopes are provided for viewing the performance variables such as the internal voltage E, regulated voltage V_2, the state of charge of the battery σ as well as mode switching. The simulation block diagram is composed of a optimal state feedback controller connecting the power plant. The system can be assembled with the Mathematica package *ProPac* as a SIMULINK model targeted for simulation in MATLAB/SIMULINK/Stateflow.

- **DDG 1000 Notional System**

Our model of the DDG 1000 Integrated Power System is a slight modification of the model put in place by SYNTEK[50],[51]. The system is composed of a two generators feeding two induction motors through transmission lines. Vital loads as well as Non-vital loads are connected to various buses. The model is equipped with a UPS that can supply voltage to the vital load when the power transfer exceeds its capacity limit.

Reconfiguration

Discrete states corresponding to admissible reconfigurations can now be defined and system models developed for each discrete state. In addition, a transition structure can be defined to express allowable transitions between the discrete states. While it is always possible to allow transitions from every discrete state to any other discrete state, the specification of a transition structure has many benefits. A specification allows us to impose constraints on the reconfiguration process and to eliminate unsuitable transitions.

DAE for Hybrid system

The DDG 1000 IPS is a larger system compare with the 3-bus power system example. The differential-algebraic equations that model its dynamic is much more difficult to solve due to the complexity of the algebraic equation which cannot be solved by quantifier elimination methods. We devised a differential-algebraic algorithm which includes discrete event actions.

SIMULINK/SimPower

SimPower is an add on package of SIMULINK which provides a graphical toolset that allow for modeling of electrical, mechanical and control systems within a power system. The use of this software offers various advantages over using a model based on DAEs. It provides a well developed system of calculating power system conditions, avoiding the complications of developing a set of equation to describe network behavior and necessary approximations to make such a model function. Also the SimPower model allows for easy modification of fault conditions and other potential damage to the systems. Further, using SimPower allowed for the use of its blocks developed to model each individual component. We were able to model the DDG 1000 Integration Power System in SimPower

CHAPTER 2: PROPOSITION LOGIC AND PREDICATE LOGIC

2.1 Introduction

It is only three decades ago that the use of logic has permeated the field of operations research for the purpose of solving complex decision making problems. Propositional logic and predicate logic had become the basic tools for manipulating logic in a computational framework. In this chapter we will introduce the propositional logic and the predicate calculus language. The connection between logic specification and integer-programming formulas will be made and finally we will introduce a Mathematica-based automated conversion process for logic specification.

2.2 Modeling Framework

2.2.1 Propositional logic

The basic idea is relatively straightforward. An atomic proposition q is a variable that can assume the Boolean values True or False. Propositional logic formulas or formulas are constructed by combining atomic propositions using logical connectives:

$"\neg"$, $"\vee"$, $"\wedge"$, $"\Rightarrow"$, $"\Leftrightarrow"$ stand for not, and, or, if then, if only if respectively.

Negation

An atomic proposition q is also called a literal and its negation $\neg q$, is False if q is True.

A propositional variable will be referred to as positive literal, while its negative is a negative literal and each will be referred as opposite of the other.

In general we can associate a Boolean variable δ_{q_i} with each literal q_i such that $\delta_{q_i} = 1$ or 0 if $q_i = \text{True or False}$. That is δ_{q_i} represents the truth value of q and $1-\delta_{q_i}$ represents the truth value of $\neg q$.

2.2.2 Disjunctive Boolean Expressions:

A disjunction of literals is called a clause, for example $p \vee \neg q \vee r$ is a clause.

Let $q_1,\ q_2\ ,..,\ q_m,\ \neg q_1,\ \neg q_2\ ,..,\ \neg q_n$ be Boolean variables and define $M_j = \{q_1, q_2, ..., q_m\}$ $N_j = \{q_1, q_2, ..., q_n\}$ as positive and negative literals where $|M_j| = m$ and $|N_j| = n$.

The disjunctive Boolean expression is $q_1 \vee q_2 \vee ... \vee q_m \vee ... \vee \neg q_1 \vee \neg q_2 \vee ... \vee \neg q_n$.

The disjunctive expression can be transformed into an inequality as

$$1 \le \sum_1^m \delta_{q_i} + \sum_1^n (1-\delta_{q_j}) \qquad (2.1)$$

Examples:

Using (2.1) the Boolean expression $q_1 \vee q_2$ is converted to the linear inequality $1 \le \delta_{q_1} + \delta_{q_2}$

Using (2.1) the Boolean expression $q_1 \vee q_2 \vee \neg q_3$ is converted to the linear inequality $0 \le \delta_{q_1} + \delta_{q_2} - \delta_{q_3}$

Table 2-1: The "Inclusive Or"

q_1	q_2	$q_1 \vee q_2$
T	T	T
T	F	T
F	T	T
F	F	F

Remark: when we use the symbol "$\vee$" , we mean inclusive disjunction or Inclusive Or.

An Inclusive Or is false only in the case both literal q_1 and q_2 are false.

The Boolean expression $q_1 \vee q_2$ can be written as a sum term: $q_1 \vee q_2 \equiv q_1 + q_2$.

2.2.3 Conjunctive Boolean Expression:

Let $q_1, q_2,.., q_m, \neg q_1, \neg q_2,.., \neg q_n$ be Boolean variables. If we consider the conjunctive Boolean expression $q_1 \wedge q_2 \wedge ... \wedge q_m \wedge \neg q_1 \wedge \neg q_2 \wedge ... \wedge \neg q_n$ as a product term we can write the following inequality:

$$1 \le \prod_1^m \delta_i \prod_1^n (1-\delta_j) \qquad (2.2)$$

which further yields the set of inequalities

$\delta_{q_1} \ge 1,\ \delta_{q_2} \ge 1,..,\delta_{q_m} \ge 1, (1-\delta_{q_3}) \ge 1,..,(1-\delta_{q_n}) \ge 1$

Examples:

Using (2.2) the Boolean expression $q_1 \wedge q_2$ can be converted to the expressions, $1 \le \delta_{q_1} \bullet \delta_{q_2}$ or equivalently $\delta_{q_1} \ge 1,\ \delta_{q_2} \ge 1$.

$q_1 \wedge q_2 \wedge \neg q_3$ is converted to the expressions $\delta_{q_1} \bullet \delta_{q_2} \bullet (1-\delta_{q_3}) \ge 1$ corresponding to $\delta_{q_1} \ge 1,\ \delta_{q_2} \ge 1,\ (1-\delta_{q_3}) \ge 1$.

Remark:

The Boolean expression $q_1 \wedge q_2$ can be written as a product term: $q_1 \wedge q_2 \equiv q_1 \bullet q_2$.

2.2.4 Conjunctive Normal Form: CNF

The conjunctive normal form is the conjunction of disjunctive Booleans expressions in a form $\wedge_{j\in L}\left(\vee_{i\in M_j} q_i \vee_{i\in N_j} \neg q_i\right)$, this expression is also called product of sums term.

The utility of the CNF comes from the fact that it can be transformed into product and sum terms.

The general transformation rule of Boolean expression in CNF in a form $\wedge_{j\in L}\left(\vee_{i\in M_j} q_i \vee_{i\in N_j} \neg q_i\right)$ is

$$\begin{gathered} 1 \le \sum\nolimits_{i\in M_1} \delta_{q_i} + \sum\nolimits_{i\in N_1}\left(1-\delta_{q_i}\right) \\ \vdots \\ 1 \le \sum\nolimits_{i\in M_\ell} \delta_{q_i} + \sum\nolimits_{i\in N_\ell}\left(1-\delta_{q_i}\right) \end{gathered} \tag{2.1}$$

where $\ell = \text{length } L$.

The CNF formula $(q_1 \vee \neg q_2) \wedge (q_2 \vee \neg q_3) \wedge (q_1 \vee \neg q_4)$ can be transformed as follows:

$$\left.\begin{array}{l} q_1 \vee \neg q_2 \\ q_2 \vee \neg q_3 \\ q_1 \vee \neg q_4 \end{array}\right\} \Leftrightarrow \left.\begin{array}{l} 1 \le \delta_{q_1} + 1 - \delta_{q_2} \\ 1 \le \delta_{q_2} + 1 - \delta_{q_3} \\ 1 \le \delta_{q_1} + 1 - \delta_{q_4} \end{array}\right\} \tag{2.2}$$

2.2.5 Implication: Imply “$(\Rightarrow)$” or If then

The implication connective involves the conditional statement of two literals, $q_1 \Rightarrow q_2$ read "q_1 implies q_2" or " If q_1 then q_2". The literal q_1 is called the antecedent and q_2 is called its consequent. The conditional statement " If q_1 then q_2" is known to be false only in the case where the conjunction $q_1 \wedge \neg q_2$ is true. Therefore the conditional statement " If q_1 then q_2" is true if $\neg(q_1 \wedge \neg q_2) = \neg q_1 \vee q_2$ true. This equivalent statement $q_1 \Rightarrow q_2 \equiv \neg q_1 \vee q_2$ provides a practical way to evaluate the truth or the falsehood of the “if then” statement in Table 2-2.

Table 2-2:Implication

q_1	q_2	$\neg q_2$	$q_1 \wedge \neg q_2$	$\neg(q_1 \wedge \neg q_2)$	$\neg q_1 \vee q_2 \equiv q_1 \Rightarrow q_2$
T	*T*	*F*	*F*	*T*	*T*
T	*F*	*T*	*T*	*F*	*F*
F	*T*	*F*	*F*	*T*	*T*
F	*F*	*T*	*F*	*T*	*T*

Notice that $q_1 \Rightarrow q_2$ is logically equivalent to $\neg q_1 \vee q_2$ and an implication is false only when it antecedent is true and it consequent is false.

Using the equivalent relation in Table 2-2 and (2.1) , the implication $q_1 \Rightarrow q_2$ can be converted to the inequality $1 \le 1 - \delta_{q_1} + \delta_{q_2}$ or $\delta_{q_2} - \delta_{q_1} \ge 0$.

2.2.6 Conditional Equivalent: If and only if $(\Leftrightarrow)$ or $(\Rightarrow) \wedge (\Leftarrow)$

Two logical statements are said to be "equivalent" when they are either both are true or both are false. Equivalent is also termed bi-conditional.

The truth table below describes the derivation of the equivalent connective.

Table 2-3: If and only if

q_1	q_2	$q_1 \Rightarrow q_2$	$q_2 \Rightarrow q_1$	$(q_1 \Rightarrow q_2) \wedge (q_2 \Rightarrow q_1) \equiv q_1 \Leftrightarrow q_2$
T	*T*	*T*	*T*	*T*
T	*F*	*F*	*T*	*F*
F	*T*	*T*	*F*	*F*
F	*F*	*T*	*T*	*T*

2.2.7 Exclusive Or

The "exclusive or" (XOr) Boolean operator is often denote "$\oplus$", is the type of logical disjunction on two operands that results in a value of "true" if and only if exactly one of the operands has a value "true".

The following are equivalent expression for "exclusive or"

$$\begin{aligned}
q_1 \oplus q_2 &= (q_1 \wedge \neg q_2) \vee (\neg q_1 \wedge q_2) \\
q_1 \oplus q_2 &= (q_1 \vee q_2) \wedge (\neg q_1 \vee \neg q_2) \\
q_1 \oplus q_2 &= (q_1 \vee q_2) \wedge \neg (q_1 \wedge q_2) \\
q_1 \oplus q_2 &= \neg((q_1 \wedge q_2) \vee (\neg q_1 \wedge \neg q_2))
\end{aligned} \tag{2.3}$$

The first expression in (2.3) shows that the "exclusive or" is equivalent to the negation of a logical bi-conditional, by the rules of implication and equivalence.

The truth table in Table 2-4 illustrate the "Exclusive Or" logical behavior.

Table 2-4: Exclusive Or

q_1	q_2	$q_1 \wedge \neg q_2$	$\neg q_1 \wedge q_2$	$(q_1 \wedge \neg q_2) \vee (\neg q_1 \wedge q_2) \equiv q_1 \oplus q_2$
T	T	F	F	F
T	F	T	F	T
F	T	F	T	T
F	F	F	F	F

Remark: The "Exclusive Or " is a natural Boolean operator to implement mutual exclusive property logic systems.

Summary:

Table 2-5: Logic relation and Linear inequalities

Logical Relation	*logical specification*	*Linear inequalities*
$\vee$ *(Or)*	$q_1 \vee q_2 \vee \ldots \vee q_n$	$1 \le \delta_{q_1} + \delta_{q_2} + \ldots + \delta_{q_n}$
$\wedge$ *(And)*	$q_1 \wedge q_2 \wedge \ldots \wedge q_n$	$\delta_{q_1} \ge 1, \delta_{q_2} \ge 1, \ldots, \delta_{q_n} \ge 1$
$\Rightarrow$ *(If then)*	$q_1 \Rightarrow q_2$ *i.e.* $(\neg q_1 \vee q_2)$	$0 \le \delta_{q_2} - \delta_{q_1}$
$\Leftrightarrow$ *(iff)*	$q_1 \Leftrightarrow q_2$ *i.e.* $(q_1 \Rightarrow q_2) \wedge (q_2 \Rightarrow q_1)$	$\delta_{q_1} = \delta_{q_2}$
$\oplus$ *(Exclusive Or)*	$q_1 \oplus q_2$	$1 = \delta_{q_1} + \delta_{q_2}$

The linear inequalities are called IP-formulas.

2.3 Predicate Calculus and Modeling Language

For many years the lack of a concise modeling language and computing environment capable of computing in a reasonable time, complex decisions problems had hindered the use integer programming by non specialist. The goal of this section is to introduce a new declarative language as proposed in [1]. This modeling language with its extension represents the foundation of our hybrid system modeling approach. In the previous section we have shown how to manually transform a logic specification into inequalities for simple Boolean expressions. We will introduce a Mathematica package to automatically transform complex logic decisions problems to linear inequality constraints called IP-formulas that are solvable by any IP solvers.

Predicate calculus, first proposed by Mckinnon & Williams offers a more flexible modeling synthax than the propositional calculus of Hadjiconstantinou & Mitra or Raman

& Grossman. In predicate calculus one can declare predicates n-ary relations between objects. The achievement of Qiang Li et al [1] is to extend the Mckinnon & Williams' method [2] by first adding new predicates called meta-predicates. Their modeling language provides more rigorous definition of modeling language and systematic transformation, it is called L^+.

The L^+ modeling language is an extended language based on the first order predicate logic with no quantifiers.

2.3.1 Language L^+

Syntax:

1. Improper symbols: $(,), [,], \{,\}$

2. Integer: *,-2, 0, +1, +2,..*

3. Logical connective: $\wedge, \vee, \Rightarrow, \Leftrightarrow, \neg$

4. Function symbols: $+, -, \times$

5. Predicate symbols: $=, \leq, \geq, <, >$

6. Proposition variables: *p, q, r*

7. Formula: $P(x, y)$

Semantics:

The semantics of the L^+ language defines the formation rule for well-form formulas as follows:

1. A proposition variable is an atomic formula

2. An atomic formula is a formula

3. If P and Q are formulas and x is a variable then $\neg P, P \vee Q, P \wedge Q, P \Rightarrow Q, P \Leftrightarrow Q$ are formulas

2.3.2 Meta Predicates:

Let $m \in Z^+$ and If S is a set of literals we consider two meta-predicates

1. $atleast(m,S)$ also called "*less than or equal*" predicate.

The meta-predicate $atleast(m,S)$ with argument (m,S) expresses the fact that at least m literal in S must be true.

2. $atmost(m,S)$ also called "*greater than or equal*" predicate.

The meta-predicate $atmost(m,S)$ with argument (m,S) expresses the fact that at most m literal in S must be true.

These meta-predicates represent a compact logic specifications or declarations. Their advantages are that they avoid us from formulating lengthy Boolean or algebraic expressions for describing complex decisions systems.

Let $m = 1,\ S = \{s_1,\ s_2,\ s_3\}$

$$atleast(1,\{s_1,s_2,s_3\}) \longleftrightarrow 1 \le \delta_{s_1} + \delta_{s_2} + \delta_{s_3}$$

If 1 element of $\{s_1,\ s_2,\ s_3\}$ is true, we have: $1 \le 1$.

If 2 element of $\{s_1,\ s_2,\ s_3\}$ is true, we have: $1 \le 2$.

If 3 element of $\{s_1,\ s_2,\ s_3\}$ is true, we have: $1 \le 3$.

The second meta-predicate *atmost* with argument *(m, S)* expresses the fact that at most *m* formulas in *S* must be true.

Let $m = 1,\ S = \{s_1,\ s_2,\ s_3\}$

$$atmost(1,\{s_1,s_2,s_3\}) \longleftrightarrow \delta_{s_1} + \delta_{s_2} + \delta_{s_3} \le 1$$

If 1 element of $\{s_1,\ s_2,\ s_3\}$ is true, we have: $1 \le 1$.

If none of element of *{s₁, s₂, s₃}* is true, we have: $0 \le 1$.

Notation:

1. $atmost[1,\{s_i,\neg s_i\}] = s_i\ ,\quad i \in Z^+$

2. $none(\{s_1,s_2,s_3\}) = \neg s_1 \wedge \neg s_2 \wedge \neg s_3$

Proposition: Let $atleast(m,\{s_1,s_2,...,s_n\})$ and $atmost(m,\{s_1,s_2,...,s_n\})$ be two predicates such that $n,m \in Z^+\ and\, m \le n,\ m > 1\ m \le n\ ,\ and\ |S| = n \ge m$, then

$$exactly(m,\{s_1,s_2,..,s_n\}) = atleast(m,\{s_1,s_2,..,s_n\}) \cap atmost(m,\{s_1,s_2,..,s_n\})$$

And

$$exactly(m,\{s_1,s_2,...,s_n\})\longleftrightarrow\delta_{s_1}+\delta_{s_2}+...+\delta_{s_n}=m$$

Example:

$$exactly(1,\{s_1,\ s_2,\ s_3\})=atleast(1,\{s_1,\ s_2,\ s_3\})\bigcap atmost(1,\{s_1,\ s_2,\ s_3\})$$

$$exactly(1,\{s_1,\ s_2,\ s_3\})\longleftrightarrow 1\le\delta_{s_1}+\delta_{s_2}+\delta_{s_3}\bigcap\delta_{s_1}+\delta_{s_2}+\delta_{s_3}\le 1$$

And

$$exactly(1,\{s_1,s_2,s_3\})\longleftrightarrow\delta_{s_1}+\delta_{s_2}+\delta_{s_3}=1\,.$$

For a reason that will apparent in subsequent sections or chapters the use of $exactly(1,S)$ will be very important in modeling the mutual exclusive property of our hybrid system approach.

The predicate *exactly* with argument *(1, S)* expresses the fact that exactly one element in the set *S* must be true.

2.3.3 Special Order Set of type 1: *exactly (m , S)*

Special order set 1, SOS1, was introduced by Beale and Tomlin (1969), it is a modeling formalism often used in mathematical programming problems. Its main importance comes from the fact it can be used to model a special restriction of a set of variables, with such an equality constraint, it is not necessary to stipulate that the binary variables in the constraints are integrals [56]. This use of SOS1 constraint takes a special new meaning in a branch and bound algorithm for 0-1 integer programming.

Definition: Special Order Set 1 (SOS1)

An SOS1 is a set of variables (continuous or integer) within which exactly one variable must be non-zero[56].

Example:

If a set of integer variables $\{\delta_1,\ \delta_2,\ \delta_3,\ ......,\ \delta_n\}$ belongs to the SOS1 then its equality constraint can be written as $\delta_1 + \delta_2, + \delta_3, + ... + \delta_n = 1$.

Note: a SOS1 can be performed even for continuous variables.

2.3.4 Generalization of the XOr using via Exactly Predicate

The intrinsic property of the "exclusive or" is its ability to express the mutual exclusive property, its usefulness will be apparent in subsequent chapters where we transform a non-deterministic automaton to a deterministic one.

Note: The "exclusive or" is 2-ary operator.

Another way to express "the exclusive or" is given q_1 and q_2 two literals.

$$q_1 \otimes q_2 = exactly(1, \{q_1, q_2\})$$

From the truth table in Table 2-4 it is quite clear that for two literals or operands q_1 and q_2 operated upon by the "exclusive or " operator , to be true, it is necessary that exactly one of the two literals q_1 and q_2 ought to be true.

- The generalization involves the design of an n-ary deterministic operator or predicate based on existing non-deterministic predicate.

- $exactly(m, \{q_1, q_2, .., q_n\}),\ where\ m \leq n$ (2.4)

It is very convenient to implement the predicate exactly based on predicate *atleast* and *atmost.*

2.4 From Logic to IP Formulas using Mathematica Package GenIP

In this section simple examples will be given to illustrate the declarative modeling of the results. The main idea here is to learn how to use the Mathematica Package GenIP as a logic modeling specification tool and based on basic inequalities developed in Section 2.2 to have assess the correctness of the computation. Readers who are interested in the detail of the implementation could go to [1].

2.4.1 Synthesizing IP formulas using Mathematica

In this section simple examples will be given to illustrate the declarative modeling and commenting the results. Readers who are interested in more detail could see [1].

Example:

Given the following logic specification, $(p_1 \wedge p_2) \Rightarrow (p_3 \vee p_4)$

Let us obtain the inequality using the modeling language.

$$GenIP[(p_1 \wedge p_2) \Rightarrow (p_3 \vee p_4), \{p_1, p_2, p_3, p_4\}] = \{1 - \delta_{p_1} - \delta_{p_2} + \delta_{p_3} + \delta_{p_4},\ 0 \le \delta_{p_1} \le 1,$$
$$0 \le \delta_{p_2} \le 1,\ 0 \le \delta_{p_3} \le 1,\ 0 \le \delta_{p_4} \le 1\ \}$$

Note: the solution to this problem could have been obtained if we apply previous transformations and the disjunctive rule.

Definition:

IP-formulas (or IP-forms) are inequality constraints involve in the formulation of Integer programming problem.

Example

We buy nine types of stocks numbered by 1 to 9. If three or more types of stocks {1 to 5} are bought, or less than four types of stocks {3 to 6, 8, 9} are bought then at most two types of stocks {6 to 9} will be bought unless none of stocks {5 to 7} are bought [2].

Find the logic specification model and obtain the IP-formulas.

Answer:

1. Logic Specification

$$((atleast[3, \{s_1,\ s_2,\ s_3,\ s_4,\ s_5\}) \vee atmost[3, \{s_3,\ s_4,\ s_5,\ s_6,\ s_8,\ s_9\}) \wedge \neg none[\{s_5,\ s_6,\ s_7\}])$$
$$\Rightarrow atmost[2, \{s_6,\ s_7,\ s_8,\ s_9\}])$$

2. Logic to IP-formulas transformation

$$GenIP[((atleast[3,\{s_1,\ s_2,\ s_3,\ s_4,\ s_5\}) \vee atmost[3,\{s_3,\ s_4,\ s_5,\ s_6,\ s_8,\ s_9\}) \wedge \neg none[\{s_5,\ s_6,\ s_7\}]) \Rightarrow atmost[2,\{s_6,\ s_7,\ s_8,\ s_9\}]),\ \{s_1,\ s_2,\ s_3,\ s_4,\ s_5,\ s_6,\ s_7,\ s_8,\ s_9\}]$$

$1:\ -1 + d_1 + d_2 + d_3 \geq 0$

$2:\ 1 - d_2 + \delta_{s_5} \geq 0$

$3:\ 5 - 3d_3 + \delta_{s_1} + \delta_{s_2} - \delta_{s_3} - \delta_{s_4} - \delta_{s_5} \geq 0$

$4:\ 1 - d_2 - \delta_{s_6} \geq 0$

$5:\ 1 - d_2 - \delta_{s_7} \geq 0$

$6:\ 4 - 2d_1 - \delta_{s_6} - \delta_{s_7} - \delta_{s_8} - \delta_{s_9} \geq 0$

$7:\ -2d_3 - \delta_{s_3} - \delta_{s_4} - \delta_{s_5} - \delta_{s_6} - \delta_{s_8} - \delta_{s_9} \geq 0$

$8:\ 0 \leq d_1 \leq 1,\ 0 \leq d_2 \leq 1,\ 0 \leq d_3 \leq 1,$

$9:\ 0 \leq \delta_{s_1} \leq 1,\ 0 \leq \delta_{s_2} \leq 1,\ 0 \leq \delta_{s_3} \leq 1,$

$10:\ 0 \leq \delta_{s_4} \leq 1,\ 0 \leq \delta_{s_5} \leq 1,\ 0 \leq \delta_{s_6} \leq 1,$

$11:\ 0 \leq \delta_{s_7} \leq 1,\ 0 \leq \delta_{s_8} \leq 1,\ 0 \leq \delta_{s_9} \leq 1$

Note that all d_i, for $i = 1,..9$ are auxiliary binary variables which are introduced during the computation, dependency between inequalities are expressed by auxiliary binary variables relating them.

The inequalities in line 2, 4, 5 can be grouped together. They represent the IP-formulas of the predicate $\neg none[\{s_5,\ s_6,\ s_7\}]$.

The inequalities in line 3 and 6 represent respectively the IP-formula of the predicate

$atleast[3,\{s_1,\ s_2,\ s_3,\ s_4,\ s_5\}]$ and $atmost[3,\{s_3,\ s_4,\ s_5,\ s_6,\ s_8,\ s_9\}]$

The inequality in line 6 represents the IP-formulas of the predicate

$atmost[2,\{s_6,\ s_7,\ s_8,\ s_9\}]$

The inequalities in line 8 and 9 ,10, 11 represent the range of the binary variables and auxiliary binary variables, respectively.

Corollary: Let $atleast(n,\ \{s_1,\ s_2,...,s_m\})$ and $atmost(n,\ \{s_1,\ s_2,...,s_m\})$ be two predicates such that $n,\ m \in Z^+$ and $n \le m$ then

$$GenIP[exactly(n,\{s_1,\ s_2,..,s_m\})] =$$
$$GenIP[atleast(n,\{s_1,\ s_2,..,s_m\})] \bigcap GenIP[atmost(n,\{s_1,\ s_2,..,s_m\})]$$

Example:

$$GenIP[atmost(1,\{s_1,\ s_2,\ s_3\})] = \{1-\delta_{s_1}-\delta_{s_2}-\delta_{s_3} \ge 0,\ 0 \le \delta_{s_1} \le 1,\ 0 \le \delta_{s_2} \le 1,\ 0 \le \delta_{s_3} \le 1\ \}$$

$$GenIP[atleast(1,\{s_1,\ s_2,\ s_3\})] = \{-1+\delta_{s_1}+\delta_{s_2}+\delta_{s_3} \ge 0, 0 \le \delta_{s_1} \le 1,\ 0 \le \delta_{s_2} \le 1,\ 0 \le \delta_{s_3} \le 1\ \}$$

$$GenIP[exactly(1,\{s_1,\ s_2,\ s_3\})] = \{1-\delta_{s_1}-\delta_{s_2}-\delta_{s_3} \ge 0,\ -1+\delta_{s_1}+\delta_{s_2}+\delta_{s_3} \ge 0,$$
$$0 \le \delta_{s_1} \le 1,\ 0 \le \delta_{s_2} \le 1,\ 0 \le \delta_{s_3} \le 1\ \ \}$$

The last expression above is equivalent to:

$$GenIP[exactly(1,\{s_1,\ s_2,\ s_3\})] = \{1 = \delta_{s_1} + \delta_{s_2} + \delta_{s_3}, 0 \le \delta_{s_1} \le 1,\ 0 \le \delta_{s_2} \le 1,\ 0 \le \delta_{s_3} \le 1\ \}$$

2.5 Conclusion

Propositional logic and predicate are the cornerstones of this thesis, therefore precise definitions of logic proposition, predicate logic and their interpretations were given. The main important aspect of this chapter is the conversion from logic specification to IP-formulas. A Boolean expression containing the connective and, or, exclusive or, imply exactly, at least and at most were discussed. A Mathematica method was proposed to automate the conversion process.

CHAPTER 3:HYBRID DYNAMICAL SYSTEMS

3.1 Introduction

Hybrid dynamical system is a special type of dynamical system where where discrete event interacts with continuous dynamic. This type of dynamical system is suited to complex technological systems. Therefore a great deal of effort is now being spent to the study of the design and control of hybrid dynamical system. In this chapter we will introduce the theory of discrete event dynamical system and present the concept of hybrid automata. We will show how to convert the transition of diagram of a hybrid automaton into logic specification. The SIMULINK/Stateflow graphical tool we will be used to represent a hybrid automaton.

3.2 Theory of Automata

A finite automaton also known as finite state machine (FSM) is a control circuit encountered in digital computers or electronic systems. A finite automaton receives information input sequences of symbols from some alphabet. At each time interval the machine “reads” one symbol of the incoming input sequence and responds by going into a new internal state: the response depends both on the symbol being red and on the machine’s present internal state.

Let *Q* denote the set of internal states; and *E* the set of symbols, we describes a particular machine by specifying a function $f : Q \times E \rightarrow Q$.

If q_i is the internal state and s_j is the symbol currently being red, then $f(q_i, s_j) = q_k$ is the machine's next state. The function *f* is called the next-state function or the transition function of the machine.

Let *M* be the machine whose alphabet, i.e. event set) is $E = \{0,\ 1\}$, whose internal states set is $Q = \{q_0, q_1\}$, and whose transition function is given by in

Table 3-1: Finite State Machine

Present State	*0*	*1*
q_0	q_0	q_1
q_1	q_1	q_0

Table 3-1 asserts that:

If the machine is in state q_0 and the reading is 0, then the machine remains in state q_0.

If the machine is in state q_0 and the reading is 1, then the machine goes to state q_1.

If the machine is in state q_1 and the reading is 0, then machine remains in state q_1.

If the machine is in state q_1 and the reading is 1, then the machine goes to state q_0.

The machine *M* can be described with the aid of a state diagram, which consists of circles interconnected by arcs.

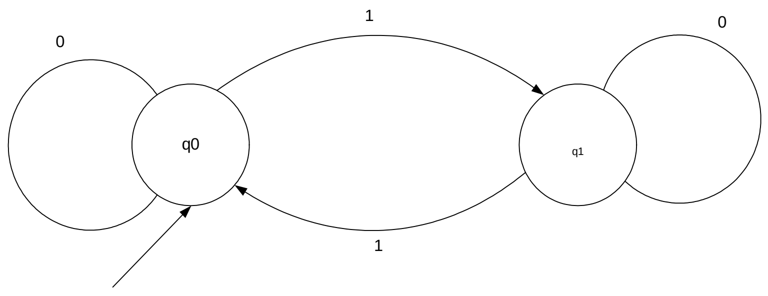


Figure 3.2-1:Finite State Machine

Note: The arrow indicates the initial state q_0 .

A finite state machine is a representation of an event-driven system. In event-driven system, the system transitions from one state to another state provided that the condition defining the change is true. For example, a state machine can be used to represent an automatic transmission or to check parity in a communication channel.

3.3 Discrete Event Systems

The discrete event system is modeled by a Deterministic Finite Automaton (DFA). The automaton is specified as a 6-tuple $S = \{Q, \Sigma, P, \Gamma, \delta, \phi\}$, where Q is a finite set of automaton states, Σ is a finite set of exogenous events (also called symbols), P is a finite set of events from the event generator, Γ is a finite set of output symbols,

$\delta : Q \times \Sigma \times P \to Q$ is the transition function, and $\phi : Q \to \Gamma$ is the output map. The action of the DFA is characterized by the equations

$$q(k) = \delta(q(k-1), \sigma(k), \rho(k)), \quad \gamma(k) = \phi(q(k))$$

Here k is a discrete time index that may be synchronous or asynchronous. In the asynchronous case the index advances when an event occurs. Ordinarily, a binary or Boolean labeling of the discrete states and events is used, i.e., $Q \subseteq \{0,1\}^{n_d}, \Sigma \subseteq \{0,1\}^{m_d}$, $\Gamma \subseteq \{0,1\}^{q_d}$.

Definition: Deterministic Finite Automaton

An automaton is composed of is a collection of the 6-tuple $S = \{Q, \Sigma, P, \Gamma, \delta, \phi\}$

Where:

Q :is a finite set of automaton states,

Σ :is a finite set of exogenous events

P :is a finite set of events from the event generator

Γ is a finite set of output symbols,

$\delta : Q \times \Sigma \times P \to Q$ is the transition function

$\phi : Q \to \Gamma$ is the output map.

3.4 Hybrid Automaton

3.4.1 Automata and Hybrid Systems:

The class of hybrid systems to be considered is defined as follows. The system operates in one of *m* states denoted $q_1, \ldots, q_m$. We refer to the set of states $Q = \{q_1, \ldots, q_m\}$ as the discrete state space. The discrete time dynamical equation describing operation in state q_i is $x_{k+1} = f_{q_i}(x_k, u_k),\ \ i=1,\ldots,m$ where $x \in X \subseteq R^n$ is the continuous state and $u \in U \subseteq R^n$ is the continuous control. Transitions can occur only between certain states. The set of admissible transitions is $E \subseteq Q \times Q$. It is convenient to view the state transition system as a graph with elements of *E* being the edges. We assume that transitions are instantaneous and take place at the beginning of the time interval. The switches that we consider, are controlled switches enabled by optimal controllers. So, if a transition systems from mode q_1 to q_2 at time *k* we would write $q(k) = q_1$ *and* $q(k^+) = q_2$. $q(k^+) = q^+(k) = q_2$ represents the successor of $q(k) = q_1$ and $q(k) = q_1$ represent the predecessor of $q(k^+) = q^+(k) = q_2$.We do not consider impulsive events. In other words, the continuous state trajectories are continuous through the event i.e., $x(k) = x(k^+)$.

Transitions are triggered by external events and guards. We denote the finite set of events Σ. It is convenient to partition the events into two types; those that are controllable, and those that are not. The latter are exogenous and occur spontaneously. Such an event might be specified by nature like a component failure, or a higher level operator such as a supervisory controller which decides changes in operational mode. We will use the symbols s to represent controllable events and p to represent uncontrollable events. Thus, $\sum = S \times P$ where $s \in S$ and $p \in P$.

A guard is a subset of the continuous state space X that enables a transition.

A transition enabled by a guard assignment function is $G : E \to 2^X$.

We consider each discrete state label, $q \in Q$, and each event, $\sigma \in \sum$, to be logical variables that take values True or False. Guards also are specified as logical conditions. In this way the transition system can be defined by a logical specification *L*.

3.4.2 Definition

A hybrid automaton is composed of a collection $HA = (Q, X, E, U, f, G, \Sigma)$, where:

- Q: Discrete state space
- X: Continuous states space
- E: set of transitions
- $G: E \longrightarrow 2^X$ guard assignment function
- U: Admissible input set
- Σ: Event set
- f: family of vector fields

3.4.3 Hybrid Automaton and Logical Specification

Consider the simple three modes hybrid system shown in Figure 3.4-1.

Each mode, q_1, q_2, q_3 is characterized by continuous dynamics

$$x_{k+1} = f_{q_i}(x_k, u_k),\ i = 1,2,3. \qquad (3.1)$$

Discrete transitions are associated with events represented by logical variables

p, s_1, s_2, s_3, i.e., $\Sigma = \{p, s_1, s_2, s_3\}$ .We use different symbols s and p to denote transition variables to underscore the fact that some transitions are controllable and others are not.

Notice that with the introduction of the Boolean variables we can replace the set of dynamical equations (3.1) with the single relation:

$$x(k+1) = f(x(k), \delta_q(k), u(k) = \delta_{q_1} f_{q_1}(x(k), u(k)) + \ldots + \delta_{q_m} f_{q_m}(x(k), u(k)) \qquad (3.2)$$

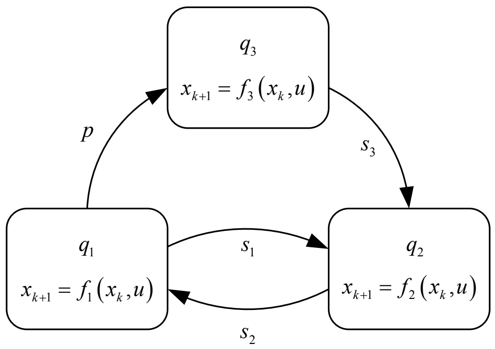


Figure 3.4-1: Three modes hybrid Automaton

$s_1, s_2 \in S$: external events set
$p \in P$: internal events set

In our formulation the transition system behavior is defined by the logical specification:

$$\begin{aligned} L = exactly(1,\{q_1(k), q_2(k), q_3(k)\}) \wedge exactly(1,\{q_1(k^+), q_2(k^+), q_3(k^+)\}) \wedge \\ (q_1(k) \wedge s_1 \Rightarrow q_2(k^+)) \wedge \ (q_1(k) \wedge p \Rightarrow q_3(k^+)) \wedge \ (q_1(k) \wedge \neg(s_1 \vee p) \Rightarrow q_1(k^+)) \wedge \\ (q_2(k) \wedge s_2 \Rightarrow q_1(k^+)) \wedge \ (q_2(k) \wedge \neg s_2 \Rightarrow q_2(k^+)) \wedge \\ (q_3(k) \wedge s_3 \Rightarrow q_2(k^+) \wedge \ (q_3(k) \wedge \neg s_3 \Rightarrow q_3(k^+)) \end{aligned} \tag{3.3}$$

The first and the second lines of (3.3) express the fact that the system can only be in one discrete state before the transition at time *k* and after the transition at time k^+. The next two lines describes all possible transitions from state q_1. Similarly, the last two lines characterize all possible transitions from states q_2 and q_3, respectively.

For computational purpose it is useful to associate with each logical variable, say *α*, a Boolean variable or indicator function, δ_α, such that δ_α assumes the values 1 or 0 corresponding respectively to *α* being True or False. It is convenient to define the discrete state vector $\delta_q = [\delta_{q_1}, \ldots.., \delta_{q_m}]$, the control event vector $\delta_s = [\delta_{s_1}, \ldots\ldots., \delta_{s_m}]$, and the exogenous event vector $\delta_p = [\delta_{p_1}, \ldots\ldots, \delta_{p_m}]$. Precisely one of the elements of δ_q will be unity and all others will be zero.

3.5 Hybrid Automaton and Matlab/SIMULINK

3.5.1 Overview

Unlike in MLD where a new conceptual framework (Hybrid Discrete Automaton) was developed, in this thesis we will take advantage of preexisting tool such as SIMULINK /Stateflow. SIMULINK with Stateflow contains all the features needed for simulating hybrid systems. Stateflow depicts well automata and SIMULINK is targeted for simulation of continuous dynamics. Interface between SIMULINK and Stateflow, makes the pair SIMULINK/Stateflow a viable modeling framework and simulation tool for hybrid systems.

We consider a hybrid automaton as the primitive model of a hybrid system. This point of view is natural because the hybrid automaton (HA) is an intuitive and efficient way to characterize a hybrid system, it has a convenient and appealing graphical representation, and SIMULINK with Stateflow is a powerful computer tools for building and simulating hybrid systems. Because of those factors the hybrid automaton is a good conceptual tool for formulating and simulating hybrid systems. We will show that SIMULINK /Stateflow is an effective approach for hybrid simulation. One of the goals of this project is to develop a computer tool that converts a hybrid automaton model to an equivalent model similar to the MLD model. Our MLD–like model joins the ordinary differential equations that describe the system dynamics with a set of mixed continuous and integer inequalities which represent the discrete event or logical specification of the system.

3.5.2 Stateflow

Stateflow is a powerful graphical and development tool for complex control and supervisory logic problems. Stateflow is a variant of the finite state machine notation established by Harel.

Some Advantages

1. Create a Stateflow diagram

2. Visualize a model and simulate complex hybrid systems based on finite state machine theory, flow diagram and state-transitions diagrams.

3. Design and development deterministic supervisory control systems.

4. Modification of design, evaluate results and verify the system's behavior at any stage of the design.

5. Take advantage of the integration with the Matlab and SIMULINK environments to model, simulate, and analyze your systems.

Examples of Applications.

1. Embedded system

2. Avionics

3. Automotive: transmission systems, cruise control

4. Telecommunication

5. Hybrid Systems

6. Air Traffic controls systems

StateFlow representation of a Power conditioning's automaton

- Power Conditioning System

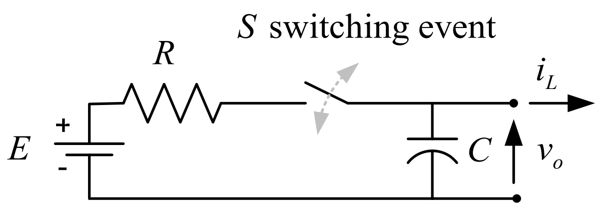


Figure 3.5-1:Power Conditioning Device

Two modes q_1, q_2 and two switches states $s, \neg s$ are needed to represent the transition diagram of the above Power conditioning.

- Stateflow Representation

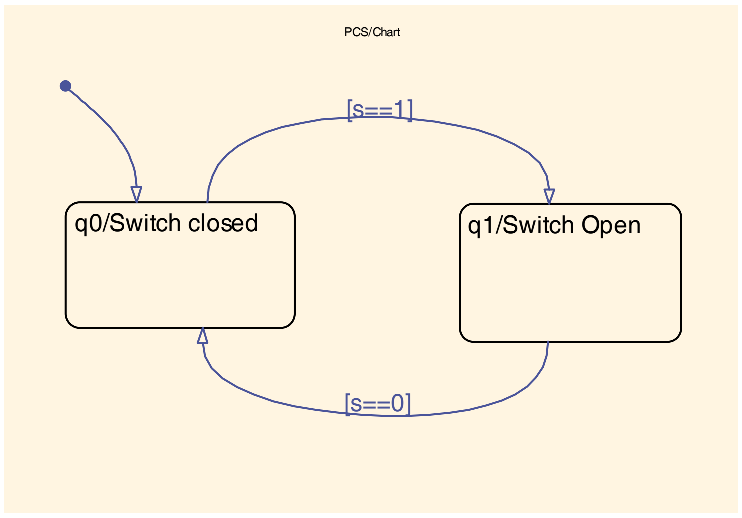


Figure 3.5-2: Power Conditioning in Stateflow

SIMULINK/Stateflow Integration:

A hybrid system is composed of a continuous dynamical system interfaced with a discrete event system. Figure 3.5-3 shows a general hybrid system framework in SIMULINK

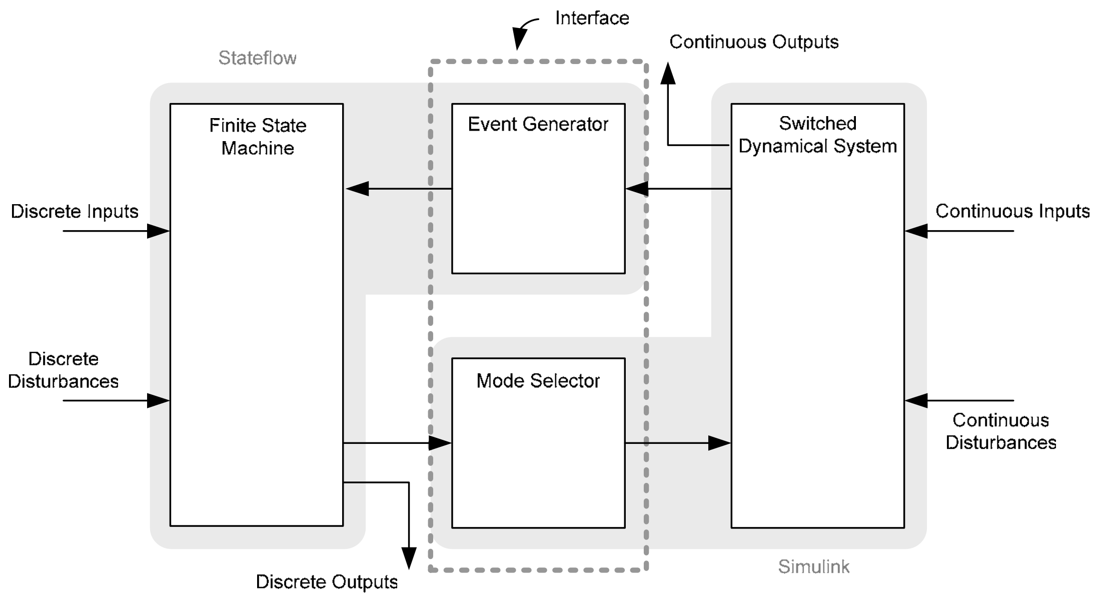


Figure 3.5-3: A hybrid system depicting of a switched dynamical system

In the basic setup, as is often used in studies of supervisory control, the continuous system is represented by a set of nonlinear ordinary differential equations.

$$\dot{x} = f(x,u,v),\ y = h(x,u,v)$$
$$x \in \mathscr{X} \subset R^n, u \in \mathscr{U} \subset R^m, v \in \mathscr{V} \subseteq \{0,1\}^{m_M}, y \in \mathscr{Y} \subset R^q \tag{3.4}$$

where x is the continuous system state, u is a vector of continuous inputs, v is a vector of discrete inputs (from the mode selector). This system is quite general. For example, a switched affine system can be represented by $\dot{x} = \sum_{i=1}^{m_m} v_i(t)\{A_i x + B_i u + f_i\}$

where $v_i : R \to \{0,1\}$ and $\sum_{i=1}^{m} v_i(t) = 1$. A possible generalization is to replace the system of ordinary differential equations by a system of differential-algebraic equations – a necessary generalization for dealing with electric power systems.

3.6 Logic Constraint

A logic constraint is the general term given to a constraint in Logic-Based preprocessing in order to reduce the processing time. Different approaches can be used to generate logic constraints. Resolution generates clauses that are prime implicant. The technique of resolution is analogous to the Chevatal's cutting plane.

We call it Control Logic a logic constraint whose setting reduces the number of feasible states.

We write the Control Logic specification as:

$$Control\ Logic =$$

$$q_1 \Rightarrow atmost[1,\{s_1,p\}] \wedge \neg s_2 \wedge \neg s_3 \wedge exactly[1,\{q_1,q_2,q_3\}] \wedge exactly[1,\{qq_1,qq_2,qq_3\}]$$

$$q_2 \Rightarrow s_2 \wedge \neg s_1 \wedge \neg s_3 \wedge exactly[1,\{q_2,q_1\}] \wedge \neg q_3 \wedge exactly[1,\{qq_2,qq_1\}] \wedge \neg qq_3$$

$$q_3 \Rightarrow s_3 \wedge \neg s_1 \wedge \neg s_2 \wedge exactly[1,\{q_3,q_2\}] \wedge \neg q_1 \wedge exactly[1,\{qq_3,qq_2\}] \wedge \neg qq_1$$

(3.5)

Note: $atmost[1,\{s_i,\neg s_i\}] = s_i$

The effect of the ControlLogic constraint to reduce search space is efficient. The computational time has been improved by a factor of 10.

3.7 Conclusion

Hybrid system was introduced using the theory of hybrid automaton which represents an excellent modeling framework. The definitions of deterministic finite automaton and hybrid automaton were given and an interpretation of an automaton as a logical specification was proposed. A three mode automaton was presented for illustration. SIMULINK/Stateflow was selected as a design frame work for hybrid system and a power conditioning example was presented. A discussion on logical constraint for improving the computational speed was proposed.

CHAPTER 4:OPTIMAL CONTROL OF HYBRID SYSTEMS

4.1 Introduction

The theory of optimal control had provided an impetus for development of space missions in the 1960s, during that period the concept of hybrid system was introduced.

The first article which referred to the hybrid system is [16]. Since then numerous studies had been conducted to reflect the ideas developed in [16], it is true that many modern complex technological systems are hybrid systems in nature and a form of optimal control such as dynamic programming are now receiving considerable attention.

4.2 Dynamic Programming

Dynamic programming was invented in 1952 by Richard Bellman, it is a powerful tool for solving dynamic optimization problem. It is based on the 'principle of optimality' which provides a mechanism for backward recursive solution as a single stage optimization problem. It can easily be applied to nonlinear continuous systems or hybrid systems, and the more constraints there are, the easier is the solution. The Bellman principle provides a good characterization of optimality for general dynamical systems including hybrid systems.

Computational complexity is often a big concern in dynamic programming; however various techniques are being proposed to reduce the computational time, in [15] a relaxed dynamic programming is studied. It consists of a parameterized sub-optimal solution of the value function by an upper and lower bound and the size of the bounds determines the computational complexity. An application to a DC-DC voltage converter modeled as switched voltage controller is given.

In this chapter, we will present the theory of dynamic programming for hybrid system by stating the Bellman's principle of optimality. We will propose a mixed-integer dynamic programming algorithm for hybrid system.

4.3 Bellman's Principle of Optimality

Let us Consider the state space X and the control space U are discretized into a finite grid.

The discretized plant dynamic is:

$$x_{k+1} = f\left(x_k, u_k\right),\ k = 0,1,\ldots,N-1$$

And let is define the cost function $J_i(x_i) = g_N + \sum_{k=1}^{N-1} L(x_i, u_i)$ where $[i,\ N]$ is the time interval of interest.

Suppose a control policy $u_{k+1}^*,\ u_{k+1}^*, \ldots, u_{N-2}^*, u_{N-1}^*$ at state x_{k+1}. By the principle of optimality, whatever the initial state x_k and the decision u_k are, the remaining decisions also must constitute an optimal policy with regard to the state x_{k+1} that is

$$J_k^*(x_k) = \min_{u_k}\{L^k(x_k, u_k) + J_{k+1}^*(x_{k+1})\} \tag{4.1}$$

The cost function $J_k(x_k)$ can be minimize only with respect to the control u_k provided the future cost $J_{k+1}^*(x_{k+1})$ is optimal, where $J_k^*(x(k))$ is the optimal cost-to-go function or the optimal value function at state x_k starting from stage from stage *k*.

The backward recursive procedure in (4.1) provides the solution to the minimization problem.

4.4 Bellman's Principle of Optimality for hybrid systems

Our approach to control design is based on finite, (receding horizon) dynamic programming. Dynamic programming leads to a feedback strategy where the computation is repeated every Δt sec. The feedback policy is computed off-line and implemented in a form such as table look-up.

We briefly summarize a form of dynamic programming needed to solve control problems involving hybrid systems. We will consider a discrete time, deterministic system that evolves over a finite time period. This period is divided into N equally spaced intervals and *k* is the discrete time index. All events, controllable or exogenous, are assumed to occur at the beginning of the interval, so we distinguish between values of variables at instant *k*, before any event, k^+ after the event, e.g., x_k, x_{k^+}.

$$x_{k+1} = f\left(x_{k^+}, \delta_{q,k^+}, u_k, \delta_{s,k}\right),$$

$$y_k = h(x_k, \delta_{q,k}), \qquad k = 0,1,\ldots,N-1 \qquad (4.1)$$

$$E_6\delta_{q^+} + E_7 d \le E_0 + E_1 x + E_2\delta_q + E_3 y + E_4\delta_s + E_5\delta_e$$

Where *k* is the discrete time index, and

x_k : the continuous state (real numbers)

$\delta_{s,k}$: the discrete state (mode) (binary or integer numbers)

u_k : the control, may be composed of discrete and continuous elements

d_k : discrete (binary) auxiliary variables

y_k : continuous (real) output or auxiliary variables

$\delta_{e,k}$: exogenous events.

A class of feedback control laws or policies consists of a sequence of functions $\pi = \left\{\mu_0\left(x_0\right), \mu_1\left(x_1\right), \ldots, \mu_{N-1}\left(x_{N-1}\right)\right\}$, so that $\{u_k, \delta_{s,k}\} = \mu_k\left(x_k, \delta_{q,k}\right)$. Given an initial state (x_0, δ_0), the problem is to find a control policy that minimizes the cost functional

$$J_\pi\left(x_0, \delta_0\right) = g_N\left(x_N, \delta_N\right) + \sum_{k=0}^{N-1} g_k\left(x_k, \delta_k, \mu_k\left(x_k, \delta_k\right)\right) \qquad (4.2)$$

The optimal cost function is

$$J^*\left(x_0, \delta_0\right) = \min_{\pi\in\Pi} J_\pi\left(x_0, \delta_0\right) \qquad (4.3)$$

And the optimal policy π^* is one that satisfies

$$J_{\pi^*}\left(x_0,\delta_0\right) \le J_\pi\left(x_0,\delta_0\right) \quad \forall \pi \in \Pi \qquad (4.4)$$

Principle of optimality: Suppose $\pi^* = \left\{\mu_1^*,\ldots,\mu_{N-1}^*\right\}$ is an optimal control policy. Then the sub-policy $\pi_i^* = \left\{\mu_i^*,\ldots,\mu_{N-1}^*\right\}$, $1 \le i \le N-1$ is optimal with respect the cost function

$$J_\pi\left(x_i,\delta_i\right) = g_N\left(x_N,\delta_N\right) + \sum_{k=i}^{N-1} g_k\left(x_k,\delta_k,\mu_k\left(x_k,\delta_k\right)\right) \qquad (4.5)$$

Let us denote the optimal cost of the trajectory beginning at x_i as $J_i^*\left(x_i,\delta_i\right)$. It follows from the principle of optimality that

$$\begin{aligned} J_{i-1}^*\left(x_{i-1},\delta_{i-1}\right) &= \min_{\pi\in\Pi} J_\pi\left(x_{i-1},\delta_{i-1}\right) = \min_{\pi\in\Pi}\left\{g_N\left(x_N,\delta_N\right) + \sum_{k=i-1}^{N-1} g_k\left(x_k,\mu_k\left(x_k,\delta_k\right)\right)\right\} \\ &= \min_{\mu_{i-1}}\left\{g_{i-1}\left(x_{i-1},\delta_{i-1},\mu_{i-1}\left(x_{i-1},\delta_{i-1}\right)\right) + J_i^*\left(x_i,\delta_i\right)\right\} \end{aligned} \qquad (4.6)$$

Equation (4.6) provides a mechanism for backward recursive solution of the optimization problem. To begin the backward recursion, we need to solve the single stage problem with $i = N$:

$$J_{N-1}^*\left(x_{N-1},\delta_{N-1}\right) = \min_{\pi\in\Pi} J_\pi\left(x_{N-1},\delta_{N-1}\right) = \min_{\mu_{N-1}}\left\{g_{N-1}\left(x_{N-1},\delta_{N-1},\mu_{N-1}\left(x_{N-1},\delta_{N-1}\right)\right) + J_N^*\left(x_N,\delta_{N-1}\right)\right\} \qquad (4.7)$$

There are two possible terminal conditions:

Case 1: x_N is fixed and $g\left(x_N\right) \equiv 0$. In this case $J_N^*\left(x_N\right) = 0$ and from Eq (4.1) we have the constraint

$$x_N = f\left(x_{N-1},\delta_{N-1},\mu_{N-1}\right) \qquad (4.8)$$

We assume that there exist solution pairs $\left(x_{N-1},\delta_{N-1},\mu_{N-1}\right)$ of Eq (4.8) . Otherwise, the problem is not well posed because x_N is not reachable. Thus,

$$J^*_{N-1}\left(x_{N-1},\delta_{N-1}\right) = \min_{\mu_{N-1}}\left\{g_{N-1}\left(x_{N-1},\delta_{N-1},\mu_{N-1}\right)\right\} \qquad (4.9)$$

Where the minimization is carried out subject to the constraint Eq (4.8).

Case 2: x_N is free. Now,

$$\begin{aligned} J^*_{N-1}\left(x_{N-1},\delta_{N-1}\right) &= \min_{\mu_{N-1}}\left\{g_{N-1}\left(x_{N-1},\delta_{N-1},\mu_{N-1}\right)+g_N\left(x_N,\delta_N\right)\right\} \\ &= \min_{\mu_{N-1}}\left\{g_{N-1}\left(x_{N-1},\delta_{N-1},\mu_{N-1}\right)+g_N\left(f\left(x_{N-1},\delta_{N-1},\mu_{N-1}\right)\right)\right\} \end{aligned} \qquad (4.10)$$

Once the pair $\mu_{N-1},J^*_{N-1}\left(x_{N-1},\delta_{N-1}\right)$ is obtained, we compute $\mu_{N-2},J^*_{N-2}\left(x_{N-2},\delta_{N-2}\right)$ from

$$\begin{aligned} J^*_{N-2}\left(x_{N-2}\right) &= \min_{\mu_{N-2}}\left\{g_{N-2}\left(x_{N-2},\mu_{N-2}\left(x_{N-2}\right)\right)+J^*_{N-1}\left(x_{N-1},\delta_{N-1}\right)\right\} \\ &= \min_{\mu_{N-2}}\left\{g_{N-2}\left(x_{N-2},\delta_{N-2},\mu_{N-2}\left(x_{N-2},\delta_{N-2}\right)\right)+J^*_{N-1}\left(f\left(x_{N-2},\delta_{N-2},\mu_{N-2}\right)\right)\right\} \end{aligned}$$

(4.11)

Continuing in this way

$$J^*_{N-i}\left(x_{N-i}\right) = \min_{\mu_{N-i}}\left\{g_{N-i}\left(x_{N-i},\mu_{N-i}\left(x_{N-i}\right)\right)+J^*_{N-i+1}\left(f\left(x_{N-i},\mu_{N-i}\right)\right)\right\}, \quad 2\le i\le N \; (4.12)$$

The procedure is illustrated in

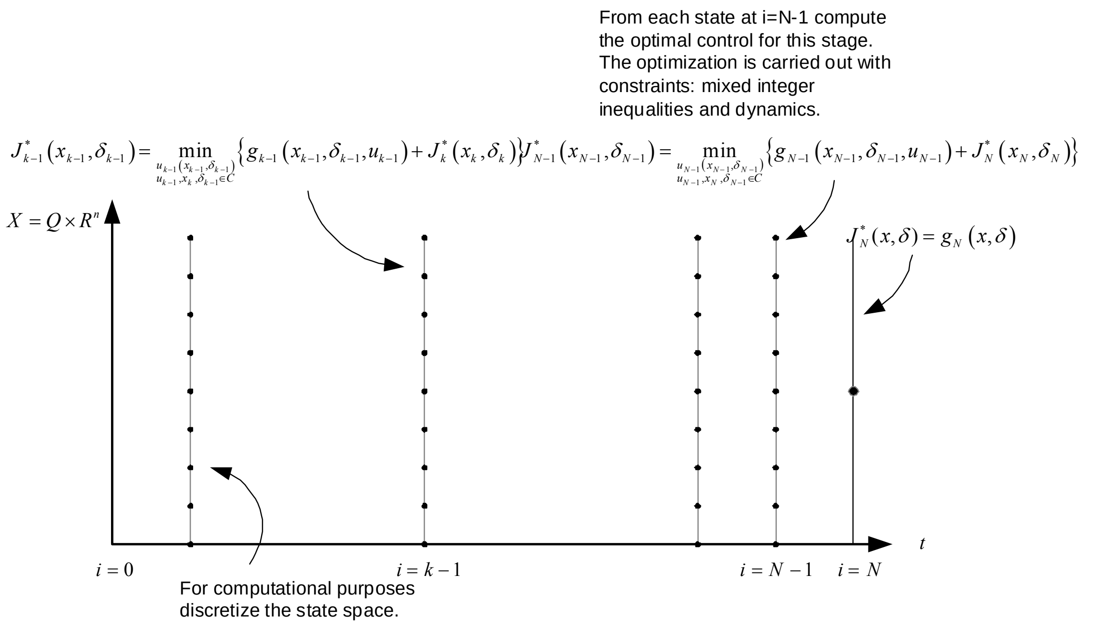

From each state at i=N-1 compute the optimal control for this stage. The optimization is carried out with constraints: mixed integer inequalities and dynamics.
$J^*_{k-1}(x_{k-1},\delta_{k-1}) = \min_{\substack{u_{k-1}(x_{k-1},\delta_{k-1})\\ u_{k-1},x_k,\delta_{k-1}\in C}} \{g_{k-1}(x_{k-1},\delta_{k-1},u_{k-1}) + J^*_k(x_k,\delta_k)\}$
$J^*_{N-1}(x_{N-1},\delta_{N-1}) = \min_{\substack{u_{N-1}(x_{N-1},\delta_{N-1})\\ u_{N-1},x_N,\delta_{N-1}\in C}} \{g_{N-1}(x_{N-1},\delta_{N-1},u_{N-1}) + J^*_N(x_N,\delta_N)\}$
$X = Q \times R^n$
$J^*_N(x,\delta) = g_N(x,\delta)$
t
$i = 0$
$i = k-1$
$i = N-1$
$i = N$
For computational purposes discretize the state space.

Figure 4.4-1: Depiction of dynamic programming algorithm.

The first step in solving the optimal control problem is to transform the logical constraints into a set of inequalities involving binary variables and possibly real variables, so-called *IP-formulas*. The idea of formulating optimization problems using logical constraints and then converting them to IP formulas has a long history. This concept was used as a means to incorporate qualitative information in process control [56], and generally introduced into the study of hybrid systems [29].

Authors in [1] suggested a sequence of transformations that brings a logical specification into a set of IP-formulas. A systematic algorithm is presented for doing this. We have modified this implementation in order to obtain simpler and more compact IP formulas.

The system operates in one of the m modes denoted $q_1,\ldots,q_m$. $Q=\{q_1,...,q_m\}$ is the discrete state space. The discrete time difference-algebraic equation (DAE) describing operation in mode q_i is

$$\begin{aligned} x_{k+1} &= f_i(x_k, y_k, u_k) \\ 0 &= g_i(x_k, y_k, u_k) \end{aligned} \tag{4.13}$$

with $i=1,...,m$

where $x \in X \subseteq \Re^n$ is the system continuous state, $y \in Y \subseteq \Re^p$ is the vector of algebraic variables and $u \in U \subseteq \Re^l$ is the continuous control. Transitions can occur only between certain modes. The set of admissible transitions is $\varepsilon \subseteq Q \times Q$. It is convenient to view the mode transition system as a graph with elements of ε being the edges. We assume that transitions are instantaneous and take place at the beginning of the time interval. So if the system transitions from mode q_1 to q_2 at time k we would write $q(k)=q_1$, $q(k^+)=q_2$. We allow resets. State trajectories are assumed continuous through events, i.e., $x(k)=x(k^+)$, unless a reset is specified.

Transitions are triggered by external *events* and *guards*. Events are of two types; either controlled – belonging to the set $\sum_s$, or exogenous (occur spontaneously) – belonging to the set $\sum_e$. A guard is a subset of the continuous state space X that enables a transition. A transition enabled by a guard might represent a protection device. Not all transitions might require simultaneous satisfaction of a guard and occurrence of an event.

We consider each discrete state label, $q \in Q$, and each event $s \in \sum_s$, $e \in \sum_e$ to be logical variables that take values True and False. Guards also are specified as logical conditions. In this way the transition system can be defined by a logical specification (formula).

For computational purposes it is useful to associate each logical variable, say α , a binary variable or indicator function, δ_α, such that δ_α assumes the values 1 and 0 corresponding respectively to α being True or False. It is convenient to define the discrete state vector $\delta_q = [\delta_1,...,\delta_m]$. Precisely one of the elements of δ_q will be unity and all others will be zero.

If all the guards are linear (set boundaries are composed of linear segments), then the IP formulas are system of linear constraints involving binary variables δ_q, δ_{q+}, δ_s, respectively, the discrete state before transition, the discrete state after transition, the exogenous events. They also involve a set of auxiliary binary variables, d, introduced during the transformation process, and the continuous state variables, x. With x, δ_q, δ_s given these inequalities provide a unique solution for the unknowns δ_{q+} and d.

In doing the optimal calculations we now exploit the fact that the system is highly constrained and all of the constraints are linear in the binary variables. The basic approach has been modified to the following:

4.5 Mixed-Integer Dynamic Programming Algorithm

1. Before the beginning of the time iteration:

   a. Separate the inequalities into binary and real sets, binary formulas contain only binary variables, real formulas can contain both binary and real variables

   b. For each $q \in Q$, obtain feasible solutions of the binary inequalities; a list of possible solution pairs (δ_{q+}, d)

   c. Define projection $\bar{X} \rightarrow \bar{X}_{\mathrm{P}}$ where $\bar{X}_{\mathrm{P}}$ is the subspace of real states actually appearing in the real equations.

   d. For each $x_P \in \bar{X}_{\mathrm{P}}$

      i. Pre-screen the binary solutions to eliminate those that do not produce solutions to the real inequalities – typically a very large fraction is dropped

      ii. For every feasible combination of binary variables obtained above, solve the real inequalities for the real variables.

   e. Lift real solutions to entire $\bar{X}$.

2. For each i

   a. For each pair $(q, x) \in Q \times \bar{X}$

      i. Enumerate the values of the cost to go using the feasible sets of binary and real variables

      ii. Select the minimum

In step 1b above the number of solutions corresponding to each q can be vary large because there are numerous redundant solutions associated with non active transitions. Thus, we add additional logical constraints that specify the inactive transitions. Step 1c exploits the fact that some real states do not appear in the real formulas. Because a large fraction of the binary solutions do not lead to real solutions, the pre-screening in step 1d.i is very effective in reducing the computation time. Finally, we note that the inequalities are independent of the state of the dynamic programming recursion. Thus, step 1d, which is by far the most intensive computational element of the optimization, is done only once before the recursion step 2a begins.

4.6 Conclusion

Dynamic programming is a powerful method for solving optimal control problems. In this chapter we have shown that it is still a viable method for solving hybrid control problems. One of the accomplishments of this thesis is to have device a special purpose dynamic program algorithm called mixed-integer dynamic programming for constraints based hybrid optimal control. A detail algorithm of the mixed-integer dynamic programming was presented.

CHAPTER 5:A 3-BUS POWER SYSTEM WITH UPS

5.1 Introduction

In this chapter, and the following the next two chapters, we are selecting a relatively simple power system to illustrate our new hybrid system formulation. The advantage of the simple power system is that it is has been studied extensively in literature with regard to it voltage stability [65]. Our goal in here is to design a survivable power system where voltage regulation and power supply to vital load are critical issues. Control mechanisms such as excitation system, load shedding and emergency backup will play a preponderant role in our analysis. The big challenge in this example is the addition of an uninterruptible power supply (UPS) for emergency backup.

The organization of the chapter is as follows: we will begin by a description of the power system, and highlight all the assumptions such as the type of load, the non-conventional excitation system, voltage regulation level etc. The equations for the power system will be derived, followed by the use of the method for elimination of variable on network equations and finally the hybrid model will be presented.

5.2 Power System Description

A relatively simple system that is known to exhibit interesting voltage stability characteristics is a single generator feeding an aggregated load composed of constant impedance loads and induction motors. The system has been used to study the effect of tap changing transformers and capacitor banks in voltage control, e.g., [52,53,54]. Consider the system shown in Figure 5.2-1. The system consists of a generator, a transmission line, an on-load tap changing transformer (OLTC) and an aggregated load. The generator is characterized by a 'constant voltage behind reactance' model. The generator internal bus voltage E is used to maintain the voltage at bus 2; so long as E remains within the limits imposed by the excitation current limits. The OLTC ordinarily moves in small discrete steps over a narrow range. The load is an aggregate composed of parallel induction motors and constant impedance loads. An induction motor can be characterized as impedance with slowly varying resistance; consequently, the aggregate load is represented by constant impedance – actually, slowly varying impedance, where the impedance depends on the aggregate induction motor slip. Figure 5.2-2 shows the reduced two bus network accounting for the ideal transformer.

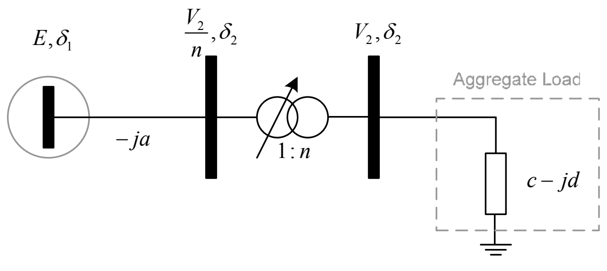


Figure 5.2-1: System configuration.

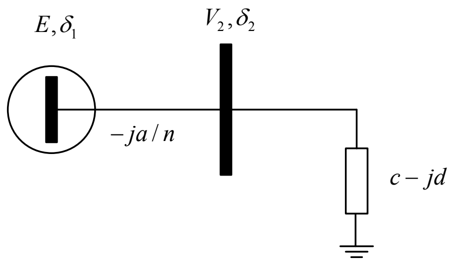


Figure 5.2-2: Equivalent circuit assuming ideal transformer.

The Network Equations:

Let us define the complex power injected at the bus *i* into the system as

$S_i = S_{Gi} - S_{Di}$ where S_{Gi} *and* S_{Di} are the bus power injections.

Using the conservation of power $S_i = \sum_{k=1}^{n} S_{ik}$, *i=1,2,....n*

We also define the current $I_i = I_{Gi} - I_{Di} = \sum_{k=1}^{n} I_{ik}$, *i=1,2,....n*

$$\text{or} \qquad I_i = \sum_{k=1}^{n} Y_{ik} V_k, \;\; i=1,2,....,n$$

The complex power S_i can be written as:

$$S_i = V_i I_i^* = V_i \sum_{k=1}^{n} Y_{ik}^* V_k^*, \; i=1,2,....n \qquad (5.1)$$

The power mismatch equations are given by the real and the reactive power as:

$$P_i = \sum_{k=1}^{n} |V_i||V_k|(g_{ik}\cos\theta_{ik} + b_{ik}\sin\theta_{ik})$$

$$Q_i = \sum_{k=1}^{n} |V_i||V_k|(g_{ik}\sin\theta_{ik} - b_{ik}\cos\theta_{ik}) \qquad (5.2)$$

Using (5.2) at bus 2 (*i=2*) and let voltage magnitude be $|V_1| = E,\ |V_2| = V_2$ and the phase angles difference $\theta_{21} = \theta_2 - \theta_1 = \theta_2$ then (5.2) reduces to:

$$\begin{aligned} P_{G2} - P_{D2} &= V_2 E(g_{21}\cos\theta_{21} + b_{21}\sin\theta_{21}) + V_2^2(g_{22}\cos\theta_{22} + b_{22}\sin\theta_{22}) \\ Q_{G2} - Q_{D2} &= V_2 E(g_{21}\sin\theta_{21} - b_{21}\cos\theta_{21}) + V_2^2(g_{22}\sin\theta_{22} - b_{22}\cos\theta_{22}) \end{aligned} \tag{5.3}$$

Since at load bus 2 , $P_{G2} = Q_{G2} = P_{D2} = Q_{D2} = 0$ and $g_{11} = g_{12} = g_{21} = g_{22} = 0$

(lossless transmission line) and $\cos\theta_{22} = 1,\ \mathit{sin}\ \theta_{22} = 0$ then (5.3) reduces to:

$$\begin{aligned} 0 &= V_2 E(b_{21}\sin\theta_{21}) + V_2^2(g_{22}) \\ 0 &= V_2 E(-b_{21}\cos\theta_{21}) + V_2^2(-b_{22}) \end{aligned} \tag{5.4}$$

Let the Y bus matrix of the network in Figure 5.2-2 be:

$$Y_{Bus} = \begin{bmatrix} g_{11} + ib_{11} & g_{12} + ib_{12} \\ g_{21} + ib_{21} & g_{22} + ib_{22} \end{bmatrix} = \begin{bmatrix} -i\dfrac{a}{n} & i\dfrac{a}{n} \\ i\dfrac{a}{n} & c - i\left(\dfrac{a}{n} + d\right) \end{bmatrix} \tag{5.5}$$

Then

$$\begin{aligned} 0 &= V_2 E\left(-\frac{a}{n}\right)\sin\theta_{21} + V_2^2(c) \\ 0 &= V_2 E\left(\frac{a}{n}\right)\cos\theta_{21} + V_2^2\left(\frac{a}{n} + d\right) \end{aligned} \tag{5.6}$$

or

$$\begin{aligned} 0 &= \left(\frac{a}{n}\right) E V_2 \sin\theta_2 - c V_2^2 \\ 0 &= \left(\frac{a}{n}\right) E V_2 \cos\theta_2 + \left(\frac{a}{n} + d\right) V_2^2 \end{aligned} \tag{5.7}$$

Power System Characteristic Curves:

Power system characteristic curves are expressions derived from the network or the load equilibrium equations. The characteristics curves that will be used in this bifurcation analysis are of the general form:

$$\begin{aligned} g_m(s, E, a, P_v) &= 0 \\ g_n\left(s, V_2, a, \eta, P_v\right) &= 0 \\ g_r\left(s, V_2, P_m\right) &= 0 \end{aligned} \tag{5.8}$$

The first two characteristic curves in (5.8) correspond to the excitation and the network characteristic curves. Both characteristic curves are obtained from the network equations. The third characteristic in (5.8) is the load characteristic curve, it is derived from the load equilibrium equations

The loss of a transmission line between two buses has the effect of decreasing the maximum power that can be delivered to the load. Since the maximum power that can be delivered is proportional to the admittance of the line, we can model transmission line faults by reducing the nominal parameter *a*. Our goal is to understand how load shedding can be effectively utilized to insure service of the vital load and to maximize the amount of non-vital load supplied. Considerable insight can be obtained by examining the graphs of the two scalar functions in the $V - s$ plane for fixed values of a, η, P_m, P_v.

5.3 Systems Modeling

5.3.1 System Modeling without and with UPS:

Power System without UPS

The network equations for the system shown in Figure 5.2-2 can easily be obtained. Suppose δ_1, δ_2 denote the voltage angles at bus 1 and, now let us define

$$I_1 \omega_0 \dot{\omega} = P_g - cV_2^2 \tag{5.9}$$

$$\begin{aligned} 0 &= \left(\frac{a}{n}\right) EV_2 \sin(\delta_2 - \delta_1) - cV_2^2 \\ 0 &= \left(\frac{a}{n}\right) EV_2 \cos(\delta_2 - \delta_1) + \left(\frac{a}{n} + d\right) V_2^2 \end{aligned} \tag{5.10}$$

Let $\theta_2 = \delta_2 - \delta_1$ from the last two equations we obtain:

$$V_2 = \frac{a/n}{\sqrt{c^2 + (a/n + d)^2}} E, \;\; \theta_2 = \tan^{-1} \frac{-c}{a/n + d} \tag{5.11}$$

Since the power absorbed by the load is $P_L + jQ_L = -|V_2|^2 Y$, we have:

$$P_L = -V_2^2 c, Q_L = V_2^2 \left(\frac{a}{n} + d\right) \tag{5.12}$$

Now, let us turn to the induction motor. An equivalent circuit for an induction motor is shown in Figure 5.3-1 below. Here, the parameters R_s, X_s denote the resistance and inductance of the stator, X_m denotes the magnetizing inductance, and R_r, X_r the rotor resistance and inductance. The resistance $R_r(1-s)/s$ represents the motor electrical output power. If we can neglect the small stator resistance and inductance, and if the approximation of large magnetizing inductance is acceptable, then the equivalent circuit reduces to that of Figure 5.3-2.

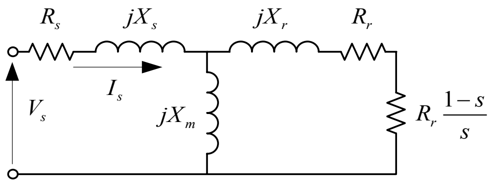


Figure 5.3-1: Induction motor equivalent circuit.

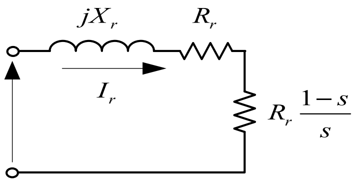


Figure 5.3-2: Simplified equivalent circuit assuming small X_s and L and large X_m.

If the simplified circuit of Figure 5.3-2 is acceptable, then we obtain the following. The real power delivered to the rotor, P_d, and the power delivered to the shaft, P_e, are:

$$P_d = V_s^2 \frac{R_r s}{R_r^2 + s^2 X_r^2}$$

$$P_e = P_d(1-s) \tag{5.13}$$

The dynamical equation for the motor (Newton's law) is:

$$\dot{\omega}_m = \frac{1}{I_m \omega_0}(P_e - P_m) \tag{5.14}$$

Introducing the slip, s,

$$s = \frac{\omega_0 - \omega_m}{\omega_0} \tag{5.15}$$

We obtain the motor dynamics in the following form:

$$\dot{s} = \frac{1}{I_m \omega_0^2}\left(P_m - P_e\right) = \frac{1}{I_m \omega_0^2}\left(P_m - V_s^2 \frac{R_r s\left(1-s\right)}{R_r^2 + s^2 X_r^2}\right) \tag{5.16}$$

Transition Structure and Load Shedding

In the following, we allow for the dropping of a fraction of the load, η. In the present case, we allow three different values of η including zero, so $\eta \in \{0, \eta_1, \eta_2\}$. Consequently, there is normal operation and two prioritized blocks of load that can be dropped in accordance with the transition behavior defined in Figure 5.3-3.

In the present case, we assume the blocks are sized such that

$$q_1 \Rightarrow \eta = 0, \quad q_2 \Rightarrow \eta = 0.4, \quad q_3 \Rightarrow \eta = 0.8$$

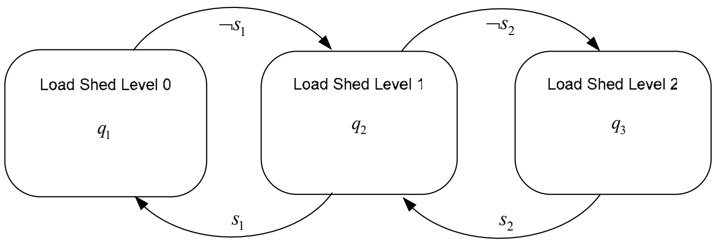


Figure 5.3-3: Transition structure.

We assume that the OLTC ratio is fixed, i.e., the OLTC is not being used for control, so $n = \text{const}$. If the OLTC is to be employed, the dynamics of tap change must be added.

$$I_1\omega_0\dot{\omega} = P_g - cV_2^2 \tag{5.17}$$

$$E = (1-\eta)\frac{\sqrt{c_0{}^2 + d_0{}^2}}{a/n}V_2 \tag{5.18}$$

$$\dot{s} = \frac{(1-\eta)}{I_m\omega_0^2}\left(P_m - V_2^2\frac{R_r s(1-s)}{R_r^2 + s^2X_r^2}\right) \tag{5.19}$$

$$c = (1-\eta)c_0, c_0 = \left(\frac{1}{R_L} + \frac{R_r s}{R_r^2 + s^2X_r^2}\right), \quad d = (1-\eta)d_0, d_0 = \left(\frac{X_r s^2}{R_r^2 + s^2X_r^2}\right) \tag{5.20}$$

Equation represents turbine-generator dynamics. Ordinarily, the power input P_g is adjusted to regulate the speed ω which is to be maintained at the value, ω_0. We assume that regulation is fast and accurate. It is possible to investigate the impact of frequency variation on system behavior. If it were assumed that frequency variations were small, then the effect on all impedances could be approximated, and this is often done. That has not been included here, so there is no apparent coupling between (5.17) and the remaining equations, so it can be dropped.

(5.18) represents the network voltage characteristic. The field voltage E is used to control the load bus voltage, V_2. We will assume that it is desired to maintain, $V_2 = 1$. If we ignore the exciter dynamics, then (5.18) allows the determination of the field voltage that yields the desired load bus voltage. However, the field voltage is strictly limited, $0 \le E \le 2$. If we assume that only the upper limit is a binding constraint, there are two possibilities for satisfying (5.18):

$$V_2 = 1, E = \frac{\sqrt{c^2 + (a/n + d)^2}}{a/n}$$
$$E = 2, V_2 = 2\frac{a/n}{\sqrt{c^2 + (a/n + d)^2}} \tag{5.21}$$

(5.19) represents the aggregated motor dynamics, and the load admittance is given by (5.20). $R_L = 2, R_r = 0.25, X_r = 0.125, a = 1\left(\text{nominal}\right), I_m\omega_0^2 = 4$

Power System with UPS

The load with induction motor as considered above is now expanded to include a UPS for the vital load. This system is shown in Figure 5.3-4.

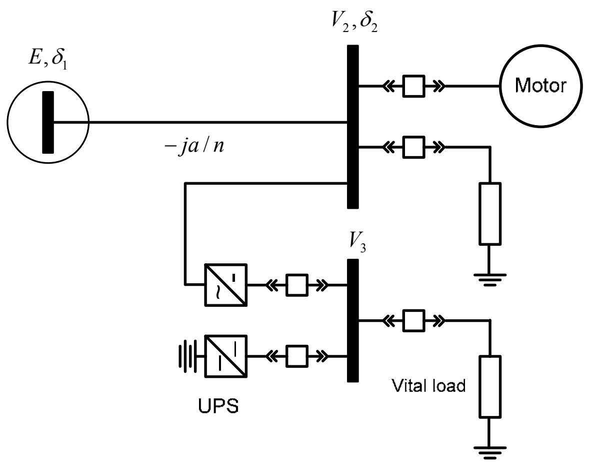


Figure 5.3-4: System with induction motor and UPS.

UNINTERRUPTIBLE POWER SUPPLY

In case of critical applications where a shutdown is unacceptable, a backup to the main power supply is provided by means of uninterruptible power supplies (UPS) [66]. An

UPS provides protection against power outages as well as voltage regulation during power line disturbances. In normal mode an AC-DC converter supplies power to the vital load. In case of emergency a DC-DC converter supplies power from the battery to DC vital load. The detail presentation of the functionality of the UPS is described in subsection 5.3.3.

The Network Equations:

The network equations of the system with UPS is similar to the network without UPS except that in (5.3) $P_{G2} = 0,\ Q_{G2} = 0,\ P_{D2} \neq 0,\ Q_{D2} \neq 0$

Let's redefine $P_v = P_{D2},\ Q_v = Q_{D2}$

Let $-S_{D2} = -P_v - jQ_v$ be the complex power consumed by the vital load at bus 2, where P_v and Q_v are the real power and the reactive power. S_2 is power factor corrected such that $Q_v = 0$

$$\begin{aligned} -P_v &= V_2E(\left(-\frac{a}{n}\right)\sin\theta_{21}) + V_2^2\left(c\right) \\ 0 &= V_2E(\left(\frac{a}{n}\right)\cos\theta_{21}) + V_2^2\left(\frac{a}{n} + d\right) \end{aligned} \tag{5.22}$$

Rearranging (5.22) we obtain the network equations:

$$\begin{aligned} P_v &= \left(\frac{a}{n}\right)EV_2\sin\theta_2 - cV_2^2 \\ 0 &= \left(\frac{a}{n}\right)EV_2\cos\theta_2 + \left(\frac{a}{n} + d\right)V_2^2 \end{aligned} \tag{5.23}$$

Where *c = (1- η)c₀ , d = (1- η)d₀* ,

And $P_v = V_3^2 R_v^{-1}$ is the power consumed by the vital load.

With,

$$c = (1-\eta)c_0, c_0 = \left(\frac{1}{R_L} + \frac{R_r s}{R_r^2 + s^2 X_r^2}\right), \quad d = (1-\eta)d_0, d_0 = \left(\frac{X_r s^2}{R_r^2 + s^2 X_r^2}\right)$$

5.3.2 Elimination Method for Unsaturated and Saturated Excitation Control

Often in simple power systems bifurcation curves can be obtained by series of transformations and eliminations in the network equations at a node of the network. The expression obtained can be graphed to produce the so-called Power-Voltage (PV curves) or nose curves. The procedures are simple and are described in many power systems books such as in [60],[64]. In some problems of low complexity elimination method or quantifier elimination method can be employed. Quantifier elimination [57] is a formal logic procedure that has been found to be an attractive method in the field of formal verification. In emergency situation such as line fault, control actions are needed to maintain a voltage profile or power supply to a vital load. We identify three such controls. The first control is the excitation control. The excitation control is relevant due to fact that when a line shortage occurs on multiple transmission lines, voltage at the receiving end decreases, excitation control can therefore be utilized to increase the excitation voltage in order to maintain the nominal post-fault voltage level. The second control is load shedding, which is a widely used mechanism for voltage collapse prevention and regulation. It is enabled whenever the excitation control saturates and it consists of reducing some fractions of the aggregate load admittance. The third control is used as an enabling of the uninterruptible power supply when the first two controls failed to maintain voltage level to the vital load bus.

Excitation System and Elimination Method:

The two possibilities are the unsaturated excitation voltage and the saturated excitation voltage.

- When the excitation voltage is unsaturated, the voltage is limited to the range $0 < E < 2$ and the regulated voltage is $V_2 = 1$. The unsaturated mode can be described by the set of constraints: $(V_2 = 1 \wedge 0 < E < 2)$
- When the excitation voltage saturates, its voltage is fixed to $E=2$, but the load bus voltage V_2 may no longer be regulated to reference voltage $V_{ref}=1$, therefore the value of the load bus voltage needs to be determined from the network equations.

Elimination method is a straight forward powerful method to eliminate generator phase angle in power system network equation. In the remaining part of this section the elimination method is performed on both the unsaturated excitation system and the saturated excitation system.

The network equation at bus 2 is as defined in (5.23)

The phase angle θ_2 can eliminated by using trigonometric identity $Cos^2\theta_2+Sin^2\theta_2=1$.

$$P_v + cV_2^2 = \left(\frac{a}{n}\right) EV_2 \sin\theta_2$$

$$\left(\frac{a}{n} + d\right) V_2^2 = \left(\frac{a}{n}\right) EV_2 \cos\theta_2 \tag{5.24}$$

Squaring and adding the two equations above yields:

$$(P_v + cV_2^2)^2 + \left(\frac{a}{n} + d\right)^2 V_2^4 = \frac{a^2}{n^2} E^2 V_2^2 (\sin^2\theta_2 + \cos^2\theta_2) \tag{5.25}$$

By expanding the above expression and using the trigonometric identity $\sin^2\theta_2 + \cos^2\theta_2 = 1$ we obtain the quartic equation in *V_2*.

$$P_v^2 + 2cV_2^2 P_v + c^2 V_2^4 + \left(\frac{a}{n} + d\right)^2 V_2^4 = \frac{a^2}{n^2} E^2 V_2^2 \tag{5.26}$$

or

$$(c^2 + \left(\frac{a}{n} + d\right)^2) V_2^4 + (2cP_v - \frac{a^2}{n^2} E^2) V_2^2 + P_v^2 = 0 \tag{5.27}$$

Equation (5.27) will be used in both the unsaturated and saturated excitation to derive the characteristic curves.

Elimination Method for Unsaturated Excitation Voltage

Let us obtain from (5.27) the excitation voltage when the load voltage is regulated (*V_2= 1* and assuming *E>0) as:*

$$P_v^2 + 2cP_v + c^2 + \left(\frac{a}{n} + d\right)^2 = \frac{a^2}{n^2}E^2 \tag{5.28}$$

Solving for E for $n=1$ we have:

$$E = \frac{\sqrt{(a+d)^2 + (c+P_v)^2}}{a} \tag{5.29}$$

Equation (5.29) is the excitation voltage necessary to maintain the regulation voltage to 1.

Elimination Method for Saturated Excitation Control

By setting $X = V_2^2$ and $E=2$ in (5.27), the following quadratic equation is obtained:

$$(c^2 + \left(\frac{a}{n} + d\right)^2)X^2 + (2cP_v - \frac{4a^2}{n^2})X + P_v^2 = 0 \tag{5.30}$$

Using the quadratic formula and simplifications we have:

$$X = \frac{-(2cP_v\frac{n^2}{n^2} - \frac{4a^2}{n^2}) \pm \frac{1}{n^2}\sqrt{-16cP_v a^2 n^2 + 16a^4 - n^4\left(\frac{a}{n} + d\right)^2 P_v^2}}{2(c^2 + \left(\frac{a}{n} + d\right)^2)} \tag{5.31}$$

Setting $n=1$ in V_2 we have:

$$V_2 = \sqrt{\frac{(2a^2 - cP_v) \pm \sqrt{4a^4 - 4cP_v a^2 - (a+d)^2 P_v^2}}{c^2 + (a+d)^2}} \tag{5.32}$$

The Double Root of a Quadratic Equation (Bifurcation point)

The double root as we will later see is the bifurcation point. It is the point where both solution of V_2 coincide. The value of the double root is obtained by setting the discriminate of quadratic equation (5.30) to zero.

$$4a^4 - 4cP_v a^2 - (a+d)^2 P_v^2 = 0 \tag{5.33}$$

Solving the quadratic equation (5.33) for the positive solution P_v at which both upper and lower part of V_2 coincide is:

$$P_v = \frac{2a^2(\sqrt{c^2 + (a+d)^2} - c)}{(a+d)^2} \tag{5.34}$$

The point of $P_{vmax}=P_v$ in (5.34) is the bifurcation point.

The graphical representation of (5.29) is called the excitation characteristic curve and (5.32) is called the power-voltage (PV) characteristic curve that will be used to analyze the bifurcation behavior under parameters variations.

5.3.3 Hybrid Control Modeling of an UPS

The induction load as considered in previous section is now expanded to include a vital load with an UPS. This system is shown in Figure 5.3-5. We need to consider three situations: unconnected, discharging and charging. The battery itself is modeled using the simplest reasonable model - which appears to be that in [55]. This model is composed of a differential equation describing the battery 'state of charge' σ and an output map that gives the battery terminal voltage v_b as a function of the state of charge.

$$\frac{d}{dt}\sigma = \frac{1}{C}i, v_b = f\left(\sigma\right), 0 \le \sigma \le 1 \qquad (5.35)$$

where i is the battery charging current and C is the battery effective capacitance.

The battery is connected to the DC load bus through a DC-DC converter. We consider three UPS modes:

- Battery unconnected: in this case battery current is zero, $i = 0$.
- Battery discharging; the battery and vital load are detached from the rest of the network. The battery alone supplies the load through a voltage-controlled DC-DC converter set up to keep the load voltage constant. The load is constant admittance. Typically we will have, $i < 0$.
- Battery charging: In this mode the battery is charge through a DC-DC converter operated in current controlled mode, $i > 0$. The current is controlled to a specified value.

This system includes operational constraints that we impose on the system. For example, we only allow battery charging when the system is operated without any load shedding. Also, the battery is not to be used to supply power to the system when the system is operated without any load shedding; it may be used only when some load shedding has been initiated. In other words, we want the battery to protect vital loads and not to be used as an alternative to load shedding unless the supply to vital loads is compromised.

The power system has a vital load supplied from the DC bus. We assume that the AC-DC rectifier is voltage regulated, i.e., it controls the voltage on Bus 3 to a constant voltage.

Also, we assume that the rectifier is power factor corrected so that the power factor is approximately 1. This means that from the AC side of the rectifier, the vital load looks like a resistance. The vital load is therefore characterized by an effective resistance, R_v so that the admittance is:

$$Y_v = \frac{1}{R_v} \tag{5.36}$$

Nominal Operation (no UPS, no battery charging, modes, q_1, q_2, q_3)

The network and the load characteristics are as defined in (5.29) and (5.32)

$$\begin{gathered}
V_2 = 1, E = \frac{\sqrt{\left(c + P_v\right)^2 + (a+d)^2}}{a}, \quad 0 < P_v \\
E = 2, V_2 = V_2 = \sqrt{\frac{(2a^2 - cP_v) \pm \sqrt{4a^4 - 4cP_v a^2 - (a+d)^2 P_v^2}}{c^2 + (a+d)^2}}, \\
0 < P_v < \frac{2a^2(\sqrt{c^2 + (a+d)^2} - c)}{(a+d)^2}
\end{gathered} \tag{5.37}$$

Note that once the excitation system saturates there is an upper limit to P_v. This is the voltage collapse bifurcation point that is specified in the third equation of (5.37) . Also, these relations are only good for $P_v > 0$. When $P_v = 0$ we need to use (5.21) . The first equation does approach the proper limit as $P_v \to 0$, but the second does not. This is as it should be. There are two different operating curves; one with load bus voltage controlled and the other with excitation saturated. The limiting cases are different.

Battery Discharging (UPS active, modes, q_5, q_6)

Now, consider the discharging situation. In this case the vital load and UPS are separated from the rest of the system. Thus, the load on the network is reduced so that the network equations are the original and load parameters return to those given by (5.20). The DC-DC converter regulates its output bus to $V_3 = const.$ so that the battery current is $i = -V_3 / R_v$ and

$$\frac{d\sigma}{dt} = -\frac{V_3}{CR_v} \tag{5.38}$$

Battery Charging (mode, q_4):

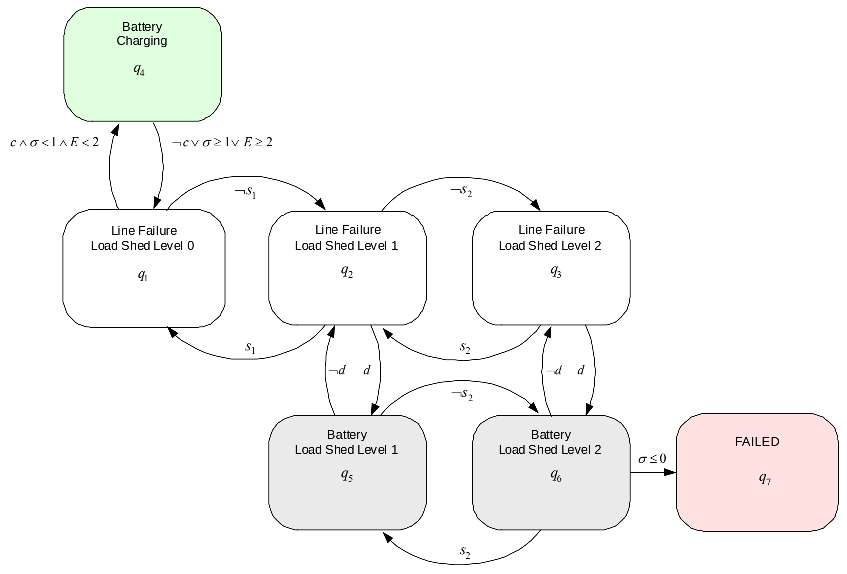
Now consider battery charging. In this case the DC-DC converter operates in current control mode do the battery is charged with constant current, $i = i_c$, so long as $\sigma < 1$. For $\sigma \geq 1$, $i = 0$. The latter situation triggers a transition out of the charging state. While charging we have:

$$\frac{d\sigma}{dt} = \frac{i_c}{C} \tag{5.39}$$

From the AC side of rectifier, the charging looks like an additional constant power load, $P_c = V_3 i_c$. The network behavior is simply the first equation of (5.37). Should, the excitation saturate there is a transition out of the charging state.

5.3.4 Logical specification and IP Formulas:

State Transition Diagram



Battery Charging q_4
$c \wedge \sigma < 1 \wedge E < 2$
$\neg c \vee \sigma \geq 1 \vee E \geq 2$
Line Failure Load Shed Level 0 q_1
Line Failure Load Shed Level 1 q_2
Line Failure Load Shed Level 2 q_3
$\neg s_1$
$\neg s_2$
s_1
s_2
$\neg d$ d
$\neg d$ d
$\neg s_2$
Battery Load Shed Level 1 q_5
Battery Load Shed Level 2 q_6
$\sigma \leq 0$
FAILED q_7
s_2

Figure 5.3-5: Transition diagram of system with UPS

Notice that we include a ‘failed’ state. A very high cost is imposed on the failed state. Hence, the optimal control should avoid it. In our formulation failure occurs when the batter is depleted ($\sigma \leq 0$) and it is not possible to supply the vital loads from other sources.

Logical specification

We can write the logical specification based on the transition diagram described above.

The discrete states: $Q = \{q_1, q_2, q_3, q_4, q_5, q_6, q_7\}$

The state q_1 is the initial state; it corresponds to the state of the system following a line fault.

The states q_2 and q_3 correspond to the state of the system following the load shedding fraction of η_1 and η_2 respectively.

The state q_4 is charging state. In this configuration the battery can only be charged from the transition q_1 to q_4.

The state q_7 the failed state. When a transition to the state q_7 occurs the system fails.

The event state is $E = \{s_1, s_2, c, d\}$, s_1 and s_2 are the switching events enabling the states q_1 , q_2 and q_3. The event *d* enables the discharging state q_5 and q_6. The event *c* enables the charging state q_4.

$$
\begin{aligned}
spec1 = {} & exactly[1,\{q_1,\ q_2,\ q_3,\ q_4,\ q_5,\ q_6,\ q_7\}] \wedge exactly[1,\{q_1^+, q_2^+, q_3^+, q_4^+, q_5^+, q_6^+, q_7^+\}] \wedge \\
& ((q_1 \wedge (\neg s_1)) \Rightarrow q_2^+) \wedge\ ((q_1 \wedge c) \Rightarrow q_4^+) \wedge\ ((q_1 \wedge (\neg(\neg s_1 \vee (c)))) \Rightarrow q_1^+) \wedge \\
& ((q_2 \wedge (\neg s_2)) \Rightarrow q_3^+) \wedge\ ((q_2 \wedge s_1) \Rightarrow q_2^+) \wedge\ ((q_2 \wedge d) \Rightarrow q_5^+) \wedge ((q_2 \wedge (\neg(s_1 \vee (\neg s_2) \vee d))) \Rightarrow q_2^+) \wedge \\
& ((q_3 \wedge s_2) \Rightarrow q_2^+) \wedge ((q_3 \wedge d) \Rightarrow q_6^+) \wedge ((q_3 \wedge \neg(s_1 \vee d)) \Rightarrow q_3^+) \wedge \\
& ((q_4 \wedge (\neg c)) \Rightarrow q_1^+) \wedge ((q_4 \wedge \neg(\neg c)) \Rightarrow q_4^+) \wedge \\
& ((q_5 \wedge (\neg d)) \Rightarrow q_2^+) \wedge ((q_5 \wedge (\neg s_2)) \Rightarrow q_6^+) \wedge ((q_5 \wedge (\neg((\neg d) \vee (\neg s_2)))) \Rightarrow q_5^+) \wedge \\
& ((q_6 \wedge s_2) \Rightarrow q_5^+) \wedge\ ((q_6 \wedge (\neg d)) \Rightarrow q_3^+) \wedge ((q_6 \wedge (\neg(s_2 \vee (\neg d)))) \Rightarrow q_6^+) \wedge \\
& (q_7 \Rightarrow q_7^+)
\end{aligned}
\tag{5.40}
$$

Logic Constraints for Improving Computational Speed

Logic constraints are Boolean expressions that will play an important role in restricting the computational domain of the mixed-integer dynamic programming. The idea is based on the fact that a current state, events and their one-step successor states, events are the only valid local states and events. The two-step or higher-step successor states and events from a current state are specified as disable. This domain decomposition translates to improving the computational speed of mixed-integer dynamic programming algorithm.

The following logic constraints which we call Control Logic, is set up so that disable states and events are discarded from the logic constraint during computation.

$$
\begin{aligned}
&\textit{Control Logic}= \\
&\{(q_1 \Rightarrow (\neg s_2 \wedge \neg d \wedge atmost[1,\{c,\neg s_1\}] \wedge \neg qq_3 \wedge \neg qq_5 \wedge \neg qq_6 \wedge \neg qq_7 \wedge exactly[1,\{qq_1,qq_2,qq_4\}])), \\
&\ (q_2 \Rightarrow (\neg c \wedge atmost[1,\{d,s_1,\neg s_2\}] \wedge \neg qq_4 \wedge \neg qq_6 \wedge \neg qq_7 \wedge exactly[1,\{qq_1,qq_2,qq_3,qq_5\}])), \\
&\ (q_3 \Rightarrow (\neg s_1 \wedge c \wedge atmost[1,\{d,s_2\}] \wedge \neg qq_1 \wedge \neg qq_4 \wedge \neg qq_5 \wedge \neg qq_7 \wedge exactly[1,\{qq_2,qq_3,qq_6\}])), \\
&\ (q_4 \Rightarrow (\neg s_1 \wedge \neg s_2 \wedge d \wedge \neg qq_2 \wedge \neg qq_3 \wedge \neg qq_5 \wedge \neg qq_6 \wedge \neg qq_7 \wedge exactly[1,\{qq_1,qq_4\}])), \\
&\ (q_5 \Rightarrow (\neg s_1 \wedge c \wedge atmost[1,\{\neg d,\neg s_2\}] \wedge \neg qq_1 \wedge \neg qq_3 \wedge \neg qq_7 \wedge exactly[1,\{qq_2,qq_5,qq_6\}])), \\
&\ (q_6 \Rightarrow (\neg s_1 \wedge c \wedge atmost[1,\{\neg d,s_2\}] \wedge \neg qq_1 \wedge \neg qq_2 \wedge \neg qq_4 \wedge exactly[1,\{qq_3,qq_5,qq_6,qq_7\}])), \\
&\ (q_7 \Rightarrow (\neg d \wedge \neg c \wedge \neg s_1 \wedge \neg s_2 \wedge \neg qq_1 \wedge \neg qq_2 \wedge \neg qq_3 \wedge \neg qq_4 \wedge \neg qq_5 \wedge \neg qq_6 \wedge \neg qq_7 \wedge exactly[1,\{qq_7\}]))\}
\end{aligned}
$$

(5.41)

The first constraint in the Control Logic (5.41), tell us that while in state q_1 the event set $\{s_2,\ d\}$ are disabled and at most one the event $\{c,\ s_1\}$ is enabled. And that higher-step states $\{qq_3,\ qq_5,\ qq_6,\ qq_7\}$ are disabled and the one-step states $\{qq_1,\ qq_2,\ qq_4\}$ could be enabled.

IP Formulas:

The transition specification in IP form:

$$\text{IP1} = \text{GenIP[spec1, } \{q_1,\ q_2,\ q_3,\ q_4,\ q_5,\ q_6,\ q_7,\ q_1^+,\ q_2^+,\ q_3^+,\ q_4^+,\ q_5^+,\ q_6^+,\ q_7^+\}]$$

$$\begin{gathered}\text{IP1} = \{\ 1-\delta_{q_1}-\delta_{q_2}-\delta_{q_3}\geq 0,\quad -1+\delta_{q_1}+\delta_{q_2}+\delta_{q_3}\geq 0,\ 1-\delta_{q_1^+}-\delta_{q_2^+}-\delta_{q_3^+}\geq 0,\\ -1+\delta_{q_1^+}+\delta_{q_2^+}+\delta_{q_3^+}\geq 0,\ 1-\delta_{q_1}+\delta_{q_1^+}-\delta_{s_1}\geq 0,\quad 1-\delta_{q_2}+\delta_{q_1^+}-\delta_{s_1}\geq 0,\\ 1-\delta_{q_2}+\delta_{q_2^+}-\delta_{s_2}\geq 0,\ 1-\delta_{q_3}+\delta_{q_2^+}-\delta_{s_2}\geq 0,\ -\delta_{q_1}+\delta_{q_2^+}+\delta_{s_1}\geq 0,\\ -\delta_{q_2}+\delta_{q_3^+}+\delta_{s_2}\geq 0,\quad -\delta_{q_3}+\delta_{q_3^+}+\delta_{s_2}\geq 0,\\ 0\leq\delta_{q_1}\leq 1, 0\leq\delta_{q_2}\leq 1, 0\leq\delta_{q_3}\leq 1,\ 0\leq\delta_{q_1^+}\leq 1,\\ 0\leq\delta_{q_2^+}\leq 1,\ 0\leq\delta_{q_3^+}\leq 1,\ 0\leq\delta_{s_1}\leq 1, 0\leq\delta_{s_2}\leq 1\ \}\end{gathered} \tag{5.42}$$

The IP-formulas for the logical constraint:

$$\text{spec2} = (V_2 = 1 \wedge 0 < V_1 < 2) \vee (V_1 = 2) = ((V_1\ >\ 0)\ \wedge ((\text{-}V_1)\ >\ \text{-}2)\ \wedge\ (V_2\ =\ 1))\ \vee\ (V_1\ =\ 2)$$

$$\text{IP2} = \text{GenIP[spec2, } \{\ 0\leq V_1\leq 1,\ 0\leq V_1\leq 2\ \}]$$

Using the spec2 in GenIP function, gives the following result:

$$\begin{gathered}\text{IP2} = \{\ 3-d_1-V_1>0,\ 1-d_1+V_1>0,\ -2d_2+V_1\geq 0,\ -2d_1+V_2\geq 0,\\ -2+d_1+V_2\leq 0,\ 0\leq d_1, d_2\leq 1,\ 0\leq V_1,\ V_2\leq 2\ \}\end{gathered} \tag{5.43}$$

And the IP-formulas for the load shed parameter η:

$$\begin{aligned}\mathrm{IP3} = \{\ & -0.4d_4 + \eta \geq 0,\ -0.8d_5 + \eta \geq 0,\ d_3 - \delta_{q_1^+} \geq 0,\ d_4 - \delta_{q_2^+} \geq 0,\ d_5 - \delta_{q_3^+} \geq 0 \\ & -1 + d_3 + \eta \leq 0,\ -1 + 0.6d_4 + \eta \leq 0,\ -1 + 0.2d_5 + \eta \leq 0, \\ & 0 \leq d_3 \leq 1,\ 0 \leq d_4 \leq 1,\ 0 \leq d_5 \leq 1,\ 0 \leq \eta \leq 1\ \}\end{aligned} \tag{5.44}$$

Let define two rules, z1Rule and z2Rule corresponding to the voltage regulation case $V_2 = 1$ and the saturated case $V_2 = 2$.

$$\begin{aligned}\mathrm{IP1} = \{\ & 1 - \delta_{q_1} - \delta_{q_2} - \delta_{q_3} \geq 0,\quad -1 + \delta_{q_1} + \delta_{q_2} + \delta_{q_3} \geq 0,\ 1 - \delta_{q_1^+} - \delta_{q_2^+} - \delta_{q_3^+} \geq 0, \\ & -1 + \delta_{q_1^+} + \delta_{q_2^+} + \delta_{q_3^+} \geq 0,\ 1 - \delta_{q_1} + \delta_{q_1^+} - \delta_{s_1} \geq 0,\quad 1 - \delta_{q_2} + \delta_{q_1^+} - \delta_{s_1} \geq 0, \\ & 1 - \delta_{q_2} + \delta_{q_2^+} - \delta_{s_2} \geq 0,\ 1 - \delta_{q_3} + \delta_{q_2^+} - \delta_{s_2} \geq 0,\ -\delta_{q_1} + \delta_{q_2^+} + \delta_{s_1} \geq 0, \\ & -\delta_{q_2} + \delta_{q_3^+} + \delta_{s_2} \geq 0,\quad -\delta_{q_3} + \delta_{q_3^+} + \delta_{s_2} \geq 0, \\ & 0 \leq \delta_{q_1} \leq 1, 0 \leq \delta_{q_2} \leq 1, 0 \leq \delta_{q_3} \leq 1,\ 0 \leq \delta_{q_1^+} \leq 1, \\ & 0 \leq \delta_{q_2^+} \leq 1,\ 0 \leq \delta_{q_3^+} \leq 1,\ 0 \leq \delta_{s_1} \leq 1, 0 \leq \delta_{s_2} \leq 1\ \}\end{aligned} \tag{5.45}$$

Rule 1: z1Rule

$$\begin{aligned}\mathrm{z1Rule} = \mathrm{z1} \to & (\delta_{qq1} + \delta_{qq2} + \delta_{qq3}) \frac{\sqrt{(a^2 + d^2) + (c + P_1)^2}}{a} \\ & + \delta_{qq4} \frac{\sqrt{(a^2 + d^2) + (c + P_2)^2}}{a} \\ & + (\delta_{qq5} + \delta_{qq6} + \delta_{qq7}) \frac{\sqrt{(a^2 + d^2) + c^2}}{a}\end{aligned} \tag{5.46}$$

where $P_1 = \frac{1}{P_v},\ P_2 = \frac{1}{R_v} + I_b,$

Rule 2: z2Rule

$$\begin{aligned}\text{z2Rule} = \text{z2} \to &(\delta_{qq1}+\delta_{qq2}+\delta_{qq3})\frac{\sqrt{a^2-cP+\sqrt{4a^4-4a^2cP_1-(a+d)^2P_1^2}}}{c^2+(a+d)^2}\\ &+\delta_{qq4}\frac{\sqrt{a^2-cP+\sqrt{4a^4-4a^2cP_2-(a+d)^2P_2^2}}}{c^2+(a+d)^2}\\ &+(\delta_{qq5}+\delta_{qq6}+\delta_{qq7})\frac{2a}{\sqrt{c^2+(a+d)^2}}\end{aligned} \tag{5.47}$$

where $P_1=\frac{1}{P_v},\ P_2=\frac{1}{R_v}+I_b,$

The rules z1Rule and z2Rule are specifically designed to satisfy (5.37). The logical specification is given by spec4:

spec4= $((V_2=1)\Rightarrow(V_1=z1))\wedge((V_1=2)\Rightarrow(V_2=z2))$

The corresponding result is obtained as follows:

$$\begin{aligned}\text{IP4}=\{\ &1-3d_6+V_1>0,\ 1-2d_3+V_2>0,\ -1+d_1+d_2+d_3\geq 0,\ -1+d_4+d_5+d_6\geq 0,\\ &2-2d_1+V_1-z1\geq 0,\ 2-2d_4+V_2-z2\geq 0,\ -3+d_5+V_1<0,\ -3+2d_2+V_2<0,\\ &-2+2d_1+V_1-z1\leq 0,\ -2+2d_4+V_2-z2\leq 0,\ 0\leq d_1\leq 1,\ 0\leq d_2\leq 1,\ 0\leq d_3\leq 1,\\ &0\leq d_4\leq 1,\ 0\leq d_5\leq 1,\ 0\leq d_6\leq 1,\ 0\leq V_1\leq 2,\ 0\leq V_2\leq 1,\ 0\leq z1\leq 2,\ 0\leq z2\leq 1\quad\}\end{aligned}$$

(5.48)

5.4 Conclusion

At the foundation of a good analysis lies a good modeling. This chapter had provided the modeling apparatus necessary for hybrid control system analysis.

In this chapter we have selected a 3-bus power system with UPS to illustrate our approach to hybrid system. We provided a detailed description of the 3-bus power system. Models without and with UPS were presented. We gave the derivation of all the important 3-bus power system equations including excitation characteristic, network and load characteristic curves. The elimination method was used to obtain those characteristics equation. Hybrid system modeling is an essential part of this chapter therefore we've presented a model for emergency control using an UPS and gave a concise logical specification of the transition structure of the 3-bus power system with UPS.

CHAPTER 6:BIFURCATION CONTROL OF THE 3-BUS POWER SYSTEM

6.1 Introduction

It is becoming well-accepted in emergency control design of complex system that hybrid structure approach [22] brings new design promises. Network reconfiguration and protection can now be formulated using the hybrid control approach. However, in the hybrid systems literature, very little attention is paid to the role of bifurcation control to the underling continuous-discrete structure. As power systems operate increasingly close to their stability limits due to high load demand, voltage excitation alone is not sufficient to maintain an adequate network voltage profile. The control effectiveness of a generator excitation system is inherently limited, by saturation which affects both the ability to regulate voltage and the stability of the system, consequently the power system voltage inevitably will collapse. Although the basic understanding of the mechanism of voltage collapse and spontaneous oscillation in power systems dynamics are well known, the picture is not complete because voltage collapse usually involves a period of discrete events associated with the action of various protection systems (AVR i.e. exciter control, circuit breakers etc..) intended to prevent failures. The goal of this chapter is to show how bifurcation in a small power system modeled as hybrid system could be used to avoid voltage collapse and how continuous and switching control could be used to establish an appropriate equilibrium operating point.

The structure of the chapter is as follows: In section 6.2, we will describe the stability of equilibria and will give essential definitions and theorems on stability, also the method of linearization will be introduced. In section 6.3 we will present the bifurcation analysis of the 3-bus power system where the network and load equilibrium characteristics curves are described. In section 6.4, the excitation system will be described and its effect on the system will be evaluated. The network and the load characteristic curves will be analyzed and the effectiveness of switching to maintain stable post-fault steady-state operating condition will be investigated. In section 6.5 the bifurcation of diagram of the 3-bus power system that incorporate the dynamics of the motor will be analyzed based on various load shedding.

6.2 Stability of Equilibria

The stability of dynamical systems in literature is numerous. Our interest in stability of power systems, modeled as hybrid system, is voltage stability and voltage collapse prevention. The object of stability analysis is to draw conclusions about the behavior of the system around equilibrium points, and to design feedback strategies that improve the system performance. Hybrid systems arise in practice when there is a need to model discrete changes of a dynamical system due to component failures acting as disturbances. The stability of nonlinear hybrid systems is still an open subject even though much progress had been made in the case of linear hybrid system. The study of power system stability that deals with voltage regulation during component failure can be cast as a problem of existence of equilibrium points and their associated eigentructures.

Definition: Equilibrium points

Let the set of n ordinary differential equations written in a compact form, $\dot{x} = f(x)$. The equilibrium points are given by the solution x^* of the algebraic equation $f(x) = 0$.

Stability of an equilibrium point

An equilibrium point x^* is called stable if all solutions with an initial condition close to x^* remains near x^* for all time. [64].

An equilibrium point that is not stable is called unstable.

One can say a great deal about the qualitative behavior of nonlinear systems in the vicinity of an equilibrium point from the study of their linearization at the equilibrium point. The qualitative behavior of the nonlinear system can often be approximated by the linear systems. The Hartman-Grobman theorem provides a justification for linearization method.

The linearization consists of a small perturbation defined by $\Delta x = x - x^*$ about an equilibrium point. By keeping only the first-order term of the taylor series expansion, one obtained the linear system, $\Delta\dot{x} = A\Delta x$ where $A = \left.\dfrac{\partial f}{\partial x}\right|_{x=x^*} = f_x(x^*)$ is called the Jacobian.

Stable, Unstable and Center Manifolds

Consider the autonomous system $\dot{x} = f(x)$ and suppose $x = 0$ is an equilibrium point so that $f(0) = 0$, let $A = \partial f(0)/\partial x$. Define the three subspaces of R^n:

- The stable subspace, E^s : the eigenspace of eigenvalues with negative real parts.
- The unstable subspace, E^u : the eigenspace of eigenvalues with positive real parts

- The center subspace, E^c : the eigenspace of eigenvalues with zero real parts.

Definition: An equilibrium point x^* of $\dot{x} = f(x)$ is said to be hyperbolic if all the eigenvalues of the Jacobian matrix $D_x f(x^*)$ have nonzero real parts [69]

Theorem: Hartman-Grobman

Let *f(x)* be a C^k vector field on R^n with *f(0)*=0 and $A = f_x(0)$. If *A* is hyperbolic then there is a neighborhood *U* of the origin in R^n on which the nonlinear flow of $\dot{x} = f(x)$ and the linear flow of $\dot{x} = Ax$ are topologically conjugate [59].

Bifurcation

The term “Bifurcation“ originates from the concept of different branches of equilibrium points intersecting each other and thus bifurcating. At the bifurcation points x^* of the system $\dot{x} = f(x)$ the Jacobian $D_x f(x^*)$ is singular [65].

We say that a bifurcation occurs at any point in parameter space, for which the qualitative structure of the system changes for small variation of the parameter vector. The number of solution can change as the parameter is varied. The qualitative changes in the system that will be investigated are:

- Number of equilibrium points.

- Stability of equilibrium points.

Definition

Consider the set of *n* ordinary differential equations given by $\dot{x} = f(x,\mu)$ where *x* is the state vector and μ is a parameter vector.

The equilibrium points are given by the solution x^* of the equilibrium manifold $f(x^*,\mu) = 0$. The existence of equilibrium solution is given by the implicit function theorem.

Theorem : Implicit Function Theorem

Suppose that $f : R^k \times R \to R;\ (x\ ,\ \mu) \mapsto f(x\ ,\ \mu)$, is a C^1 function satisfying

$f(0,0) = 0\ and f_x(0,0) \neq 0$.

Then there are constant $\delta > 0\ and\ \eta > 0,\ and\ C^1\ function$

$\psi : \{\mu : \|\mu\| < \delta\} \to R$ such that for $\psi(0) = 0\ and f(\psi(\mu),\mu)=0$ for $\|\mu\| < \delta$ [69].

The implicit Function Theorem can be used to study the equilibria in the following contest. Let $\dot{x} = f(x,\mu)$ be a differential equation depending on the parameter μ. If $x^* = 0$ is hyperbolic equilibrium point of the differential equation $\dot{x} = f(x,\mu)$ at $\mu = 0$, then the conditions of the Implicit Function Theorem are satisfied. This guarantees that the equation $f(x^*,\mu) = 0$ may be solved locally for $x = \psi(\mu)$ as function of the parameter μ.

Definition: Bifurcation

A one parameter family of vector fields of the form $\dot{x} = f(x, \mu)$ also satisfying the conditions $f(0,0) = 0$, and $D_x f(0,0) = 0$ is said to undergo a bifurcation at $\mu = 0$ if the flow of the equation at $\mu = 0$ is not C^0 equivalent (not qualitatively similar) to the flow for μ near zero [68].

Differential-Algebraic Equations in Power Systems

The power system model that characterizes the system is composed of differential-Algebraic Equations:

$$\begin{aligned} \dot{x} &= f(x, y, \mu) \\ 0 &= g(x, y, \mu) \end{aligned} \qquad (6.1)$$

where $\mu \in R^m$ is the vector of parameter representing change in the system such as control set point (e.g. generator voltage regulator) or perturbation such as (e.g. load changes). The vector $x \in R^n$ is the state vector (e.g. excitation , motor slip , etc) and $y \in R^p$ is the vector of algebraic constraints variables (e.g. phase-angle, voltage etc..) The functions $f : R^{n+p+m} \to R^m, g : R^{n+p+m} \to R^p$ define the evolution of the state.

By collecting the variable x and y into a single vector $z = [x \quad y]^T$ the DAE (6.1) can be written in compact form as:

$$\dot{z} = F(z, \mu) \qquad (6.2)$$

The equilibrium points of the DAE in (6.1) is determine by the solutions of:

$$\begin{aligned} 0 &= f(x, y, \mu) \\ 0 &= g(x, y, \mu) \end{aligned} \tag{6.3}$$

Equation (6.3) represents the load flow solution of a power system.

The stability of the equilibrium is determined by linearizing about the equilibrium points.

Using the taylor series expansion on equation (6.1) the linearized equation is:

$$\begin{bmatrix} \Delta\dot{x} \\ 0 \end{bmatrix} = J \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}$$

$$\textit{where } \mathrm{J}=\begin{bmatrix} f_x & f_y \\ g_x & g_y \end{bmatrix} \text{ is the unreduced Jacobian of (6.3)} \tag{6.4}$$

If we assume that g_y is nonsingular (det $g_y \neq 0$), then the algebraic variable Δy can be eliminated in (6.4) by the Schur complement:

$$\Delta\dot{x} = \left[f_x - f_y g_y^{-1} g_x \right] \Delta x \tag{6.5}$$

or

$$\Delta\dot{x} = A\Delta x$$

$$\textit{where } \mathrm{A}=\mathrm{F_z} = \left[f_x - f_y g_y^{-1} g_x \right] \text{ is the reduced Jacobian of (6.3)} \tag{6.6}$$

$$\textit{for } \mathrm{z}=\begin{bmatrix} x & y \end{bmatrix}^T$$

Fold bifurcation: Fold bifurcation is the basic mechanism by which fixed points or equilibrium points are created and destroyed. As a parameter is varied, two fixed points move toward each other, collide, and mutually annihilate [67]

Theorem: Sufficient Condition for fold bifurcation

A fold bifurcation occur at the equilibrium (z^*, μ^*) where the Jacobian $D_zF|_*$ of (6.2) has a simple and unique zero eigenvalue, with nonzero right an left eigenvectors v and w, respectively, i.e.

$$\begin{aligned} &F(z^*,\mu^*)=0 \\ &D_ZF|_* v = D_Z^TF|_* w = 0 \\ &\|v\| \ or \ \|w\| \neq 0 \end{aligned} \tag{6.7}$$

Proof:

This proof demonstrates that optimization can be used to derive the necessary condition based on the Kuhn-Tucker optimality condition.

Based on the definition of bifurcation point the optimization problem can be written as follows:

$$\begin{aligned} &\min_{z,\mu} \ \zeta(\mu) \\ &s.t \quad F(z,\mu)=0 \end{aligned} \tag{6.8}$$

Let us define the Lagrangian as:

$$L = \zeta(\mu) + w^T F(z,\mu) \tag{6.9}$$

where w is the Lagrange multiplier.

The first order optimality condition is:

$$\begin{aligned} &\nabla_w L = 0 \Leftrightarrow F(z,\mu) = 0 \\ &\nabla_\mu L = 0 \Leftrightarrow \nabla_\mu \zeta + D_\mu^T w = 0 \\ &\nabla_z L = 0 \Leftrightarrow D_z^T F w = 0 \end{aligned} \tag{6.10}$$

The first equation in (6.10) is the equilibrium condition. By choosing ζ such as $\nabla_\mu \zeta \neq 0$ from the second equation in (6.10) it follows that w cannot be zero, hence $\|w\| \neq 0$. Therefore:

$$\begin{aligned} &F(z^*,\mu^*) = 0 \\ &D_Z^T F |_* \, w = 0 \\ &\|w\| \neq 0 \end{aligned} \tag{6.11}$$

The same approach can be used to prove for the right eigenvector by taking an appropriate lagrangian.

Theorem: Necessary condition for fold bifurcation

$$\begin{aligned} &\det g_y \neq 0 \\ &detJ{=}det\begin{bmatrix} f_x & f_y \\ g_x & g_y \end{bmatrix} = \det D_z F = 0 \end{aligned} \tag{6.12}$$

6.3 Static Bifurcation Control Analysis of the 3-bus Power System

Bifurcation control is a control strategy capable of modifying the bifurcation structure of a control system. The role of bifurcation control analysis in power systems is to characterize the behavior of a power system under parameter variations and to investigate the effect of network switching capabilities that may lead to or prevent unstable operating conditions. In view of the recent emergence of rich hybrid systems theory, such a perspective brings new insight into these problems; particularly with respect to the affects of switching. However, the existing theory is primarily limited to linear or benign nonlinear continuous dynamics. Consequently, there is a need for an investigation of bifurcation control structure of the hybrid systems in which the discrete and continuous dynamics exhibit complex behaviors. The loss of a transmission line between buses decreases the maximum power that can be delivered to the load. Since the maximum power that can be delivered is proportional to the admittance of the line, we can model transmission line faults by reducing the nominal parameter a. Our goal is to understand how load shedding can be effectively utilized to insure service of the vital load and to maximize the amount of non-vital load supplied. Considerable insight can be obtained by examining the graphs of the two scalar functions in the $V-s$ plane for fixed values of $a,\ \eta,\ P_m,\ P_v$. The intersections of these graphs are equilibrium points. The graph of the equilibrium points of the form $F(x,\mu)=0$ is defined as the implicit characteristic curve. The graph of the equilibrium points can also be defined as an explicit characteristic curve. Through out this chapter and beyond; an explicit characteristic curve will simply be called the characteristic curve. We will mainly deal with the network characteristic curve and the load characteristic curve.

In this section, starting with network equation for the 3-bus power system, and the load equilibrium equation we will derive the excitation and the network curves by using the elimination method. And we will obtain the load characteristic curves using the load equilibrium equation. Plotting of the different curves will be represented graphically to illustrate influence of control such as load shedding on the stability of the system.

6.3.1 Excitation System's Control and Voltage Regulation

The classical characterization of the generator is a voltage behind transient reactance model. In network studies it is common practice to ignore the dynamics of the excitation system and to consider the voltage (behind reactance) as the network input. In our approach we take this approach and model a static excitation system with a two mode operational: the unsaturated and the saturated modes.

During the unsaturated excitation mode the terminal (or load) voltage is regulated to the value of $V_2 = 1$. The excitation voltage characteristic curve is as in (5.29) is:

$$E = \frac{\sqrt{(a+d)^2 + (c+Pv)^2}}{a} \tag{6.13}$$

The table below describes the excitation voltage for different load shedding and line parameters. Notice that the load admittance parameters *c* and *d* depends on the load shedding.

Table 6-1:Excitation computation

P_m=0.5	η=0	η=0.4	η=0.8
a=1	c_0=1.086, c=1.086, d_0=0.043, d=0.043 $E=\sqrt{1.088+(1.086+P_v)^2}$	c_0=1.086, c=0.652, d_0=0.043, d=0.026 $E=\sqrt{1.052+(0.652+P_v)^2}$	c_0=1.086, c=0.217, d_0=0.043, d=0.008 $E=\sqrt{1.017+(0.217+P_v)^2}$
a=0.7	c_0=1.086, c=1.086, d_0=0.043, d=0.043 $E=1.428\sqrt{0.552+(1.086+P_v)^2}$	c_0=1.086, c=0.652, d_0=0.043, d=0.026 $E=1.428\sqrt{0.527+(0.652+P_v)^2}$	c_0=1.086, c=0.217, d_0=0.043, d=0.008 $E=1.428\sqrt{0.502+(0.217+P_v)^2}$
a=0.375	c_0=1.086, c=1.086, d_0=0.043, d=0.043 $E=2.667\sqrt{0.175+(1.086+P_v)^2}$	c_0=1.086, c=0.652, d_0=0.043, d=0.026 $E=2.667\sqrt{0.161+(0.652+P_v)^2}$	c_0=1.086, c=0.217, d_0=0.043, d=0.008 $E=2.667\sqrt{0.147+(0.217+P_v)^2}$

So long as the load bus voltage $V_2 = 1$, the system equilibrium operating points are completely characterized by the motor behavior .

Table 6-2:Colors:

Cyan: No load shedding *(η = 0)*
Blue: First load shedding *(η = 0.4)*
Red: Second load shedding *(η = 0.8)*

Table 6-3 below gives the results of the feasibility analysis of the excitation systems.

Table 6-3:Excitation Diagrams

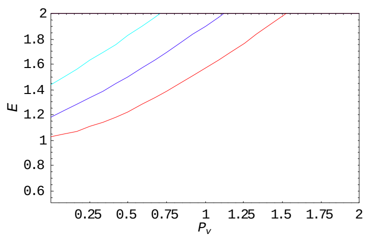


Figure 6.3-1: Excitation voltage

a=1

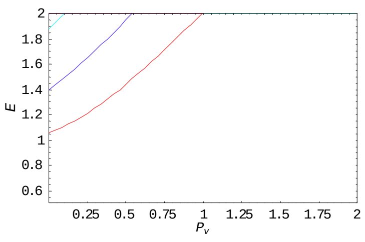


Figure 6.3-2: Excitation voltage

a=0.7

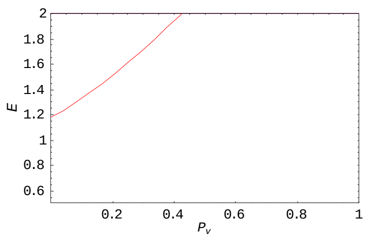


Figure 6.3-3: Excitation voltage

a=0.375

Figure 6.3-1 shows the excitation voltage required to maintain 1 pu load bus voltage with nominal transmission line, *a=1*. The three curves correspond to *η=0, 0.4, 0.8,* respectively left to right. The P_v values at saturation for the corresponding *η* are *0.6199, 1.0649, 1.5097*. Figure 6.3-2 is similar except the transmission line is impaired with *a=0.7*. Notice that the excitation voltage saturates at 2 pu for sufficiently large P_v. The P_v values at saturation for the corresponding *η* are *0.0999, 0.5452, 0.9901*, Figure 6.3-3 the transmission line is more severely impaired with *a=0.375*. Here, only *η* =0.8 allows regulation of load bus voltage at least for some values of P_v. The P_v values at saturation for *η=0.8* is *0.4271.*

Table 6-4 below, describes the values of the regulated load bus voltage V_2 and the power consumed by the vital load P_v, for different load shedding, *η*, and transmission line parameters, *a*, when the excitation reaches it saturation limits of *E=2.*

Table 6-4: Excitation, Voltage, Power Consumed

P_m=0.5,s=0.1474	a=1, E=2	a=0.7, E=2,	a=0.375, E=2,
η=0	V_2=1, P_v=0.6199	V_2=1, P_v=0.0999	V_2=1, P_v=-0.4638
η=0.4	V_2=1, P_v=1.0649	V_2=1, P_v=0.5452	V_2=1, P_v=-0.018
η=0.8	V_2=1.4632, P_v=1.5097	V_2=1.3358, P_v=0.9914	V_2=1, P_v=0.4271

6.3.2 The Load and The Network Characteristic Curves

The graphical representation of the load and the network characteristic curves are powerful geometric concept for investigation the steady-state operating equilibrium point of a simple power system.

6.3.2.1 The Load Characteristic:

We will use the load equilibrium curve (i.e. the slip equation of the induction motor) to derive the load characteristic curve.

Slip equation:

$$\dot{s} = \frac{1}{I_m \omega_0^2}\left(P_m - P_e\right) = \frac{1}{I_m \omega_0^2}\left(P_m - V_2^2 \frac{R_r s(1-s)}{R_r^2 + s^2 X_r^2}\right) \tag{6.14}$$

The load equilibrium condition is obtained as follows:

$$\frac{1}{I_m \omega_0^2}\left(P_m - V_2^2 \frac{R_r s(1-s)}{R_r^2 + s^2 X_r^2}\right) = 0 \tag{6.15}$$

Since the load equilibrium equation is quadratic in V_2, two equilibrium branches exist for fixed P_m. The plotting of slip *vs* V_{2} , reveals a left and a right branches.

The load characteristic curve is:

$$V_2 = \pm\sqrt{P_m\left(\frac{R_r^2 + s^2 X_r^2}{R_r s(1-s)}\right)} \tag{6.16}$$

The load characteristic curve is the voltage at the load bus as a function of the motor slip for constant $P_m = 0.5$.

Representation of The Load Characteristic:

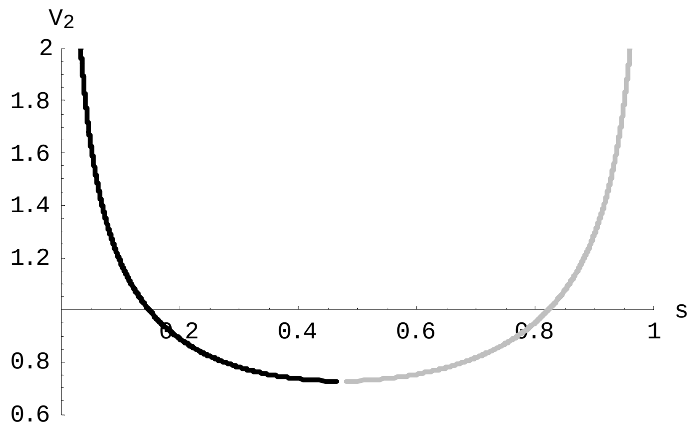


Figure 6.3-4: *s vs V_2 for P_m=0.5*

For a voltage of $V_2 = 1$, there are two equilibrium values of the slip: *s = 0.14750* and *s = 0.82223.*

For a load bus voltage of V_2 = 0.72, there is only one equilibrium point *s = 0.47*. This point is the bifurcation point.

6.3.2.2 The Network and Load Characteristics:

In this subsection we highlight the interaction between the load characteristic and the network characteristic in order to gain insight into the system bifurcation behavior. The governing equations are of the form:

$$g_m(s, V_2, P_m) = 0$$
$$g_n(s, V_2, a, \eta, P_v) = 0$$

The intersection points of the two curves represent various possible operating points of the power system. The stability of those points is crucial to the safe operation of the power system. The absence of the intersection points indicates that the bus voltage will collapse if no action is taken to change parameters of the system. An eigenstructure analysis will reveal the stability type of the equilibrium points.

The operating characteristics of the load and the network will be plotted in the $V_2 - s$ plane for discrete values of P_m, η, a. The load and the network equilibrium manifolds are given by:

$$V_2 = \pm\sqrt{P_m\left(\frac{R_r^2 + s^2 X_r^2}{R_r s(1-s)}\right)} \tag{6.17}$$

$$V_2 = \sqrt{\frac{2a^2 - cP_v \pm \sqrt{4a^4 - 4a^2 cP_v - (a+d)^2 P_v^2}}{c^2 + (a+d)^2}} \tag{6.18}$$

Where $c = (1-\eta)c_0$, $d = (1-\eta)d_0$.

Representation of The Network and The Load characteristic

In the following diagrams for fixed value of P_m in (6.17) and fixed value of P_v, a, in (6.18), and η a discrete parameter, we can plot the variation of the voltage V_2 as function of the slip s as a family of discrete load shedding curves . For a specific load shedding amount there are two equilibrium points. Due to operating constraints one equilibrium may be favored over the other, the eigenvalue analysis will be used to determine the stable operating equilibrium point.

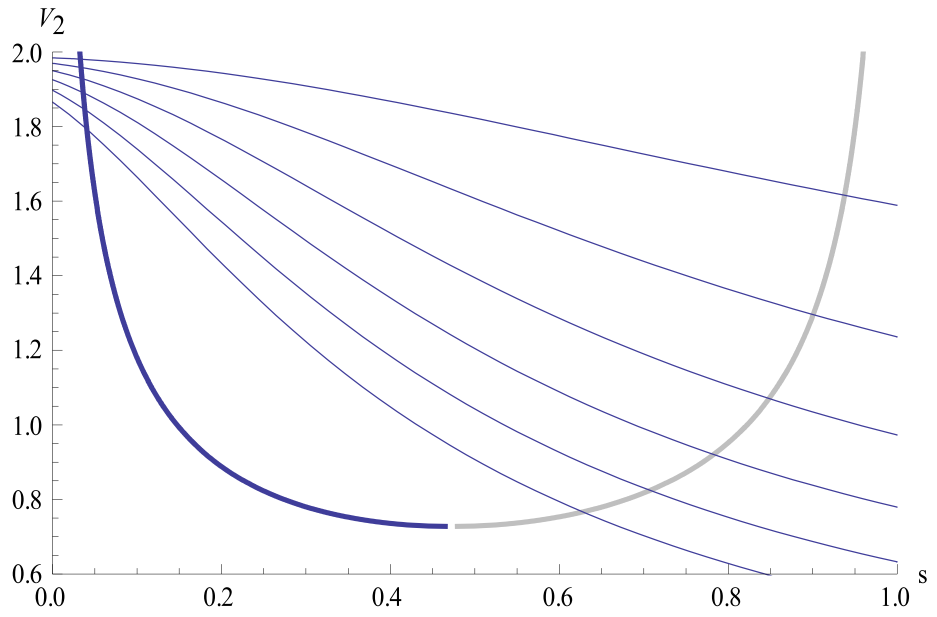


Figure 6.3-5: The nominal network, $a = 1$, graphs are shown with $P_v = 0.3$ and for six values of the load shed parameter, η, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, from left to right respectively. The motor characteristic is shown for $P_m = 0.5$.

In the nominal network, only the upper branches of the network characteristic intersect the motor (load) characteristics branches. The lower branches are not plotted because they don’t contribute to the set of equilibrium points.

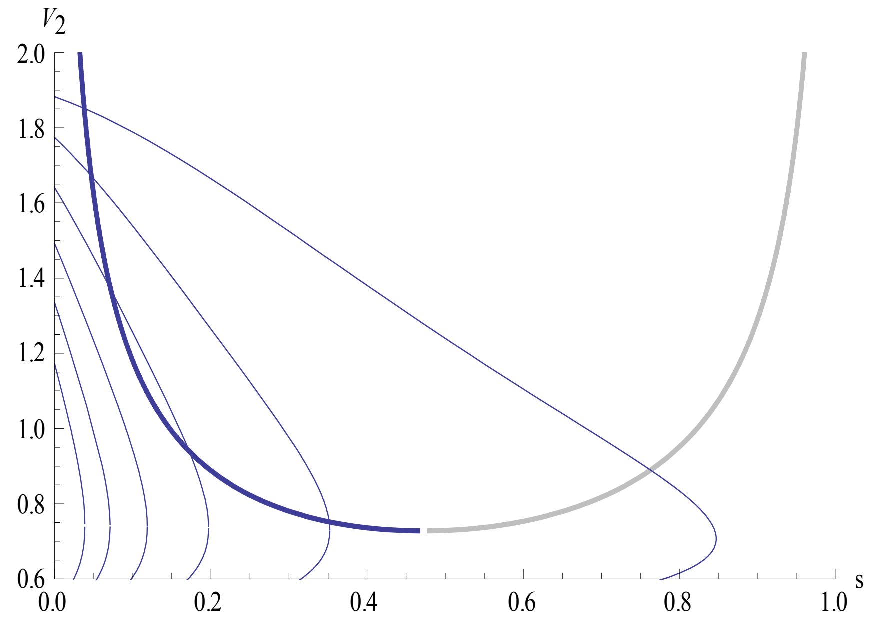


Figure 6.3-6: An impaired network, a = 0.375.

All other parameters are the same

Figure 6.3-6 compares the network and load characteristic for several load shedding amounts with an impaired transmission line. As the parameter *a* decreases higher load shedding is needed to establish equilibrium points. Sufficiently high levels of load shedding exhibit two equilibrium points. Clearly, one critical value of load shedding produces exactly one equilibrium point. In section 6.5, we will discuss the stability of these equilibria.

Stability of equilibrium points

The stability of power system requires that a stable equilibrium point exists. Consequently, we need to assess the stability of the equilibria. This will be done by linearization at each equilibrium point and computation of the eigenvalues.

Number of equilibrium points

The number and the stability of the equilibrium points can be predicted. First, let us look at equilibria:

- When the network characteristic intersects the load characteristic transversally, then two equilibriums points exist, of which one stable and one unstable.
- When the network characteristic intersects the load characteristic, but not transversally, then number of equilibrium is one.
- When the network characteristic doesn't intersect the load characteristic, then there are no equilibria.

We will analyze in detail two cases. Case 1 is for load shedding $\eta = \{0, \mathit{0.4, 0.8}\}$ and a single fixed mechanical, whereas Case 2 is for load shedding $\eta = \{0, \mathit{0.4, 0.8}\}$ and multiple discrete mechanical powers.

Case 1: $\eta = \{0, \mathit{0.4, 0.8}\}$ and $P_m = 0.5$ fixed

For a single mechanical load power the characteristics curves and the system equilibrium states are represented below.

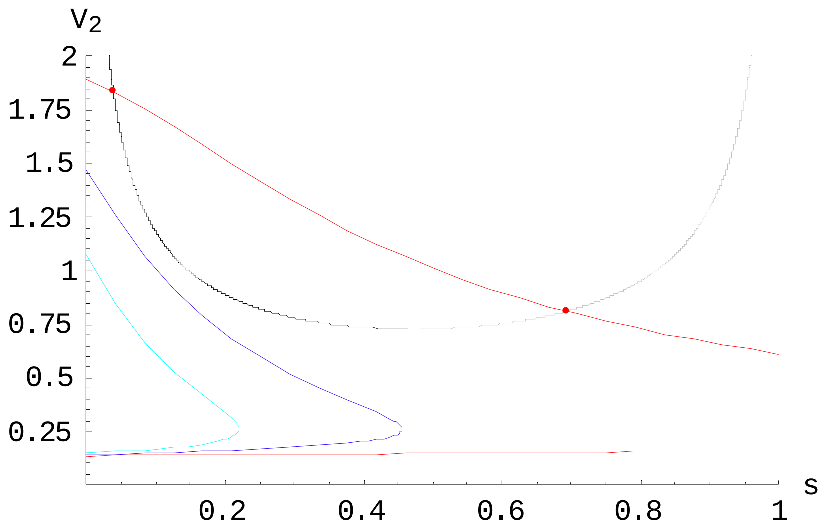


Figure 6.3-7: Equilibrium Points for

η ={0, 0.4, 0.8}, P_m=0.5 P_v=0.1

This diagram shows that the amount of load shedding that is necessary to maintain steady-state equilibrium is $\eta = 0.8$.

Stability and Eigenvalue Computation

The results of the computation, shown in Table 6-5: gives the equilibrium points and their corresponding eigenvalues shown in the table bellow.

Table 6-5: Equilibrium points and corresponding eigenvalues, $P_v = 0.1,\ P_m = 0.5$.

P_v=0.1, P_m=0.5	Equilibrium point and eigenvalue
η=0	None
η=0.4	None
η=0.8	(s,V_2)=(0.03857,1.83), λ=-11.54 (s,V_2)=(0.6923,0.81), λ=2.11

Based on linearization and eigenvalue computation for $P_v = 0.1$ and $P_m = 0.5$, we can study the stability of the two equilibrium points that exist at load shedding $\eta = 0.8$. The stable one, $(s_2, V_2) = \{0.03857, 1.83\}$ and the unstable one, $(s_2, V_2) = \{0.6923, 0.81\}$.

The diagram below, Figure 6.3-8, shows the behavior of the equilibrium points structure of the system when the power consumed by the vital load is $P_v = 0.3$.

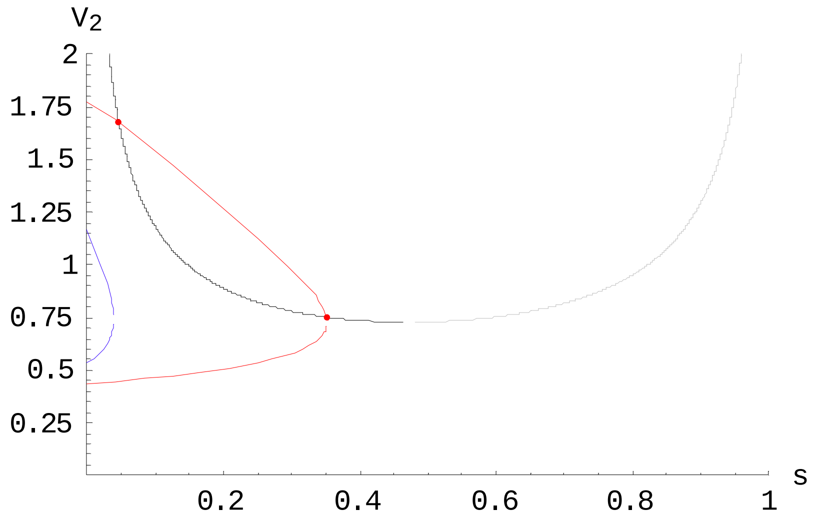


Figure 6.3-8: Equilibrium Points for
η ={ 0.4, 0.8}, P_m=0.5, P_v=0.3

In this scenario where the $P_v = 0.3$, the margin of stability decreases, see Table 6-6, this is reflected in a smaller negative eigenvalue and also, we see that the two equilibria are closer together. The diagram shows that the amount of load shedding that is necessary maintain steady state equilibrium is again $\eta = 0.8$.

Table 6-6: Equilibrium points and corresponding eigenvalues, $P_v = 0.3$, $P_m = 0.5$

P_v=0.3, P_m=0.5	Equilibrium point and eigenvalue
η=0	None
η=0.4	None
η=0.8	(s,V_2)=(0.0469,1.67), λ=-8.70 (s,V_2)=(0.3514,0.75), λ=20.73

The diagram below, Figure 6.3-9, shows the behavior of the equilibrium point structure of the system when the power consumed by the vital load is $P_v = 0.5$.

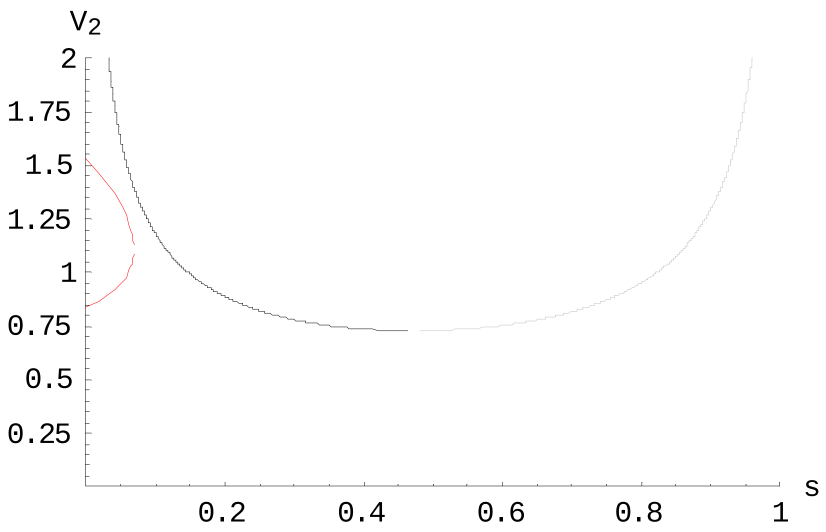


Figure 6.3-9: No equilibrium points for P_v=0.5

In this scenario where the $P_v = 0.5$, the margin of stability vanishes, i.e. no equilibrium point exists and the system will collapse.

Table 6-7: Equilibrium points and corresponding eigenvalues, $P_v = 0.5$ $P_m = 0.5$

P_v=0.5, P_m=0.5	Equilibrium point and eigenvalue
η=0	None
η=0.4	None
η=0.8	None

Case 2: $\eta = \{0,\ 0.4,\ 0.8\}$ and multiple discrete mechanical powers

$P_m = \{0.2,\ 0.4,\ 0.6,\ 0.8,\ 1\}$.

This case is quiet interesting because by using the discrete controls $\eta\, and\ P_m$, we can increase the ability of the power system to maintain stable operating condition. The diagram below, Figure 6.3-10, shows the equilibrium points of the system when the power consumed by the vital load is $P_v = 0.1$.

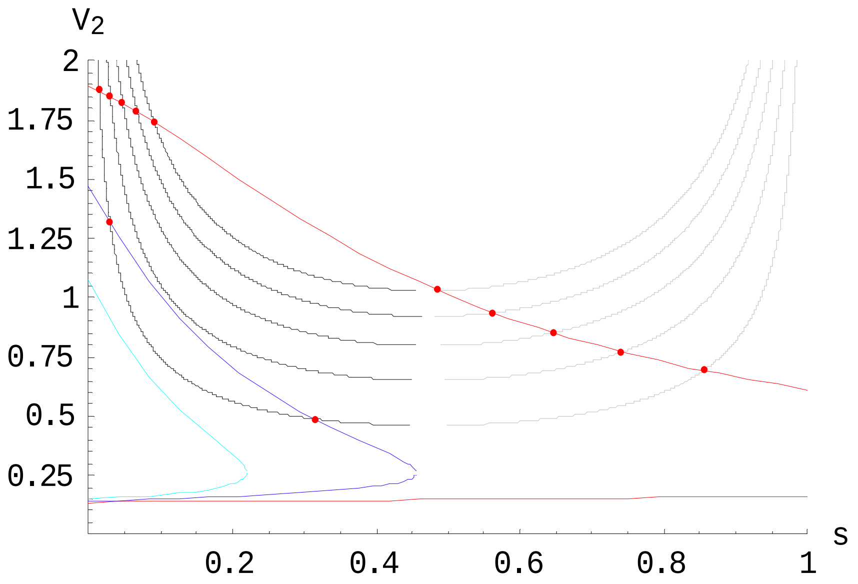


Figure 6.3-10: Equilibrium points for various load shedding values and discrete mechanical power .

From Figure 6.3-10 above, we can conclude that the chance of establishing stable operations has increased due to the multiplicity of discrete mechanical powers. The stability of the equilibrium points is shown in the tables below.

Table 6-8 gives the number of equilibrium points for the set of control $\{\eta, P_m\}$ for $P_v = 0.1$.

Table 6-8:Number of equilibrium points

P_v=0.1	P_m=0.2	P_m=0.4	P_m=0.6	P_m=0.8	P_m=1
η=0	0	0	0	0	0
η=0.4	2	0	0	0	0
η=0.8	2	2	2	2	2

The equilibrium points in Table 6-8 can be expanded to give the stability results for the equilibrium points Table 6-9 gives eigenvalues associated with the equilibrium points.

Table 6-9: Number of equilibrium points and stability analysis.

P_v=0.1	P_m=0.2	P_m=0.4	P_m=0.6	P_m=0.8	P_m=1
η=0	0	0	0	0	0
η=0.4	1 Stables (s,V_2)=(0.03,1.31) λ=-4.953 1 Unstable (s,V_2)=(0.3149,0.49) λ=0.952	0	0	0	0
η=0.8	1 Stable (s,V_2) =(0.014,1.87) λ=-13.322 1 Unstable: (s,V_2) =(0.85,0.69), λ=1.58	1 Stable (s,V_2)=(0.3014,1.85) λ=-12.17 1Unstable: (s,V_2)=(0.7417,0.77) λ=1.19	1 Stable: (s,V_2)=(0.0475,1.82) λ=-10.87 1 Unstable: (s,V_2)=(0.0475,1.82) λ=2.26	1 Stable: (s,V_2)=(0.0673,1.79) λ=-9.4 1Unstable: (s,V_2)=(0.5630,0.94) λ=2.52	1 Stable: (s,V_2)=(0.0907,1.74) λ=-7.85 1 Unstable: (s,V_2)=(0.486,1.03) λ=2.65

By refining the control switching $\eta = \{0.4,\ 0.5,\ 0.6,\ 0.7,\ 0.8,\ 0.9\}$ and

$P_m = \{0.2,\ 0.4,\ 0.6,\ 0.8,\ 1\}$ we obtain a dense bifurcation diagrams with higher number of equilibrium points.

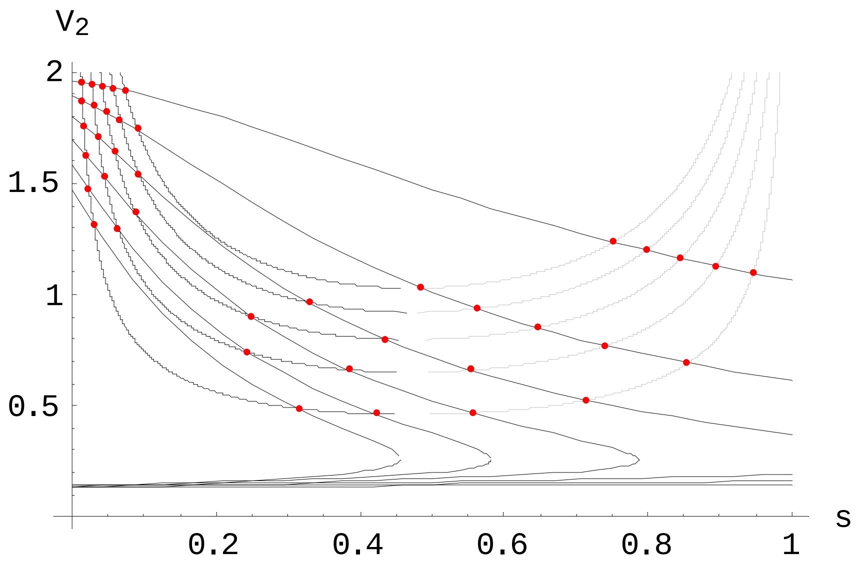


Figure 6.3-11: Equilibrium Points for multiple control switching at P_v=0.1

The computational results in Table 6-9 and Figure 6.3-11 show that the set of equilibrium points on the top left side of the diagram are the stable equilibriums points, while the rest of equilibrium points are the unstable equilibrium points.

The Table 6-10 gives the number of equilibrium points and the type of stability for a set of control $\{\eta,\ P_m\}$ for $P_v = 0.1$. The type of stability can be described as

- *1+1=2: 1 Stable+1 Unstable points*

Table 6-10:The Number of Equilibrium points

P_v=0.1	η=0.4	η=0.5	η=0.6	η=0.7	η=0.8	η=0.9
P_m=0.2	1+1=2	1+1=2	1+1=2	1+1=2	1+1=2	1+1=2
P_m=0.4	0	1+1=2	1+1=2	1+1=2	1+1=2	1+1=2
P_m=0.6	0	0	1+1=2	1+1=2	1+1=2	1+1=2
P_m=0.8	0	0	0	1+1=2	1+1=2	1+1=2
P_m=1	0	0	0	0	1+1=2	1+1=2

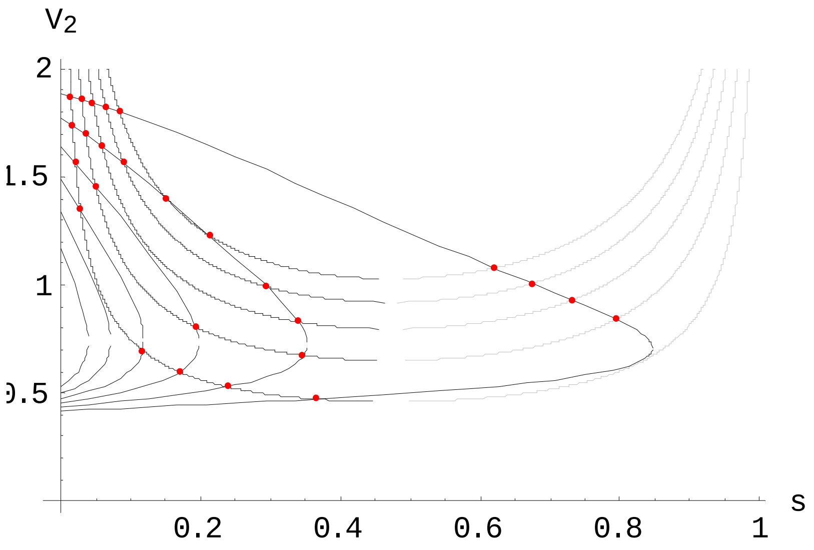


Figure 6.3-12: Equilibrium Points for multiple control switching at P_v=0.3

Figure 6.3-12 gives a similar diagram and the equilibrium points for a set of control $\{\eta,\ P_m\}$ for $P_v = 0.3$.

Table 6-11:Stable Equilibrium points:

P_v=0.3	η=0.4	η=0.5	η=0.6	η=0.7	η=0.8	η=0.9
P_m=0.2	0	0	1+1=2	1+1=2	1+1=2	1+1=2
P_m=0.4	0	0	0	1+1=2	1+1=2	1+1=2
P_m=0.6	0	0	0	0	1+1=2	1+1=2
P_m=0.8	0	0	0	0	1+1=2	1+1=2
P_m=1	0	0	0	0	1+1=2	1+1=2

The Table 6-11 gives the number of equilibrium points and the type of stability for the set of control $\{\eta,\ P_m\}$ for $P_v = 0.3$.

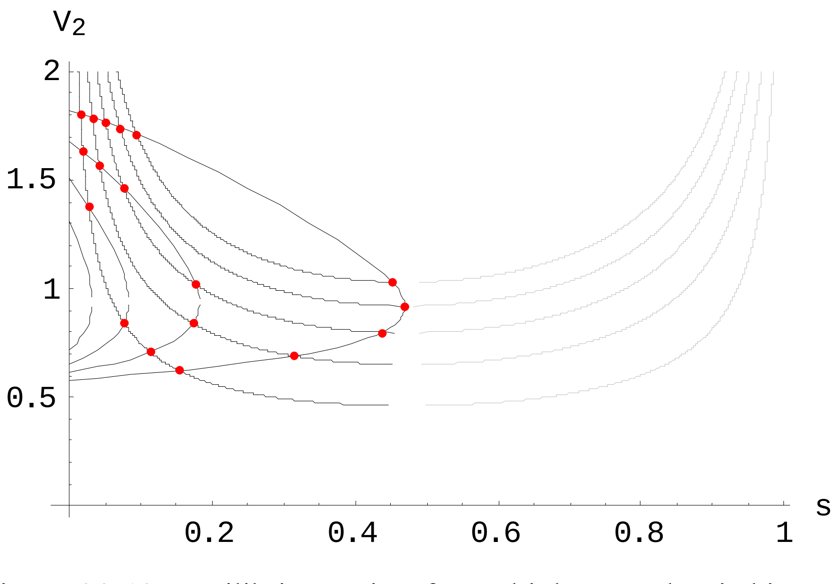


Figure 6.3-13: Equilibrium Points for multiple control switching at P_v=0.4

Figure 6.3-13 gives a similar diagram and the equilibrium points for a set of control $\{\eta,\ P_m\}$ for $P_v = 0.4$.

We can see that as the load power P_v increases the number of equilibrium points decreases.

Table 6-12: Stable Equilibrium points

P_v=0.4	*η=0.4*	*η=0.5*	*η=0.6*	*η=0.7*	*η=0.8*	*η=0.9*
P_m=0.2	*0*	*0*	*0*	*1+1=2*	*1+1=2*	*1+1=2*
P_m=0.4	*0*	*0*	*0*	*0*	*1+1=2*	*1+1=2*
P_m=0.6	*0*	*0*	*0*	*0*	*1+1=2*	*1+1=2*
P_m=0.8	*0*	*0*	*0*	*0*	*0*	*1+1=2*
P_m=1	*0*	*0*	*0*	*0*	*0*	*1+1=2*

Table 6-12 gives the number of equilibrium points and the type of stability for a set of control $\{\eta,\ P_m\}$ for $P_v = 0.4$.

6.4 Voltage Stability Analysis and Bifurcation Curves

In the previous diagrams we used characteristic curves to analyze, the stability of the 3-bus power system, in this section however, we will analyze stability using bifurcation diagrams.

6.4.1 Bifurcation Curves

In this analysis we will us bifurcation curves for assessing voltage regulation and the bifurcation point.

The Differential-Algebraic-Equations of the 3-bus power system

The slip equation (5.16) along with the network equations form the Differential-Algebraic Equations of the 3-bus power systems are:

$$\begin{aligned}
\dot{s} &= \frac{1}{I_m\omega_0^2}\left(P_m - P_e\right) = \frac{1}{I_m\omega_0^2}\left(P_m - V_2^2\frac{R_r s\left(1-s\right)}{R_r^2 + s^2X_r^2}\right)\\
0 &= -P_v + \left(\frac{a}{n}\right)EV_2\sin\theta_2 - cV_2^2\\
0 &= \left(\frac{a}{n}\right)EV_2\cos\theta_2 + (\frac{a}{n} + d)V_2^2
\end{aligned}\qquad(6.19)$$

Since for the unsaturated excitation case, the response of the load is negligible a classical bifurcation analysis will provide the same results as in 6.3.1.What will be analyzed here is the saturated case

The DAE for the saturated case (*E=2*) can be written as:

$$\begin{aligned}
\dot{s} &= \frac{1}{I_m\omega_0^2}\left(P_m - P_e\right) = \frac{1}{I_m\omega_0^2}\left(P_m - V_2^2\frac{R_r s\left(1-s\right)}{R_r^2 + s^2X_r^2}\right)\\
0 &= -P_v + 2\left(\frac{a}{n}\right)V_2\sin\theta_2 - cV_2^2\\
0 &= 2\left(\frac{a}{n}\right)V_2\cos\theta_2 + (\frac{a}{n} + d)V_2^2
\end{aligned}\qquad(6.20)$$

or

$$\dot{s} = \frac{1}{I_m \omega_0^2}\left(P_m - V_2^2 \frac{R_r s(1-s)}{R_r^2 + s^2 X_r^2} \right)$$

$$V_2 = \sqrt{\frac{2a^2 - cP_v \pm \sqrt{4a^4 - 4a^2 cP_v - (a+d)^2 P_v^2}}{c^2 + (a+d)^2}} \tag{6.21}$$

Bifurcation Point:

In section 5.3.2 we derived the bifurcation point as:

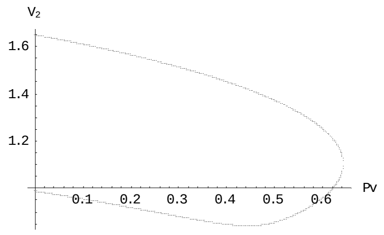
$$P_{v\max} = \frac{2a^2(\sqrt{c^2 + (a+d)^2} - c)}{(a+d)^2} \tag{6.22}$$

Where *c = (1- η)c₀ , d = (1- η)d₀* ,

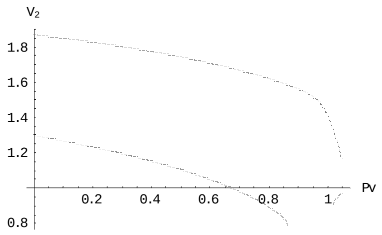
With $c = (1-\eta)c_0,\ c_0 = \left(\frac{1}{R_L} + \frac{R_r s}{R_r^2 + s^2 X_r^2} \right),\quad d = (1-\eta)d_0,\ d_0 = \left(\frac{X_r s^2}{R_r^2 + s^2 X_r^2} \right)$

Normal System: *a = 1*

Table 6-13:Normal system a =1



igure 6.4-1: P_v=0.6199, s=0.1474,V_2=1,η=0



igure 6.4-2: P_v=1.0649, s=0.1474,V_2=1, η=0.4

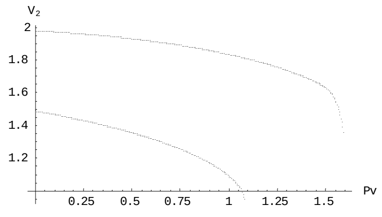


Figure 6.4-3: s=0.1474, η=0.8

These bifurcation curves can be used to visualize the saturation point and the fold bifurcation point. The unsaturated and the saturated excitation system are both depicted in the curves. The unsaturated case corresponds to the regulation line $V_2=1$. The saturation point besides being the point where the excitation system saturates is also the point where both unsaturated and saturate cases are satisfied, which is the point where the switching between the two cases occurs. The saturation point, when it exists, is given without ambiguity by the excitation characteristic curves at saturation as shown in Table 6-3, when using bifurcation curves, from the right to the left, the saturation point is the first intersection point provided that the intersection point belongs to the upper branch when the bifurcation curve is degenerated. (i.e fragmented). In Figure 6.4-1 the excitation saturation point occurs at $P_v=0.6199$ for a load shedding amount of $\eta=0$ and the occurrence of fold bifurcation point is delayed which suggests a security margin ΔP_v greater than zero which represents an increase in the power consumed by the vital load before the bifurcation point is reached.

In Figure 6.4-2 the excitation saturation point occurs at $P_v=1.0649$, and the fold bifurcation seems to occur at the saturation point. Therefore the security margin ΔP_v is close to zero. In

Figure 6.4-3 the excitation voltage is infeasible for maintaining the load bus voltage ($V_2=1$) for the load shedding amount of $\eta=0.8$, which means that even the normal system ($a=1$) can fail to regulate the voltage ($V_2=1$) despite load shedding.

These bifurcation curves show that for load shed of $\eta=0$ and $\eta=0.4$ the voltage regulation is admissible. However for load shedding $\eta=0.8$ the voltage V_2 cannot be regulated at a value of 1.

Impaired System : $a = 0.7$

Table 6-14: Impaired system a=0.7

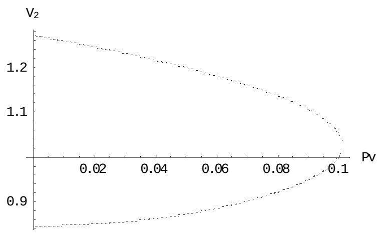


Figure 6.4-4: P_v=0.0999

s=0.1474,V_2=1, η=0

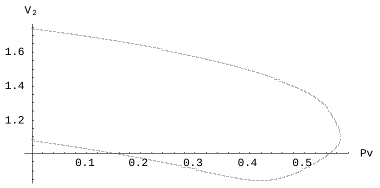


Figure 6.4-5: P_v=0.5452,

s=0.1474,V_2=1, η=0.4

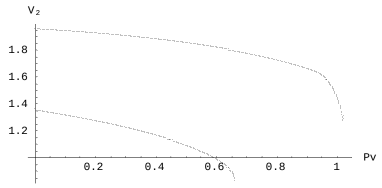


Figure 6.4-6: P_v=

η=0.8

For the impaired system (*a=0.7*) the interpretation is similar to the normal system (*a=1*), In Figure 6.4-4 and in Figure 6.4-5 the excitation saturation points don't coincide with the bifurcation points. And in

Figure 6.4-6 the saturation is infeasible which means that right after the line fault (*a=0.7*) the excitation system fails to regulate the voltage (*V_2=1*) and a voltage collapse will follow.

Severely Impaired System: *a=0.375, η=0.8*

For a severely impaired line only a high load shedding unlike in the two previous cases, is capable to maintain the load voltage during the excitation period.

Table 6-15: Severely impaired system a=0.375, η=0.8

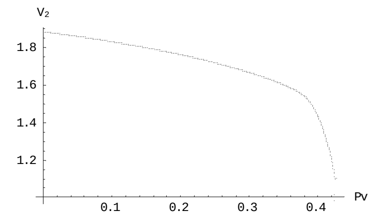


Figure 6.4-7: P_V=0.5452,

s=0.1474, V_2*=1*

The following diagram shows that the load bus voltage V_2 can be regulated with the maximum load shed, and the plots the other shedding are infeasible.

6.5 Conclusion

Bifurcation control was defined as the control of the appearance of bifurcation phenomenon. As technological control systems become more and more complex traditional control studies had failed to explain the global control of systems subject to multiple parameter variation and changing dynamics. At the core of our hybrid bifurcation control study, is not just how continuous parameter variation affects the stability of the system but also how discrete parameter such as load shedding that were used as an effective control for preventing voltage collapse. We have presented the voltage stability analysis for the nominal system, an impaired system and a severely impaired system. Two types of bifurcation control analysis were conducted for the 3-bus network. The first type involved the network and load characteristic curves. This method gives considerable insight as to how various parameters affect the existence of equilibria. The equilibrium points of the 3-bus power system are found to be the points were the two characteristic curves intersect. The analysis of the stability of the equilibrium was relatively straight forward, using linearization and computing the eigenvalues. The second type of bifurcation control analysis, utilizes conventional bifurcation curves for the entire system. This bifurcation control analysis developed in this chapter gives an insight into the how discrete changes in the system affect its capability to continue operation. It suggests ways to set up optimization criteria for the hybrid controller that will be investigated in chapter 7.

CHAPTER 7: OPTIMAL HYBRID CONTROL OF THE 3-BUS POWER SYSTEM WITH UPS

7.1 Introduction

The problem of voltage regulation and emergency control to prevent voltage collapse following severe disturbance in a 3-bus power system with UPS can be formulated as an optimization problem. Optimal control theory has started to play big role in the design of hybrid control and in addition it represents a natural the generalization of all existing control methods such as linear quadratic regulator, dynamic programming, variable structure control and method such Model Predictive Control (MPC) originally designed for process control is now successfully being applied to power system control [58]. The advantage of the MPC method pertains to real-time simulation capability. The innovation in this thesis is the adoption of a new method based on mixed-integer dynamic programming. We will design an optimal controller in an offline fashion that will be utilized in real-time as a lookup table. This chapter is organized as follows. Section 7.2 describes the optimal control problem of the 3-bus power system; section 7.3.1 describes the synthesis of an optimal controller. We will give an overview of the Mathematica code that generates the controller. By using SIMULINK /StateFlow, we will build a simulation framework. The simulation results are presented to illustrate the effectiveness of our approach. In Section 7.4 we will give a summary of the chapter.

7.2 The Optimal Control Problem

The optimal control theory was discussed in CHAPTER 4:. In this section we will define the optimal control problem and present the problem formulation for the case of 3-bus power without UPS and for the case with UPS.

Problem Formulation

Our approach to optimal control design is based on finite, (receding horizon) dynamic programming. Dynamic programming leads to a feedback strategy that can be computed off-line. The plant to be controlled is described by the linear discrete-time dynamic of the motor slip which evolves over a finite time period. This period is divided into *N=25* equally spaced intervals and *k= 0.5* is the discrete time index. All events, controllable or exogenous, are assumed to occur at the beginning of the interval, so we distinguish between values of variables at instant *k*, before any event, k^+ after the event, e.g., s_k, s_{k^+}.

The state space of the slip value is $\{0, 0.1, 0.2, 0.3, 0.4, 0.5\}$. The control variables are $E(k), \eta(k)$ and the regulated variable is V_2. The goal is to keep the load voltage V_2 close to one, specifically, we require $0.95 \le V_2 \le 1.05$. Our intent is to use the field voltage, E, to regulate the terminal voltage, V_2 to 1 p.u. Because $0 < E \le 2$ is constrained, we specify that solutions must satisfy the excitation-controlled voltage constraint

$(V_2 = 1 \wedge 0 < E < 2) \vee (E = 2)$.

The dynamics of the system:

$$s_{k+1} = f\left(s_{k^+}, V_2, \eta\right)$$
$$E = \left(1-\eta\right)\frac{\sqrt{c^2 + (a+d)^2}}{a} V_2 \qquad k = 0,1,\ldots,19 \tag{7.1}$$

Transition, constraints and Cost Function of the 3-bus power system without UPS

In the following transition structure, we allow for the dropping a fraction of the load, η. In the present case, we allow three different values of η including zero, so $\eta \in \{0, \eta_1, \eta_2\}$. Consequently, there is normal operation and two prioritized blocks of load that can be dropped in accordance with the transition behavior defined in Figure 7.2-1.

In the present case, we assume the blocks are sized such that

$$q_1 \Rightarrow \eta = 0, \quad q_2 \Rightarrow \eta = 0.4, \quad q_3 \Rightarrow \eta = 0.8 \tag{7.2}$$

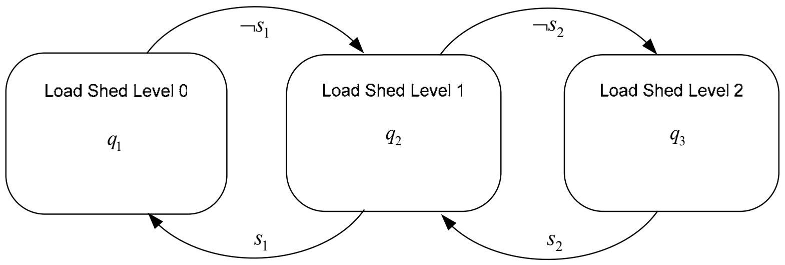


Figure 7.2-1: Transition structure.

The IP formulas obtained from the logical specification of the transition diagram, the excitation-controlled voltage and the load shedding is:

Transition Constraints

$$1-\delta_{q_1}-\delta_{q_2}-\delta_{q_3}\ge 0,\quad -1+\delta_{q_1}+\delta_{q_2}+\delta_{q_3}\ge 0$$
$$1-\delta_{q_1^+}-\delta_{q_2^+}-\delta_{q_3^+}\ge 0,\quad -1+\delta_{q_1^+}+\delta_{q_2^+}+\delta_{q_3^+}\ge 0$$
$$1-\delta_{q_1}+\delta_{q_1^+}-\delta_{s_1}\ge 0,\quad 1-\delta_{q_2}+\delta_{q_1^+}-\delta_{s_1}\ge 0$$
$$1-\delta_{q_2}+\delta_{q_2^+}-\delta_{s_2}\ge 0,\quad 1-\delta_{q_3}+\delta_{q_2^+}-\delta_{s_2}\ge 0$$
$$-\delta_{q_1}+\delta_{q_2^+}+\delta_{s_1}\ge 0$$
$$-\delta_{q_2}+\delta_{q_3^+}+\delta_{s_2}\ge 0,\quad -\delta_{q_3}+\delta_{q_3^+}+\delta_{s_2}\ge 0$$

Excitation -Controlled Voltage Constraints

$$3-d_1-E>0,\quad 1-d_1+E>0,\quad -2d_2+E\ge 0$$
$$-2d_1+V_2\ge 0,\quad -2+d_1+V_2\le 0$$

Load Shedding Constraints

$$-0.4d_4+\eta\ge 0,\quad -0.8d_5+\eta\ge 0,$$
$$d_3-\delta_{q_1^+}\ge 0,\quad d_4-\delta_{q_2^+}\ge 0,\quad d_5-\delta_{q_3^+}\ge 0$$
$$-1+d_3+\eta\le 0,\quad -1+0.6d_4+\eta\le 0,\quad -1+0.2d_5+\eta\le 0$$

Variable Constraints

$$0\le d_3\le 1,\quad 0\le d_4\le 1,\quad 0\le d_5\le 1,\quad 0\le\eta\le 1 \tag{7.3}$$
$$0\le d_1,d_2\le 1,\quad 0\le E,V_2\le 2$$
$$0\le\delta_{q_1}\le 1, 0\le\delta_{q_2}\le 1, 0\le\delta_{q_3}\le 1$$
$$0\le\delta_{q_1^+}\le 1, 0\le\delta_{q_2^+}\le 1, 0\le\delta_{q_3^+}\le 1$$
$$0\le\delta_{s_1}\le 1, 0\le\delta_{s_2}\le 1$$

Cost function:

As an optimal control problem we need a cost function. A cost function is function defined by the designer to make the system behave in a certain way.

We seek an optimal control policy, i.e., a sequence of controls $u(0),\ldots,u(N-1)$, $u(k)=\eta(k)$, that minimizes the cost function

$$J=\sum_{k=0}^{N-1}\left(\left\|V_2(k)-1\right\|^2+r_1\left\|\eta_L(k)\right\|^2\right)/25 \tag{4}$$

subject to the system constraints. We can make some rough assessments of appropriate weighting constants r_1. Load shedding should be avoided with respect to regulating V_2 unless the V_2 tolerance is violated. Hence we want $r_1>0.25^2/0.05^2=1/25$.

Notice that there are a number of variations to this setup. For example, we could penalize soft constraint violations of the generator terminal voltage, V_2, e.g.,

Summary:

$$
\begin{gathered}
J=\sum_{k=0}^{N-1}\left(\left\|V_2(k)-1\right\|^2+r_1\left\|\eta_L(k)\right\|^2\right)/25 \\
s.t \qquad s_{k+1}=f\left(s_{k^+},V_2,\eta\right) \\
E=\left(1-\eta\right)\frac{\sqrt{c^2+(a+d)^2}}{a}V_2 \qquad k=0,1,\ldots,19 \\
1-\delta_{q_1}-\delta_{q_2}-\delta_{q_3}\ge 0,\quad -1+\delta_{q_1}+\delta_{q_2}+\delta_{q_3}\ge 0 \\
1-\delta_{q_1^+}-\delta_{q_2^+}-\delta_{q_3^+}\ge 0,\quad -1+\delta_{q_1^+}+\delta_{q_2^+}+\delta_{q_3^+}\ge 0 \\
1-\delta_{q_1}+\delta_{q_1^+}-\delta_{s_1}\ge 0,\quad 1-\delta_{q_2}+\delta_{q_1^+}-\delta_{s_1}\ge 0 \\
1-\delta_{q_2}+\delta_{q_2^+}-\delta_{s_2}\ge 0,\quad 1-\delta_{q_3}+\delta_{q_2^+}-\delta_{s_2}\ge 0 \\
-\delta_{q_1}+\delta_{q_2^+}+\delta_{s_1}\ge 0 \\
-\delta_{q_2}+\delta_{q_3^+}+\delta_{s_2}\ge 0,\quad -\delta_{q_3}+\delta_{q_3^+}+\delta_{s_2}\ge 0 \\
3-d_1-E>0,\quad 1-d_1+E>0,\quad -2d_2+E\ge 0 \\
-2d_1+V_2\ge 0,\quad -2+d_1+V_2\le 0 \\
-0.4d_4+\eta\ge 0,\quad -0.8d_5+\eta\ge 0, \\
d_3-\delta_{q_1^+}\ge 0,\quad d_4-\delta_{q_2^+}\ge 0,\quad d_5-\delta_{q_3^+}\ge 0 \\
-1+d_3+\eta\le 0,\quad -1+0.6d_4+\eta\le 0,\quad -1+0.2d_5+\eta\le 0 \\
0\le d_3\le 1,\quad 0\le d_4\le 1,\quad 0\le d_5\le 1,\quad 0\le\eta\le 1 \\
0\le d_1,d_2\le 1,\quad 0\le E,V_2\le 2 \\
0\le\delta_{q_1}\le 1,0\le\delta_{q_2}\le 1,0\le\delta_{q_3}\le 1 \\
0\le\delta_{q_1^+}\le 1,0\le\delta_{q_2^+}\le 1,0\le\delta_{q_3^+}\le 1 \\
0\le\delta_{s_1}\le 1,0\le\delta_{s_2}\le 1
\end{gathered}
\tag{7.5}
$$

Due to space limitation the logic constraint is not included in (7.5).

In the following we provide an illustration of the optimal load shedding strategy following a line failure represented as a reduction of a. The feedback control is given as a function of the state - the latter composed of the continuous slip and the three discrete states. At each state, the values of the control actions $\delta_{s_1},\delta_{s_2}$ are given. The controlled transitions are also indicated .

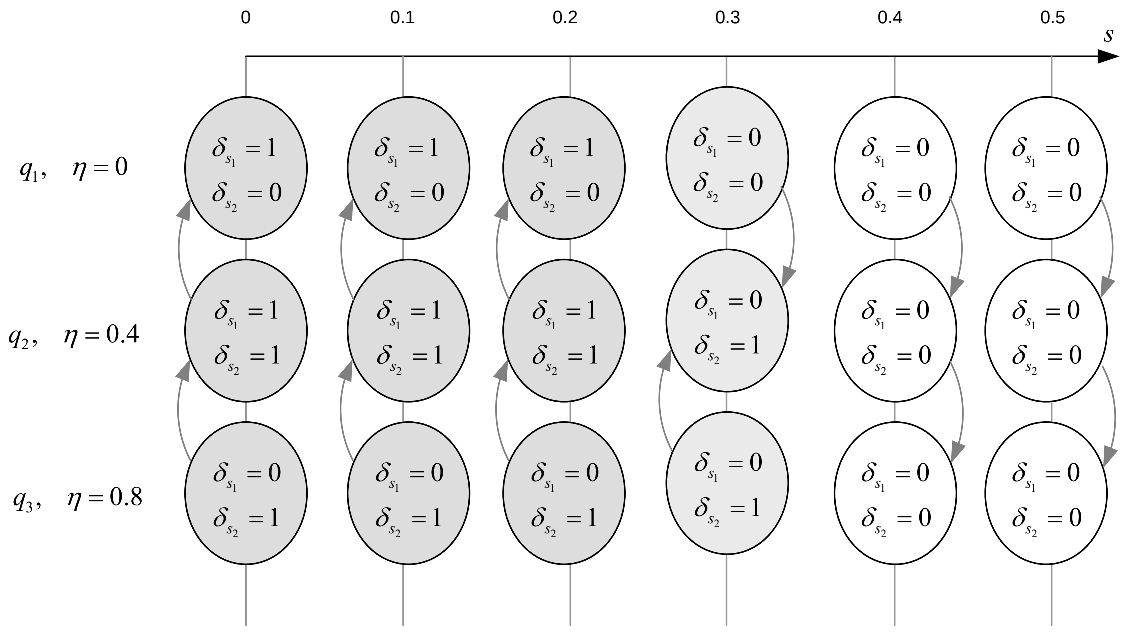


Figure 7.2-2: Depiction of the optimal feedback controller.

Suppose immediately post-failure, the system is in mode q_1, with a reduced slip of 0.1, then the system will respond as follows. Given a mechanical power level of 0.7, the equilibrium slip is about 0.47. As slip increase, the first block of load is dropped at about $s = 0.3$ and the second at about $s = 0.4$.

Power system with UPS

In this case, in addition of the motor slip dynamics we include the dynamic of the battery, a detail modeling of the battery dynamic and the IP formulas were given in subsection 5.3.3.

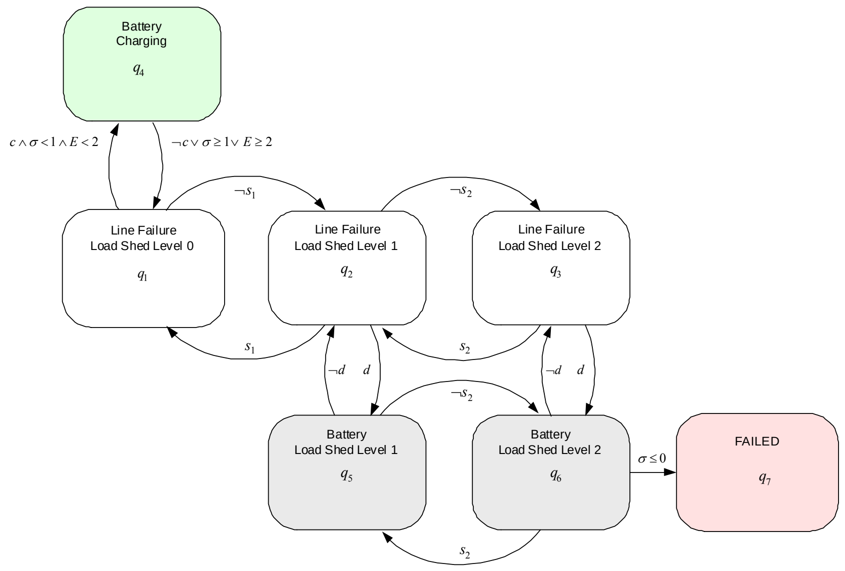

Battery Charging q_4
$c \wedge \sigma < 1 \wedge E < 2$
$\neg c \vee \sigma \geq 1 \vee E \geq 2$
$\neg s_1$
$\neg s_2$
Line Failure Load Shed Level 0 q_1
Line Failure Load Shed Level 1 q_2
Line Failure Load Shed Level 2 q_3
s_1
s_2
$\neg d$
d
$\neg d$
d
$\neg s_2$
Battery Load Shed Level 1 q_5
Battery Load Shed Level 2 q_6
$\sigma \leq 0$
FAILED q_7
s_2

Figure 7.2-3: Transition diagram of system with UPS

Notice that we include a 'failed' state.

Dynamic of the motor slip:

$$s_{k+1} = f\left(s_k, V_2, \eta\right) \tag{7.6}$$

Dynamic of the battery:

$$\begin{gathered}\text{Continuous-time:}\\ \text{Discharging}: \frac{d\sigma}{dt} = \frac{i_c}{C} = -\frac{V_3}{CR_v} \\ \text{Charging}: \quad \frac{d\sigma}{dt} = \frac{i_c}{C} = \frac{P_c}{V_3 C}\end{gathered} \tag{7.7}$$

The two modes can be combined in discrete-time as:

Discrete-Time:

$$\sigma_{k+1} = \delta_c f_c(\sigma_k, i) + \delta_d f_d(\sigma_k, i) \tag{7.8}$$

Optimization Constraints:

- **Equalities constraints: Excitation-Controlled Voltage**

$$z1 = (\delta_{qq1} + \delta_{qq2} + \delta_{qq3})\frac{\sqrt{(a^2+d^2)+(c+P_1)^2}}{a} + \delta_{qq4}\frac{\sqrt{(a^2+d^2)+(c+P_2)^2}}{a} + (\delta_{qq5} + \delta_{qq6} + \delta_{qq7})\frac{\sqrt{(a^2+d^2)+c^2}}{a}$$

$$z2 = (\delta_{qq1} + \delta_{qq2} + \delta_{qq3})\frac{\sqrt{a^2 - cP_1 + \sqrt{4a^4 - 4a^2cP_1 - (a+d)^2P_1^2}}}{c^2+(a+d)^2} + \delta_{qq4}\frac{\sqrt{a^2 - cP_2 + \sqrt{4a^4 - 4a^2cP_2 - (a+d)^2P_2^2}}}{c^2+(a+d)^2} + (\delta_{qq5} + \delta_{qq6} + \delta_{qq7})\frac{2a}{\sqrt{c^2+(a+d)^2}}$$

(7.9)

- **Inequalities constraints: Switching Rules z1, z2 Constraints**

$$\begin{gathered} 1-3d_6+E>0,\ 1-2d_3+V_2>0,\ -1+d_1+d_2+d_3\geq 0,\ -1+d_4+d_5+d_6\geq 0, \\ 2-2d_1+E_1-z1\geq 0,\ 2-2d_4+V_2-z2\geq 0,\ -3+d_5+V_1<0,\ -3+2d_2+V_2<0, \\ -2+2d_1+E_1-z1\leq 0,\ -2+2d_4+V_2-z2\leq 0,\ 0\leq d_1\leq 1,\ 0\leq d_2\leq 1,\ 0\leq d_3\leq 1, \\ 0\leq d_4\leq 1,\ 0\leq d_5\leq 1,\ 0\leq d_6\leq 1,\ 0\leq E_1\leq 2,\ 0\leq V_2\leq 1,\ 0\leq z1\leq 2,\ 0\leq z2\leq 1 \end{gathered} \tag{7.10}$$

Transition Constraints:

$$1-\delta_{q_1}-\delta_{q_2}-\delta_{q_3}\ge 0,\quad -1+\delta_{q_1}+\delta_{q_2}+\delta_{q_3}\ge 0$$
$$1-\delta_{q_1^+}-\delta_{q_2^+}-\delta_{q_3^+}\ge 0,\quad -1+\delta_{q_1^+}+\delta_{q_2^+}+\delta_{q_3^+}\ge 0$$
$$1-\delta_{q_1}+\delta_{q_1^+}-\delta_{s_1}\ge 0,\quad 1-\delta_{q_2}+\delta_{q_1^+}-\delta_{s_1}\ge 0$$
$$1-\delta_{q_2}+\delta_{q_2^+}-\delta_{s_2}\ge 0,\quad 1-\delta_{q_3}+\delta_{q_2^+}-\delta_{s_2}\ge 0$$
$$-\delta_{q_1}+\delta_{q_2^+}+\delta_{s_1}\ge 0$$
$$-\delta_{q_2}+\delta_{q_3^+}+\delta_{s_2}\ge 0,\quad -\delta_{q_3}+\delta_{q_3^+}+\delta_{s_2}\ge 0$$

Excitation -Controlled Voltage Constraints:

$$3-d_1-E>0,\quad 1-d_1+E>0,\quad -2d_2+E\ge 0$$
$$-2d_1+V_2\ge 0,\quad -2+d_1+V_2\le 0$$

Load Shedding Constraints:

$$-0.4d_4+\eta\ge 0,\quad -0.8d_5+\eta\ge 0,$$
$$d_3-\delta_{q_1^+}\ge 0,\quad d_4-\delta_{q_2^+}\ge 0,\quad d_5-\delta_{q_3^+}\ge 0$$
$$-1+d_3+\eta\le 0,\quad -1+0.6d_4+\eta\le 0,\quad -1+0.2d_5+\eta\le 0$$

Variable Constraints:

$$0\le d_3\le 1,\quad 0\le d_4\le 1,\quad 0\le d_5\le 1,\quad 0\le\eta\le 1$$
$$0\le d_1,d_2\le 1,\quad 0\le E,V_2\le 2$$
$$0\le\delta_{q_1}\le 1, 0\le\delta_{q_2}\le 1, 0\le\delta_{q_3}\le 1 \qquad (7.11)$$
$$0\le\delta_{q_1^+}\le 1, 0\le\delta_{q_2^+}\le 1, 0\le\delta_{q_3^+}\le 1$$
$$0\le\delta_{s_1}\le 1, 0\le\delta_{s_2}\le 1$$

Cost function:

Consider the following cost function:

$$J=\sum_{k=0}^{N-1}\left(\left\|V_2(k)-1\right\|^2+r_1\left\|\eta_L(k)\right\|^2+\delta_c(\sigma-1)^2+\delta_d\sigma^2+10\delta qq_7\right)/25 \qquad (7.12)$$

The cost function is a multi-objective cost function that includes a deviation of the reference voltage term $||V_2(k)-1||^2$. The other cost elements are: $||\eta_L(k)||^2$, $(\sigma-1)^2$, σ^2, δqq_7 where the indicator δqq_7 corresponds to the fail mode at the next step of an execution. A very high cost is imposed on the failed state. Hence, the optimal control should avoid it. In our formulation failure occurs when the batter is depleted ($\sigma \leq 0$) and it is not possible to supply the vital loads from other sources. The constants r_1 , *10* are the weighting constants. They play the role of penalties on the load shedding and on the transition to the failed mode. We can make some rough assessments of appropriate weighting constant r_1. Load shedding should be avoided with respect to regulating V_2 unless the V_2 tolerance is violated. Hence we want $r_1 > 0.25^2/0.05^2 = 1/25$. The indicators δ_c , δ_d are binary variables, they correspond respectively to charging and discharging mode of the battery.

Notice that there are a number of variations to this setup. For example, we could penalize soft constraint violations of the generator terminal voltage, V_2, e.g.,

$$J = \sum_{k=0}^{N-1}\left(\left\|V_2\left(k\right)-1\right\|^2 + r_1\left\|\eta_L\left(k\right)\right\|^2 + +\delta_c(\sigma-1)^2 + \delta_d\sigma^2 + 10\delta_7 + r_2\left(\delta_{V_2^+} + \delta_{V_2^-}\right)\right)/25$$

(7.13)

Where we define the binary variables

$$\delta_{V_2^-} = \begin{cases} 1 & V_2 < 0.9 \\ 0 & V_2 \geq 0.9 \end{cases} \qquad \delta_{V_2^+} = \begin{cases} 1 & V_2 > 1.1 \\ 0 & V_2 \leq 1.1 \end{cases} \tag{7.14}$$

To establish appropriate values for r_2, we take the view that it is appropriate to use all available load shedding to eliminate any violation of the constraint (7.14) imposed on V_2. Thus, we should have $r_2 > r_1 \max \|\eta_L\|^2 = 1/100$.

In summary, we have slip dynamics in discrete time form

$$J=\sum_{k=0}^{N-1}\left(\left\|V_2(k)-1\right\|^2+r_1\left\|\eta_L(k)\right\|^2+\delta_c(\sigma-1)^2+\delta_d\sigma^2+10\delta qq_7\right)/25$$

$$s.t:\ s_{k+1}=f\left(s_k,V_2,\eta\right)$$

$$\sigma_{\text{k+1}}=\delta_c f_c\left(\sigma_k,i\right)+\delta_d f_d\left(\sigma_k,i\right)$$

$$\text{z1}=(\delta_{qq1}+\delta_{qq2}+\delta_{qq3})\frac{\sqrt{(a^2+d^2)+(c+P_1)^2}}{a}+\delta_{qq4}\frac{\sqrt{(a^2+d^2)+(c+P_2)^2}}{a}$$

$$+(\delta_{qq5}+\delta_{qq6}+\delta_{qq7})\frac{\sqrt{(a^2+d^2)+c^2}}{a}$$

$$\text{z2}=(\delta_{qq1}+\delta_{qq2}+\delta_{qq3})\frac{\sqrt{a^2-cP_1+\sqrt{4a^4-4a^2cP_1-(a+d)^2P_1^2}}}{c^2+(a+d)^2}$$

$$+\delta_{qq4}\frac{\sqrt{a^2-cP_2+\sqrt{4a^4-4a^2cP_2-(a+d)^2P_2^2}}}{c^2+(a+d)^2}+(\delta_{qq5}+\delta_{qq6}+\delta_{qq7})\frac{2a}{\sqrt{c^2+(a+d)^2}}$$

$$1-3d_6+E>0,\ 1-2d_3+V_2>0,\ -1+d_1+d_2+d_3\geq 0,\ -1+d_4+d_5+d_6\geq 0,$$

$$2-2d_1+E_1-z1\geq 0,\ 2-2d_4+V_2-z2\geq 0,\ -3+d_5+V_1<0,\ -3+2d_2+V_2<0,$$

$$-2+2d_1+E_1-z1\leq 0,\ -2+2d_4+V_2-z2\leq 0,\ 0\leq d_1\leq 1,\ 0\leq d_2\leq 1,\ 0\leq d_3\leq 1,$$

$$1-\delta_{q_1}-\delta_{q_2}-\delta_{q_3}\geq 0,\quad -1+\delta_{q_1}+\delta_{q_2}+\delta_{q_3}\geq 0$$

$$1-\delta_{q_1^+}-\delta_{q_2^+}-\delta_{q_3^+}\geq 0,\quad -1+\delta_{q_1^+}+\delta_{q_2^+}+\delta_{q_3^+}\geq 0$$

$$1-\delta_{q_1}+\delta_{q_1^+}-\delta_{s_1}\geq 0,\quad 1-\delta_{q_2}+\delta_{q_1^+}-\delta_{s_1}\geq 0$$

$$1-\delta_{q_2}+\delta_{q_2^+}-\delta_{s_2}\geq 0,\quad 1-\delta_{q_3}+\delta_{q_2^+}-\delta_{s_2}\geq 0$$

$$-\delta_{q_1}+\delta_{q_2^+}+\delta_{s_1}\geq 0$$

$$-\delta_{q_2}+\delta_{q_3^+}+\delta_{s_2}\geq 0,\quad -\delta_{q_3}+\delta_{q_3^+}+\delta_{s_2}\geq 0$$

$$3-d_1-E>0,\quad 1-d_1+E>0,\quad -2d_2+E\geq 0$$

$$-2d_1+V_2\geq 0,\quad -2+d_1+V_2\leq 0$$

$$-0.4d_4+\eta\geq 0,\quad -0.8d_5+\eta\geq 0,$$

$$d_3-\delta_{q_1^+}\geq 0,\quad d_4-\delta_{q_2^+}\geq 0,\quad d_5-\delta_{q_3^+}\geq 0$$

$$0\leq d_4\leq 1,\ 0\leq d_5\leq 1,\ 0\leq d_6\leq 1,\ 0\leq E_1\leq 2,\ 0\leq V_2\leq 1,\ 0\leq z1\leq 2,\ 0\leq z2\leq 1$$

$$-1+d_3+\eta\leq 0,\quad -1+0.6d_4+\eta\leq 0,\quad -1+0.2d_5+\eta\leq 0$$

$$0\leq d_3\leq 1,\quad 0\leq d_4\leq 1,\quad 0\leq d_5\leq 1,\quad 0\leq\eta\leq 1$$

$$0\leq d_1,d_2\leq 1,\quad 0\leq E,V_2\leq 2$$

$$0\leq\delta_{q_1}\leq 1,0\leq\delta_{q_2}\leq 1,0\leq\delta_{q_3}\leq 1$$

$$0\leq\delta_{q_1^+}\leq 1,0\leq\delta_{q_2^+}\leq 1,0\leq\delta_{q_3^+}\leq 1$$

$$0\leq\delta_{s_1}\leq 1,0\leq\delta_{s_2}\leq 1$$

(7.15)

7.3 Optimal Control and Simulation

The optimal controller consists of a code in Mathematica that implements the mixed integer dynamic programming; more detail on the implementation of the code can be found in the appendix.

7.3.1 Mathematica Code for Controller Synthesis

The Mathematica code is presented as a front end code that implements the mixed integer dynamic programming. The state space is composed of descretized continuous state such as the slip, the battery state of charge one side and the discrete state such as the hybrid modes. The descretization step size, h is fixed.

A cost function with no terminal cost which specified the voltage deviation is proposed. The hybrid modes reflected the simulation through the transition structure. The dynamics of the system are the slip and the battery state of charge. An optimal controller is synthesized as a look up table in term of the status of switches that can be enabled or disabled over the entire time horizon. The code is described below;

```
ContinuousS = Flatten[Outer[List, Range[0.1, 0.5, 0.1], Range[0.25, 1, 0.25]];
h = 0.5; m = 25;r = 1/25;
DiscreteS = Range[7];
S = Map[Flatten[#] &,  Flatten[Outer[List, ContinuousS, DiscreteS, 1], 1]];
G[BinaryVars_List, RealVars_List, StateVars_List,  n_Integer] :=
    $(((V2 - 1)^2 + \delta_c(\sigma - 1)^2 + \delta_d\sigma^2 + r\,\eta^2 + 10\delta_{qq7})/m)/.\{\eta -> (\delta_{qq2} + \delta_{qq5})0.4 + (\delta_{qq3} + \delta_{qq6} + \delta_{qq7})0.8;$
GN[StateVars_List] := 0 ;  (*terminal cost*)
BinaryStateVars = $\{\delta_{q1} + \delta_{q2} + \delta_{q3} + \delta_{q4} + \delta_{q5} + \delta_{q6} + \delta_{q7}\};$
BooleanStateVars = {q1, q2, q3, q4, q5, q6, q7, qq1, qq2, qq3, qq4, qq5, qq6, qq7};
DiscreteState = BooleanStateVars;
NextBinaryState = $\{\delta_{qq1}, \delta_{qq2}, \delta_{qq3}, \delta_{qq4}, \delta_{qq5}, \delta_{qq6}, \delta_{qq7}\};$
BinaryControlVars = $\{\delta_c, \delta_d, \delta_{s1}, \delta_{s2}\};$
BooleanControlVars = {c, d, s1, s2};
RealStateVars = $\{s, \sigma\};$
RealVars = {V1, V2};
AllRealVars = $\{V1, V2, s, \sigma\};$
BinaryVars = Complement[varBinary, BinaryStateVars];
DynamicsRHS = {SlipEq, ChargeEq} $/.\{\eta -> (\delta_{qq2} + \delta_{qq5})0.4 + (\delta_{qq3} + \delta_{qq6} + \delta_{qq7})0.8;$
{T, Control} = OptimalPolicy[eqBinary, eqReal, ControlLogic, BinaryVars, RealVars,
    BinaryStateVars, BooleanStateVars, NextBinaryState, RealStateVars, DynamicsRHS,
    BinaryControlVars, BooleanControlVars, S, GN, G, m]];
```

Description of the Code:

These lines described the meaning of the code. The code appears as implemented in Mathematica except that there is no the line numbering *1* through *18*.

Line 01: Describes the grid design of the continuous space of the two continuous variables respectively: the slip *s*, and the state of charge of the battery, σ.

Line 02: $h = 0.5$ is the descretization step size, m is the number of steps, *r* is the ratio of the penalty weight in the cost function.

Line 03: DiscreteS correspond to the seven modes of the hybrid automaton.

Line 04: S is the hybrid continuous and discrete state space.

Line 05: G corresponds to the running cost of the cost function. The mean square error of the voltage *V1* from the desired voltage 1 pu is minimized as well as the state of charge σ , the load shedding η, and the failed mode qq7. The substitution rule signifies that the load shedding η is set to 0.4 when the system is mode q2 or q4 and it is set 0.8 when the system is in mode q3 or q6 or q7.

Line 06: GN express the fact that the terminal cost is free.

Line 07: BinaryStateVars is the summation of all binary nodes

Line 08: BooleanStateVars is the collection of all modes and next modes.

Line 10: NextStateVars is the collection of the next binary states variables.

Line 11: BinaryControlVars: is the collection of the four binary control variables.

Line 12: BooleansControlVars is the corresponding Booleans control variables of the binary control variables.

Line 13: RealStateVars is the collection two real variables symbols

Line 14: RealVars is the collection of the real variables that are not Real state variables.

Line 15: AllRealVars is the collection Real state variables and real variables.

Line 16: BinaryVars

Line 17: DynamicsRHS={ Slip, Charge} is the two dynamics considered in this example. The substitution rule signifies that the load shedding η is set to 0.4 when the system is mode q2 or q4 and it is set 0.8 when the system is in mode q3 or q6 or q7.

Line 18: {T, Control} is the pair composed of time T, the total computational time and the synthesized optimal feedback controller.

Note: The controller can be viewed from the appendix.

To put the controller in a form readable in Matlab/Simulink, the following mathematica code builds a table lookup and generates a mfile called TableDefs, ready to be used in the Simulink block.

Mathematica command for lookup table building:

Our controller is obtained by an exhaustive optimization over the entire state space, therefore the controller is presented as a look up table. For every continuous and discrete mode state they correspond an optimal action represented by switches enabling or disabling.

BuildLookupTable[{5, 4, 7}, BinaryControlVars, {“AAA”, “BBB”, “CCC”, “DDD”}, Control[[-1]], “TableDefs.m”]

The first argument of BuildLookupTable, {5, 4, 7} is the list of statedimension, where 5 is the dimension of the discretized slip space, 4 is the dimension of the discretized state of charge of the battery, and 7 is the dimension of discrete mode (i.e. Dim[DiscreteS] = 7)

The second argument is the list of Control Variables.

The third argument, {“AAA”, “BBB”, “CCC”, “DDD”} is the list of Table names corresponding to the four control variables.

The fourth argument is the Controller: Control[[-1]]

The fifth argument is the Table look up file name: TableDefs.m

The optimal control was computed in about 9 minutes on a laptop with 1.1 Ghz Pent M.

Building a look up Table

The SIMULINK block diagrams below generate the look up table which represents the controller based on the mixed dynamic programming described earlier.

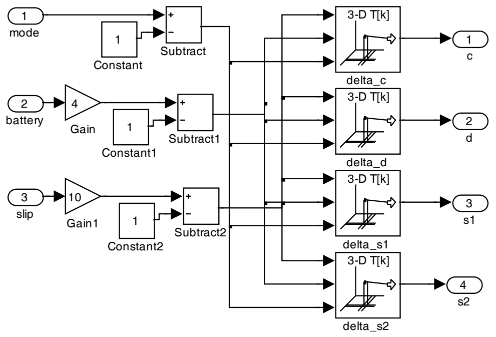


Figure 7.3-1: Look up Table

The first table with output c, is the table AAA. The output c is the discrete action of the battery's charging.

The second table with output d, is the table BBB. The output d is the discrete action of the battery's discharging.

The third table with output s1, is the table CCC. The output s1 is the discrete action of the load shedding with load shedding fraction 0.4.

The fourth table with output s_2, is the table DDD. The output s_2 is the discrete action of the load shedding with load shedding fraction 0.8.

7.3.2 Simulation

Stateflow diagram of 3-bus system with UPS

The state diagram the 7 mode automaton can easily be drawn using the Stateflow tool is the SIMULINK simulation environment.

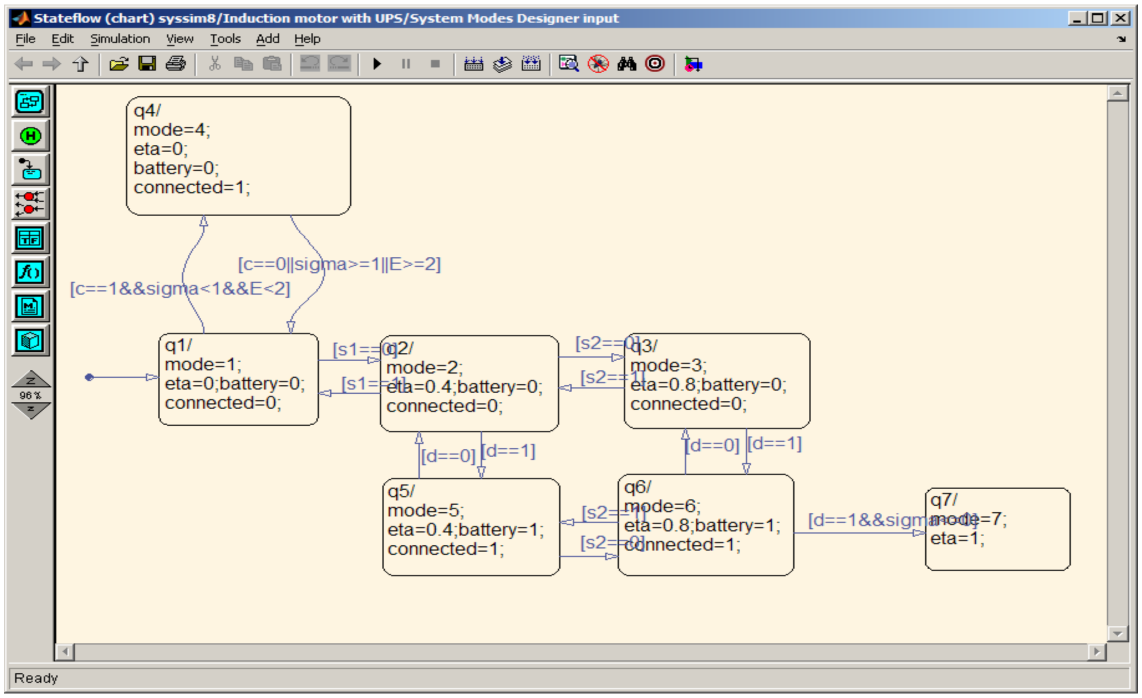


Figure 7.3-2: Stateflow

The seven hybrid modes of the 3-bus power system with UPS are represented using the Stateflow graphical interface. The load shedding and battery's information are indicated in each mode.

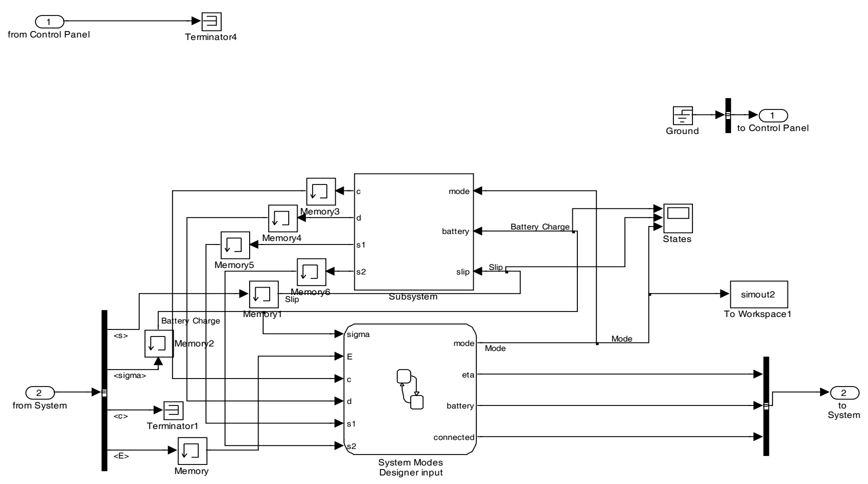


Figure 7.3-3: Optimal Controller

This schematic describes the structure of the optimal controller. It is composed of a interconnection a Stateflow diagram and a lookup table.

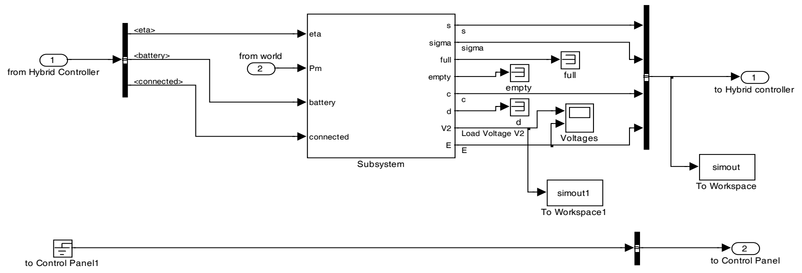


Figure 7.3-4: Power Plant

The power plant is designed such that it can receive input such as η, P_m , battery information. The output are the slip (s), state of charge (sigma), excitation voltage (*E*), terminal voltage (V_2) , battery information (empty, *c, d*).

Simulation

The following simulation is obtained with an optimal controller for a line fault of reduced admittance a = *0.375*. (*a = 1* represents no line fault). The initial battery state of charge $\sigma = 0.1$ and slip $s = 0$.

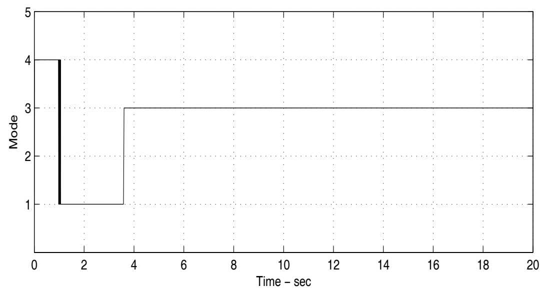


Figure 7.3-5:Mode vs time

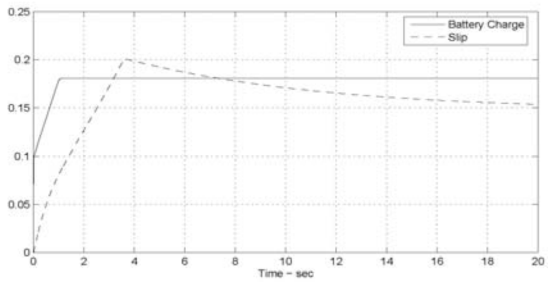


Figure 7.3-6: Battery State of charge vs time

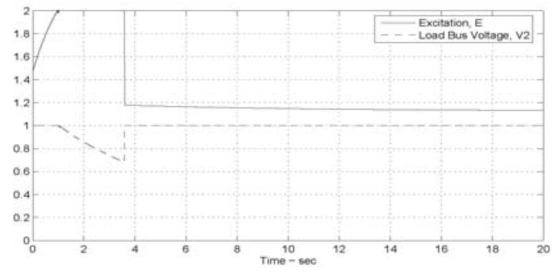


Figure 7.3-7: Excitation vs time

Simulation Results

The battery, initially merely charged, switched instantaneously to the charging mode, q_4 with increasing excitation *E*. At time 1 sec the excitation reached saturation and the mode q1 is activated. The voltage performance starts to grade. The terminal voltage, V_2 drops to 0.7 p.u. at time 1.8 sec. At time 1.8 sec the first load shedding occurs with an instantaneously switch to mode q_3 without dwelling in mode q_2. From time 1.8 to 20 sec the system stays in mode q_3, the battery 's state of charge remains at $\sigma = 0.18$ (i.e not being discharged). The slip starts to decrease due to the load shedding, the excitation drop to about 1.18 p.u. and as expected the terminal V_2 voltage is back to regulation immediately after load shedding.

7.4 Conclusion

The goal achieved in this chapter is to have successfully designed an optimal controller and used it in the emergency control of complex but simple system such a 3-bus power system modeled as hybrid system. In this chapter, we showed that the optimal control method is a viable method with regard to hybrid control structure. A multi-objective cost was setup to reflect the voltage regulation and various objectives such as battery charging or discharging and the fail mode and constraints for the mixed-integer dynamic programming are the integer programming formulas derived in subsection 5.3.4. The Mathematica code executes the mixed-integer dynamic program and generates an optimal controller. The optimal controller was further integrated to the SIMULINK/Stateflow representation of the 3-bus power system with UPS. A simulation of the controlled power plant was run and the results were conclusive.

CHAPTER 8:SHIPBOARD POWER SYSTEMS

8.1 Introduction

The adoption of the concept of Integrated Power Systems (IPS) brings new possibilities for safety critical system design of ship power systems with respect to a single or multiple faults due to internal failures or enemy attack. Survivability following a fault event is essential to the safe operation of a ship. In case of failure, the topological structure of the network should be reconfigured to best accommodate the post-fault configuration of the system. In most ship power systems, load is prioritized as non-vital and vital and sometimes a finer graduation is available. Non-vital loads are loads such as auxiliary service loads for which interruption for a short period of time does not impact the safety or mission of the ship. However for vital loads such as aircraft launch, weapon or communications systems, a short power interruption could be detrimental to safety or mission completion. Such loads are treated with high priority. In the design of future naval ships, a great deal of effort will be spent in identifying ways to maintain power delivery to vital loads during failure of various network components.

The goal of this research is to identify the best possible strategies for maintaining power supply to vital loads in emergency situations. As in CHAPTER 7:, optimal control will play a crucial role in the hybrid control design of reconfiguration strategies. This chapter will be organized as follows: In section 8.2 below we will provide a description of an Integrated Power System and its main functionalities, follow by a per-unit analysis. In section 8.3 we will layout the reconfiguration strategy for the Integrated Power System. In section 8.4 we will provide the dynamic model of the Integrated Power System.

8.2 The Integrated Power System for Shipboard

The Integrated Power Systems (IPS) as conceived for future ship is exemplified by the notional system for the DDG1000 destroyer as shown in Figure 8.2-1. It is a tightly coupled system composed of two parts, the starboard and the port distribution buses [50,51]. The starboard distribution system is a mirror image of the port distribution. They are interconnected through two transmission lines. There is a main turbine generator (MTG) on each port with 36 MW capacity. The generators are 3-phases synchronous generators powered by gas turbines. Also two Auxiliary Turbine generators (ATG) of capacity 4 MW each support the total generation, their primary role is to provide power to vital and non-vital loads. For the model simplification purpose, the main turbine generator and the auxiliary turbine generator are combined giving a 40 MW generation capacity. Vital loads are supplied via a DC bus in two ways. Either through an AC/DC converter designed to appear as unity power factor to the AC system or through a DC/DC converter whenever the battery supply is activated. An auctioneering diode guarantees the power supply from either the starboard or port side of the network.

In this power system the majority of the load is attributed to motor loads. Each generator primarily supplies power to the motor load bus through a transformer and a transmission line. Two alternate transmission lines (bus ties) from the starboard to the port distribution play the role of interconnecting the two subsystems. In the present work, we consider the propulsion motors to be induction motors.

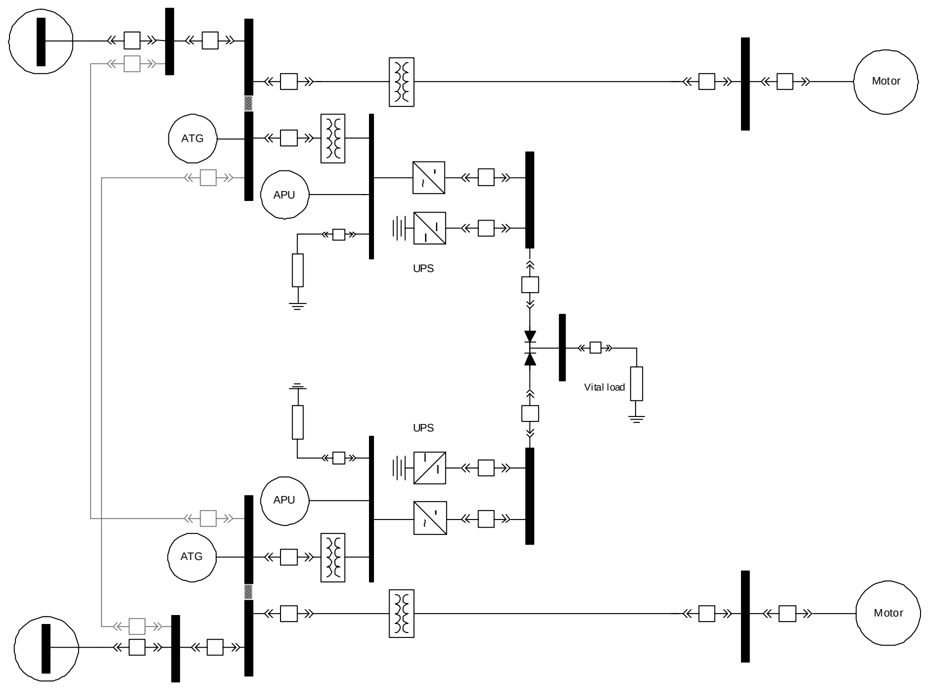


Figure 8.2-1: DDG 1000 distribution system abstraction

- Reconfiguration Strategies

Reconfiguration strategies consist of pre-specifying all modes of operation of the impaired system. When fault occurs the system will operate at reduced propulsion.

- Two types of faults will be investigated: Generator fault and Line fault.
- Non-vital load shedding will be performed at various modes.
- Vital load will be switched to battery mode when necessary.

The DDG 1000 Abstraction and Circuit Breakers

- **The DDG 1000 Abstraction**

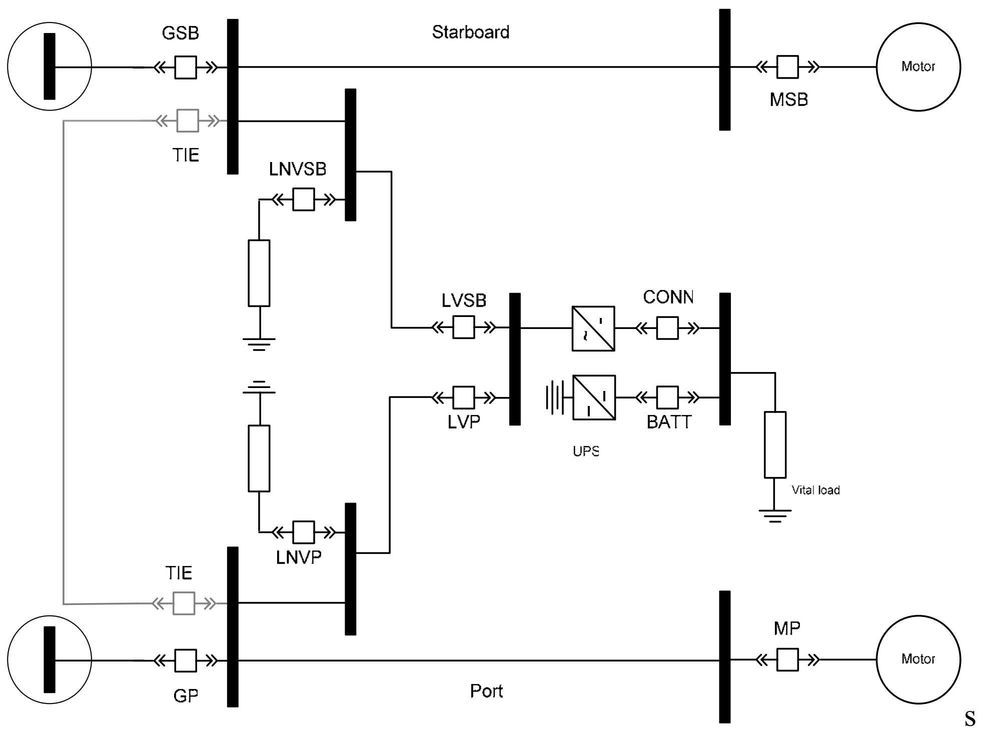


Figure 8.2-2.: Configuration of simulated system.

Figure 8.2-2 represents the abstraction of this system is shown in Figure 8.2-1.

- **DDG 1000 circuit breakers:**

The table below gives the nomenclature of the circuit breakers present in Figure 8.2-2.

Table 8-1:Circuit Breakers nomenclature

GSB	Generator Starboard
GP	Generator Port
MSB	Motor Starboard
MP	Motor Port
TIE	Tie Breaker
NVLSB	Non Vital Load on the Starboard
NVLP	Non Vital Load on the Port
VLSB	Vital Load on the Starboard
VLP	Vital on the Port
BATT	Battery
CONN	Connected

The Admittance Matrix:

Based on the network configuration in Figure 8.2-2 we can derive the following admittance matrix:

$$\begin{bmatrix} -ib_{12} & ib_{12} & 0 & 0 & 0 & 0 & 0 \\ ib_{12} & -i(b_{21}+b_{23}+b_{24}+b_{27}) & ib_{23} & ib_{24} & 0 & 0 & ib_{27} \\ 0 & ib_{23} & -i(b_{23}+b_{mvl1})+g_{mvl1} & 0 & 0 & 0 & 0 \\ 0 & ib_{24} & 0 & c_1-i(d_1+b_{24}) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & c_2-i(d_2+b_{57}) & 0 & ib_{57} \\ 0 & 0 & 0 & 0 & 0 & -i(b_{67}+b_{mvl2})+g_{mvl2} & ib_{67} \\ 0 & ib_{27} & 0 & 0 & ib_{57} & ib_{67} & -i(b_{27}+b_{57}+b_{67}) \end{bmatrix}$$

(8.1)

8.3 Reconfiguration of an Integrated Power System

Reconfiguration of a system is the change from a current topological configuration to a new configuration in order to improve a performance towards an intended functional mode. Reconfigurability is a desirable property of modern safety critical system. It is the ability of a system to alter its structure so as to satisfy a performance specification.

8.3.1 Reconfiguration Strategy

Let us describe the reconfiguration strategy by first presenting the reconfiguration transition diagram and various strategies.

Reconfiguration Transition Diagram

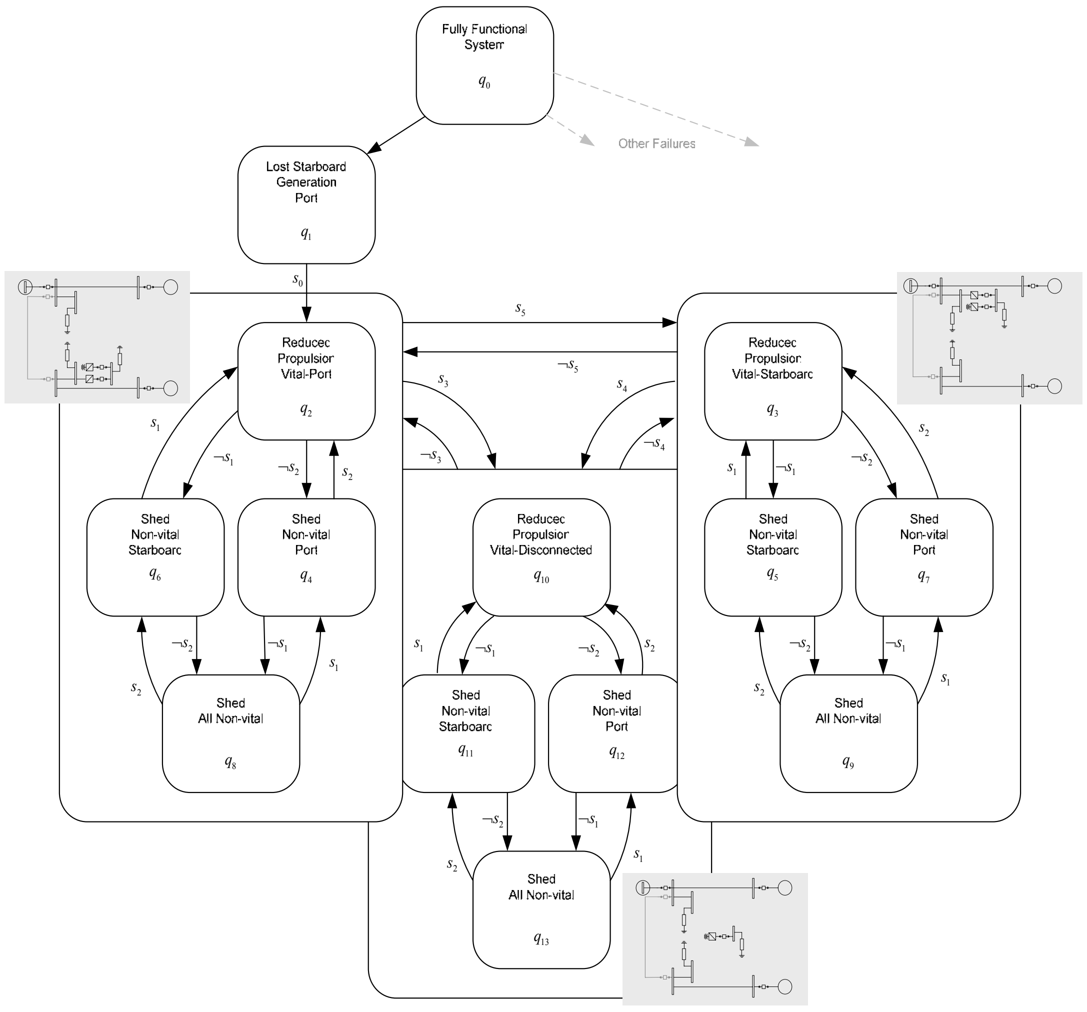


Figure 8.3-1: Discrete state transition diagram illustrates discrete actions.

The system is assumed to be operating at full capacity in mode q_0 of Figure 8.3-1. We consider a complete loss of main turbine generator and auxiliary turbine generator connected to bus 7. The following reconfigurations are possible:

- vital loads
- remain connected to bus 3

- switched to bus 6
- switched to battery
- non-vital loads
- non-vital load connected to bus 3 can remain connected or dropped completely
- non-vital load connected to bus 6 can remain connected or dropped completely
- propulsion motors
  - propulsion motor 1, connected to bus 4, can remain at full power or reduced to 50% power, or dropped completely.
  - propulsion motor 2, connected to bus 5, can remain at full power or reduced to 50% power, or dropped completely.

Discrete states corresponding to admissible reconfigurations can now be defined and system models developed for each discrete state. In addition, a transition structure can be defined that expresses allowable transitions between the discrete states. While it is always possible to allow transitions from every discrete state to every other discrete state, the specification of a transition structure has many benefits. A specification allows us to impose constraints on the reconfiguration process and to eliminate obviously unsuitable transitions. The severity of the failure obviously requires that both motors be dropped to 50% or that motor 1 is dropped completely. It is preferable that both motors remain in operation so the first transition is clearly to a state with both motors at 50%. Figure 8.3-1 shows one possible transition specification.

8.3.2 Reconfiguration Transition Diagram and Logic Specification

The table bellow describes the status (0/1) of the circuit breakers which enables or disables all the power system components in the DDG 1000.

Table 8-2: Circuit Breakers

Breaker	**N.**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**
GSB	1	1	1	1	1	1	1	1	1	1	1	1	1	1
GP	1	0	0	0	0	0	0	0	0	0	0	0	0	0
MSB	1	1	1	0	1	0	1	0	1	0	0	1	1	1
MP	1	1	0	1	0	1	0	1	0	1	0	0	0	0
TIE	1	1	1	1	1	1	1	1	1	1	1	1	1	1
NVLSB	1	1	1	1	1	0	0	1	0	0	1	0	1	0
NVLP	1	1	1	1	0	1	1	0	0	0	1	1	0	0
VLSB	1	1	0	1	0	1	0	1	0	1	0	0	0	0
VLP	1	1	1	0	1	0	1	0	1	0	0	0	0	0
BATT	--	0	0	0	0	0	0	0	0	0	1	1	1	1
CONN	--	0	0	0	0	0	0	0	0	0	1	1	1	1

Let us describe a few mode of the transition configuration.

1) In mode q_0 (fully functional system) all circuit are normally closed except the battery and the connection which are left unspecified.

2) In mode q_1, the circuit breakers on the port generation, the battery and connection are switched off.

3) In mode q_2, the circuit breakers on the port generation, propulsion on the port side, the vital load on the starboard side, the battery and connection are switched off.

Logic specification

Based on the discrete states transition diagram in Figure 8.3-1 there are:

- 12 discrete states, $\{q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9, q_{10}, q_{11}, q_{12}, q_{13}\}$,
- 5 switches , $\{s_1, s_2, s_3, s_4, s_5\}$

The following logic specification is formulated.

$$
\begin{aligned}
&spec1 = exactly[1,\ \{q_2,\ q_3,\ q_4,\ q_5,\ q_6,\ q_7,\ q_8,\ q_9,\ q_{10},\ q_{11},\ q_{12},\ q_{13}\}] \wedge \\
&\qquad exactly[1,\ \{q_2^+,\ q_3^+,\ q_4^+,\ q_5^+,\ q_6^+,\ q_7^+,\ q_8^+,\ q_9^+,\ q_{10}^+,\ q_{11}^+,\ q_{12}^+,\ q_{13}^+\}] \wedge \\
&((q_2 \wedge (\neg s_1)) \Rightarrow q_6^+) \wedge ((q_2 \wedge (\neg s_2)) \Rightarrow q_4^+) \wedge ((q_2 \wedge s_3) \Rightarrow q_{10}^+) \wedge ((q_2 \wedge s_5) \Rightarrow q_3^+) \wedge \\
&\qquad ((q_2 \wedge (\neg((\neg s_1) \vee (\neg s_2) \vee s_3 \vee s_5)) \Rightarrow q_2^+) \wedge \\
&((q_3 \wedge (\neg s_1)) \Rightarrow q_5^+) \wedge ((q_3 \wedge (\neg s_2)) \Rightarrow q_7^+) \wedge ((q_3 \wedge \neg s_5) \Rightarrow q_2^+) \wedge ((q_3 \wedge s_4) \Rightarrow q_{10}^+) \wedge \\
&\qquad ((q_3 \wedge (\neg((\neg s_1) \vee (\neg s_2) \vee (\neg s_5) \vee s_4)) \Rightarrow q_3^+) \wedge \\
&((q_4 \wedge s_2)) \Rightarrow q_2^+) \wedge ((q_4 \wedge (\neg s_1)) \Rightarrow q_8^+) \wedge ((q_4 \wedge s_3) \Rightarrow q_{12}^+) \wedge ((q_4 \wedge s_5) \Rightarrow q_7^+) \wedge \\
&\qquad ((q_4 \wedge (\neg((\neg s_1) \vee (s_2 \vee s_3 \vee s_5)) \Rightarrow q_4^+) \wedge \\
&((q_5 \wedge s_1) \Rightarrow q_3^+) \wedge ((q_5 \wedge (\neg s_2)) \Rightarrow q_9^+) \wedge ((q_5 \wedge (\neg s_5)) \Rightarrow q_6^+) \wedge ((q_5 \wedge s_4) \Rightarrow q_{11}^+) \wedge \\
&\qquad ((q_5 \wedge \neg(s_1 \vee (\neg s_2) \vee s_4 \vee (\neg s_5))) \Rightarrow q_5^+) \wedge \\
&((q_6 \wedge s_1) \Rightarrow q_2^+) \wedge ((q_6 \wedge (\neg s_2)) \Rightarrow q_8^+) \wedge ((q_6 \wedge s_3) \Rightarrow q_{11}^+) \wedge ((q_6 \wedge s_5) \Rightarrow q_5^+) \wedge \\
&\qquad ((q_6 \wedge \neg(s_1 \vee \neg s_2 \vee s_4 \vee s_5)) \Rightarrow q_6^+) \wedge \\
&((q_7 \wedge s_2) \Rightarrow q_3^+) \wedge ((q_7 \wedge (\neg s_1)) \Rightarrow q_9^+) \wedge ((q_7 \wedge (\neg s_5)) \Rightarrow q_4^+) \wedge ((q_7 \wedge s_4) \Rightarrow q_{12}^+) \wedge \\
&\qquad ((q_7 \wedge \neg(s_2 \vee \neg s_1 \vee s_4 \vee \neg s_5)) \Rightarrow q_{12}^+) \wedge \\
&((q_8 \wedge s_2) \Rightarrow q_6^+) \wedge ((q_8 \wedge s_1) \Rightarrow q_4^+) \wedge ((q_8 \wedge s_5) \Rightarrow q_9^+) \wedge ((q_8 \wedge s_3) \Rightarrow q_{13}^+) \wedge \\
&\qquad ((q_8 \wedge \neg(s_1 \vee s_2 \vee s_3 \vee s_5)) \Rightarrow q_8^+) \wedge \\
&((q_9 \wedge s_2) \Rightarrow q_5^+) \wedge ((q_9 \wedge s_1) \Rightarrow q_7^+) \wedge ((q_9 \wedge s_4) \Rightarrow q_{13}^+) \wedge ((q_9 \wedge (\neg s_5)) \Rightarrow q_8^+) \wedge \\
&\qquad ((q_9 \wedge \neg(s_1 \vee s_2 \vee s_4 \vee \neg s_5)) \Rightarrow q_9^+) \wedge \\
&((q_{10} \wedge (\neg s_1)) \Rightarrow q_{11}^+) \wedge ((q_{10} \wedge (\neg s_2)) \Rightarrow q_{12}^+) \wedge ((q_{10} \wedge (\neg s_4)) \Rightarrow q_3^+) \wedge ((q_{10} \wedge (\neg s_3)) \Rightarrow q_2^+) \wedge \\
&\qquad ((q_{10} \wedge \neg(\neg s_1 \vee \neg s_2 \vee \neg s_4 \vee \neg s_4)) \Rightarrow q_{10}^+) \wedge \\
&((q_{11} \wedge (\neg s_4)) \Rightarrow q_5^+) \wedge ((q_{11} \wedge (\neg s_3)) \Rightarrow q_6^+) \wedge ((q_{11} \wedge s_1) \Rightarrow q_{10}^+) \wedge ((q_{11} \wedge (\neg s_2)) \Rightarrow q_{13}^+) \wedge \\
&\qquad ((q_{11} \wedge \neg(s_1 \vee \neg s_2 \vee \neg s_4 \vee \neg s_4)) \Rightarrow q_{11}^+) \wedge \\
&((q_{12} \wedge (\neg s_4)) \Rightarrow q_7^+) \wedge ((q_{12} \wedge (\neg s_3)) \Rightarrow q_4^+) \wedge ((q_{12} \wedge (\neg s_1)) \Rightarrow q_{13}^+) \wedge ((q_{12} \wedge s_3) \Rightarrow q_{10}^+) \wedge \\
&\qquad ((q_{12} \wedge \neg(\neg s_1 \vee s_2 \vee \neg s_3 \vee \neg s_4)) \Rightarrow q_{12}^+) \wedge \\
&((q_{13} \wedge (\neg s_4)) \Rightarrow q_9^+) \wedge ((q_{13} \wedge (\neg s_3)) \Rightarrow q_8^+) \wedge ((q_{13} \wedge s_2) \Rightarrow q_{11}^+) \wedge ((q_{13} \wedge s_1) \Rightarrow q_{12}^+) \wedge \\
&\qquad ((q_{13} \wedge \neg(s_1 \vee s_2 \vee \neg s_3 \vee \neg s_4)) \Rightarrow q_{13}^+)
\end{aligned}
\tag{8.2}
$$

For the same reason given in 5.3.4 we set up the logic constraint.

For example in state q_2 the switch s_4 is always set to zero (i.e. disabled) and at most one of $\{\neg s_1, \neg s_2, s_3, s_5\}$ switches is can be enabled. Similarly the Control Logic specifications are obtained for the others modes.

Control Logic:

$$
\begin{aligned}
&\mathit{Control\ Logic} = \{\\
&q_2 \Rightarrow (\neg s_4 \wedge atmost[1,\{\neg s_1,\neg s_2,s_3,s_5\}]))) \wedge\\
&q_3 \Rightarrow (\neg s_3 \wedge atmost[1,\{\neg s_1,\neg s_2,s_4,\neg s_5\}]))) \wedge\\
&q_4 \Rightarrow (\neg s_4 \wedge atmost[1,\{\neg s_1,s_2,s_3,s_5\}]))) \wedge\\
&q_5 \Rightarrow (\neg s_3 \wedge atmost[1,\{s_1,\neg s_2,s_4,\neg s_5\}]))) \wedge\\
&q_6 \Rightarrow (\neg s_4 \wedge atmost[1,\{s_1,\neg s_2,s_3,s_5\}]))) \wedge\\
&q_7 \Rightarrow (\neg s_3 \wedge atmost[1,\{\neg s_1,s_2,s_4,\neg s_5\}]))) \wedge\\
&q_8 \Rightarrow (\neg s_4 \wedge atmost[1,\{s_1,s_2,s_3,s_5\}]))) \wedge\\
&q_9 \Rightarrow (\neg s_3 \wedge atmost[1,\{s_1,s_2,s_4,\neg s_5\}]))) \wedge\\
&q_{10} \Rightarrow (\neg s_5 \wedge atmost[1,\{\neg s_1,\neg s_2,\neg s_3,\neg s_4\}]))) \wedge\\
&q_{11} \Rightarrow (\neg s_5 \wedge atmost[1,\{s_1,\neg s_2,\neg s_3,\neg s_4\}]))) \wedge\\
&q_{12} \Rightarrow (\neg s_5 \wedge atmost[1,\{\neg s_1,s_2,\neg s_3,\neg s_4\}]))) \wedge\\
&q_{13} \Rightarrow (\neg s_5 \wedge atmost[1,\{s_1,s_2,\neg s_3,\neg s_4\}])))\\
&\}
\end{aligned}
\tag{8.3}
$$

8.4 Dynamic Models and Network Equations

The dynamics of the integrated power system as represented for the purpose of the deriving the controller is composed of the dynamics of the two motors and the dynamics of the battery. Unlike the 3-bus power system where the network equation was easily reducible to a characteristic equation, in a relatively large power such as the IPS, the elimination method is not applicable, therefore we are compelled to use a direct numerical analysis of the differential-Algebraic system.

8.4.1 Motor and Battery Dynamics

Motor Dynamics

The model of each motor are composed of a constant impedance load with a slow varying parameter which is the motor slip, aggregated with a constant impedance non vital load. The model of the motor is as obtained from the equivalent circuit in the 3-bus example shown in (5.16).

The slip dynamics of the two motors are defined as:

Motor 1: $s_{m1} = \frac{\omega_0 - \omega_m}{\omega_0}, \dot{s}_{m1} = \frac{1}{I_m \omega_0^2}\left(P_{m2} - P_e\right) = \frac{1}{I_m \omega_0^2}\left(P_{m1} - V_4^2 \frac{R_r s_{m1}\left(1 - s_{m1}\right)}{R_r^2 + s_{m1}{}^2 X_r^2}\right)$ (8.4)

Motor 2: $s_{m2} = \frac{\omega_0 - \omega_m}{\omega_0}, \dot{s}_{m2} = \frac{1}{I_m \omega_0^2}\left(P_{m2} - P_e\right) = \frac{1}{I_m \omega_0^2}\left(P_{m2} - V_5^2 \frac{R_r s_{m2}\left(1 - s_{m2}\right)}{R_r^2 + s_{m2}{}^2 X_r^2}\right)$ (8.5)

Modeling of the Battery:

The modeling of the battery is similar to what was described previously except that there are no charging mode, and the nominal and the discharging mode are defined according to the transition state diagram.

2) Nominal mode (UPS inactive, modes: $q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9$)

$$\frac{d\sigma}{dt} = 0 \tag{8.6}$$

1) Battery Discharging (UPS active, modes: $q_{10}, q_{11}, q_{12}, q_{13}$)

$$\frac{d\sigma}{dt} = \frac{i_c}{C} \tag{8.7}$$

where $i = -V_3 / R_v,\ V_3 = 1$

Summary:

$$\frac{d\sigma}{dt} = \delta_n \times 0 + \delta_d \times \frac{i}{C} \tag{8.8}$$

Where $\delta_n = \delta_{q_2^+} + \delta_{q_3^+} + \delta_{q_4^+} + \delta_{q_2^+} + \delta_{q_5^+} + \delta_{q_6^+} + \delta_{q_7^+} + \delta_{q_8^+} + \delta_{q_9^+},\ \delta_d = \delta_{q_{10}^+} + \delta_{q_{11}^+} + \delta_{q_{12}^+} + \delta_{q_{13}^+}$

This expressed the fact that in modes $q_2^+, q_3^+, q_4^+, q_5^+, q_6^+, q_7^+, q_8^+, q_9^+$ the battery is inactive, and active in modes $q_{10}^+, q_{11}^+, q_{12}^+, q_{13}^+$.

The descretized form of the battery:

The two modes can be combined in discrete-time as:

Discrete-Time:

$$\sigma_{k+1} = \delta_n f_n(\sigma_k, i) + \delta_d f_d(\sigma_k, i) \tag{8.9}$$

where $\delta_n = \delta_{q_2^+} + \delta_{q_3^+} + \delta_{q_4^+} + \delta_{q_2^+} + \delta_{q_5^+} + \delta_{q_6^+} + \delta_{q_7^+} + \delta_{q_8^+} + \delta_{q_9^+},\ \delta_d = \delta_{q_{10}^+} + \delta_{q_{11}^+} + \delta_{q_{12}^+} + \delta_{q_{13}^+}$

By this approach the change in the battery dynamics is taken into consideration with the change in the parameters from configuration changes.

Per Unit Analysis and Network Equations

When different bases, and phases (i.e voltage , current, power) are used for power system components such generators, transformers, transmission lines it is often quiet convenient to pick one component power an voltage bases and perform a per unit normalization other power and voltage levels. A great advantage of the per unit values is that no computations are required to refer an impedance from one side of a transformer to the other. The goal of the normalization of the DDG 1000 Integrated power system is to reduced to scale the power system so as to give flexibility in computation and to prevent any numerical problem during computation. The detail of the per-unit calculation is shown in appendix A.

The goal of the per-unit normalization is to simplify computations. We will use the per unit normalization on the impedance diagram of the DDG 1000 network model.

- **The Admittance Matrix: Y Bus**

The admittance matrix is essential in deriving the network equations.

$$\begin{bmatrix} -ib_{12} & ib_{12} & 0 & 0 & 0 & 0 & 0 \\ ib_{12} & -i(b_{21}+b_{23}+b_{24}+b_{27}) & ib_{23} & ib_{24} & 0 & 0 & ib_{27} \\ 0 & ib_{23} & -i(b_{23}+b_{mvl1})+g_{mvl1} & 0 & 0 & 0 & 0 \\ 0 & ib_{24} & 0 & -c_1-i(d_1+b_{24}) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -c_2-i(d_2+b_{57}) & 0 & ib_{57} \\ 0 & 0 & 0 & 0 & 0 & -i(b_{67}+b_{mvl2})+g_{mvl2} & ib_{67} \\ 0 & ib_{27} & 0 & 0 & ib_{57} & ib_{67} & -i(b_{27}+b_{57}+b_{67}) \end{bmatrix}$$

(8.10)

Where:

$$c_1 = 0.5 + \frac{0.25sl1}{0.0625 + 0.015625sl1^2},\ d_1 = \frac{0.125sl1}{0.0625 + 0.015625sl1^2}$$

$$c_2 = 0.5 + \frac{0.25sl2}{0.0625 + 0.015625sl2^2},\ d_2 = \frac{0.125sl2}{0.0625 + 0.015625sl2^2}$$

Network Equations:

The swing bus is taken as bus 1, therefore we ca make the following transformations

$$\delta_1 - \delta_2 = \theta_2,\ \delta_2 - \delta_3 = \theta_{23},\ \delta_2 - \delta_4 = \theta_{24},\ \delta_2 - \delta_7 = \theta_{27},\ \delta_5 - \delta_7 = \theta_{57},\ \delta_6 - \delta_7 = \theta_{67}$$

$$P_2 = 0,\ Q_2 = 0,\ P_3 = -P_{v_1},\ Q_3 = 0,\ P_4 = 0,\ Q_2 = 0,\ P_5 = 0,\ Q_5 = 0,$$

$$P_6 = -P_{v_2},\ Q_2 = 0,\ P_7 = 0,\ Q_7 = 0$$

The real and the reactive power equation is obtained using the formulas in (5.2).

The equation $P_1 - b_{12}\sin(\theta_2)V_1V_2$ at bus 1 is eliminated from the network equations.

Real Equations:

$$0 = b_{12} \sin(\theta_2) V_1 V_2 - b_{23} \sin(\theta_{23}) V_2 V_3 - b_{24} \sin(\theta_{24}) V_2 V_4 - b_{27} \sin(\theta_{27}) V_2 V_7$$

$$0 = -P_{v_1} + b_{23} \sin(\theta_{23}) V_2 V_3 - g_{33} V_3^2$$

$$0 = b_{24} \sin(\theta_{24}) V_2 V_4 - g_{44} V_4^2$$

$$0 = -b_{57} \sin(\theta_{57}) V_5 V_7 - g_{55} V_5^2 \tag{8.11}$$

$$0 = -P_{v_2} - b_{67} \sin(\theta_{67}) V_6 V_7 - g_{66} V_6^2$$

$$0 = b_{27} \sin(\theta_{27}) V_2 V_7 + b_{57} \sin(\theta_{57}) V_5 V_7 + b_{67} \sin(\theta_{67}) V_6 V_7$$

Reactive Equations:

$$0 = b_{12} \cos(\theta_2) V_1 V_2 - b_{22} V_2^2 + b_{23} \cos(\theta_{23}) V_2 V_3 + b_{24} \cos(\theta_{24}) V_2 V_4 + b_{27} \cos(\theta_{27}) V_2 V_7$$

$$0 = b_{23} \cos(\theta_{23}) V_2 V_3 - b_{33} V_3^2$$

$$0 = b_{24} \cos(\theta_{24}) V_2 V_4 - b_{44} V_4^2$$

$$0 = b_{57} \cos(\theta_{57}) V_5 V_7 - b_{55} V_5^2 \tag{8.12}$$

$$0 = b_{67} \cos(\theta_{67}) V_6 V_7 - b_{66} V_6^2$$

$$0 = b_{27} \cos(\theta_{27}) V_2 V_7 + b_{57} \cos(\theta_{57}) V_5 V_7 + b_{67} \cos(\theta_{67}) V_6 V_7 - b_{77} V_7^2$$

Per Unit Admittance matrix:

$$\begin{bmatrix} -0.21+21.2i & 0.21-21.29i & 0 & 0 & 0 & 0 & 0 \\ 0.21-21.29i & -0.72+91.6i & 0.31-12.49i & 0.20-9.91i & 0 & 0 & -47.97i \\ 0 & 0.31-12.49i & -0.36+12.53 & 0 & 0 & 0 & 0 \\ 0 & 0.20-9.91i & 0 & -0.20-c_1-i(d_1-9.91) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -0.20-c_2-i(d_2-9.91) & 0 & 0.20-9.91i \\ 0 & 0 & 0 & 0 & 0 & -0.36+12.53i & 0.31-12.49i \\ 0 & -47.97i & 0 & 0 & 0.20-9.91i & 0.31-12.49i & -0.51+70.37i \end{bmatrix}$$

(8.13)

8.4.2 Differential-Algebraic Equations Models

Differential-Algebraic-Equations (DAEs) form the essential mathematical model for power systems. So it is not surprising that a great deal of attention has been paid to the development of computational methods for solving them. Nevertheless, computing trajectories remains problematic and analysts often need to experiment with a variety of methods and parameters before obtaining satisfactory results. When the system involves switching or mode transitions the difficulty is magnified many times, and, to this date, very little thought has been given to hybrid systems with continuous dynamics described by DAEs.

In section 5.3.2 we were able to solve the algebraic equations (the network equations) using elimination methods. However, this is only feasible in the case of very small networks or systems with special structure. For more complex systems we need to compute approximate solutions, possibly using numerical computations.

In our situation we need compute discrete time trajectories for hybrid-DAE systems, both forward in time (for the control problem) and backward in time (for the estimation problem). The distinguishing feature of hybrid systems vis-à-vis systems with only continuous dynamics is that not all of the dependent variables need be continuous in time. Ordinarily, there will be discontinuities at time of discrete state transitions. In the following paragraphs we describe a discrete time model for the forward case. The backward model is obtained in a similar way.

8.4.2.1 Problem Definition

Consider the semi-explicit Differential-Algebraic-Equation (DAE)

$$\begin{aligned} \dot{x} &= f\left(x, y, u\right) \\ 0 &= g\left(x, y, u\right) \end{aligned} \tag{14}$$

where time $t \in R^{+}$, $u \in R^{m}$ is an external input, the state is composed of $x \in R^{n}, y \in R^{p}$ and the functions $f : R^{n+p+m} \to R^{m}, g : R^{n+p+m} \to R^{p}$ define the evolution of the state. We will assume that the control $u\left(t\right)$ is piecewise continuous and the state $x\left(t\right)$ is continuous. Our goal is to show that under appropriate conditions the system (14) can be approximately described by a discrete time system

$$\begin{aligned} \hat{x}_{k+1} &= F_1\left(\hat{x}_k, \hat{y}_k, u_k\right) \\ \hat{y}_{k+1} &= F_2\left(\hat{x}_k, \hat{y}_k, u_k\right) \end{aligned} \tag{15}$$

Where $\hat{x}_k, \hat{y}_k$ are approximations to $x(t_k), y(t_k)$, respectively.

8.4.2.2 Differential Algebraic-Difference – Hybrid Case

We will derive a discrete time representation where $t_k = kh, k = 0,1,2,\ldots$ and $h > 0$ is the time increment. It is assumed that $u\left(t\right) = u_k$, a constant, for $t \in [t_k, t_{k+1})$. A basic assumption is that mode changes can occur only at the time instants, t_k. In other words, for the interval $t \in [t_k, t_{k+1})$ we the system is described by

$$\begin{aligned} \dot{x} &= f_{i(t_k)}(x, y, u) \\ 0 &= g_{i(t_k)}(x, y, u) \end{aligned} \tag{16}$$

where $i \in I_{\text{mode}}$, the mode index set. If we disallow resets during mode transitions, then it is reasonable to assume that $x(t)$ is continuous. On the other hand, $y(t)$ cannot be expected to be continuous across a mode transition. This implies that at each discrete time instant t_k it is necessary to compute $y_k^+ = y(t_k^+)$ from

$$0 = g_{i(t_k)}\left(x_k, y_k^+, u\left(t_k^+\right)\right) = g_{i(t_k)}\left(x_k, y_k^+, u_k\right)$$

We will drop the subscript $i(t_k)$which is to be understood in the following expressions. Thus, y_k^+ is obtained from:

$$0 = g\left(x_k, y_k^+, u_k\right) \tag{17}$$

Now, we use a backward difference formula, in particular the trapezoidal formula, to obtain from (14):

$$\begin{aligned} x_{k+1} &= x_k + \tfrac{1}{2}h\left(f\left(x_k, y_k^+, u_k\right) + f\left(x_{k+1}, y_{k+1}, u_k\right)\right) \\ 0 &= g\left(x_{k+1}, y_{k+1}, u_k\right) \end{aligned} \tag{18}$$

With x_k, y_k^+, u_{k+1} known, it is necessary to solve (18) for x_{k+1}, y_{k+1}. To do this Taylor expand $f\left(x_{k+1}, y_{k+1}, u_k\right)$ and $g\left(x_{k+1}, y_{k+1}, u_k\right)$ about an initial estimate x_{k+1}^0, y_{k+1}^0 to obtain

$$\begin{aligned}
x_{k+1} &= x_k + \tfrac{h}{2}\left(f\left(x_k, y_k^+, u_k\right) + f\left(x_{k+1}^0, y_{k+1}^0, u_k\right)\right) \\
&\quad + \tfrac{h}{2}\frac{\partial f}{\partial x}\left(x_{k+1}^0, y_{k+1}^0, u_k\right)\left(x_{k+1} - x_{k+1}^0\right) \\
&\quad + \tfrac{h}{2}\frac{\partial f}{\partial y}\left(x_{k+1}^0, y_{k+1}^0, u_k\right)\left(y_{k+1} - y_{k+1}^0\right) + h.o.t \\
0 &= g\left(x_{k+1}^0, y_{k+1}^0, u_k\right) + \frac{\partial g}{\partial x}\left(x_{k+1}^0, y_{k+1}^0, u_k\right)\left(x_{k+1} - x_{k+1}^0\right) \\
&\quad + \frac{\partial g}{\partial y}\left(x_{k+1}^0, y_{k+1}^0, u_k\right)\left(y_{k+1} - y_{k+1}^0\right) + h.o.t
\end{aligned}$$

By neglecting higher order terms, these equations can be approximately solved for the unknowns x_{k+1}, y_{k+1} using the linear equations:

$$\begin{gathered}
\begin{bmatrix} I - \tfrac{h}{2}\frac{\partial f}{\partial x}\left(x_{k+1}^0, y_{k+1}^0, u_k\right) & -\tfrac{h}{2}\frac{\partial f}{\partial y}\left(x_{k+1}^0, y_{k+1}^0, u_k\right) \\ -\frac{\partial g}{\partial x}\left(x_{k+1}^0, y_{k+1}^0, u_k\right) & -\frac{\partial g}{\partial y}\left(x_{k+1}^0, y_{k+1}^0, u_k\right) \end{bmatrix} \begin{bmatrix} \left(x_{k+1} - x_{k+1}^0\right) \\ \left(y_{k+1} - y_{k+1}^0\right) \end{bmatrix} = \\
\begin{bmatrix} \left(x_k - x_{k+1}^0\right) + \tfrac{h}{2}\left(f\left(x_k, y_k^+, u_k\right) + f\left(x_{k+1}^0, y_{k+1}^0, u_k\right)\right) \\ g\left(x_{k+1}^0, y_{k+1}^0, u_k\right) \end{bmatrix}
\end{gathered} \tag{19}$$

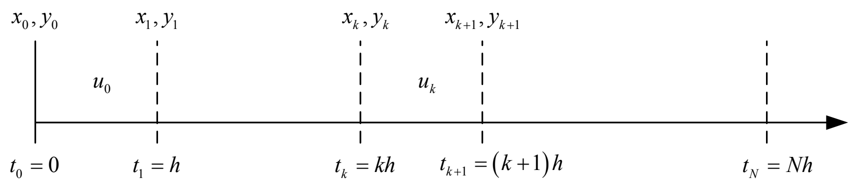
The approximation can be improved by replacing the initial values with the solution of (19), $x_{k+1}^0 \leftarrow x_{k+1}$, $y_{k+1}^0 \leftarrow y_{k+1}$ and resolving (19). Continuing recursively in this way is, of course, the Newton-Raphson method for finding solutions of (18).

If h is small and $x(t)$ continuous, it is common to take $x_{k+1}^0 = x_k, y_{k+1}^0 = y_k^+$, so that (19) becomes

$$\begin{bmatrix} I-\frac{h}{2}\frac{\partial f}{\partial x}\left(x_k,y_k^+,u_k\right) & -\frac{h}{2}\frac{\partial f}{\partial y}\left(x_k,y_k^+,u_k\right) \\ -\frac{\partial g}{\partial x}\left(x_k,y_k^+,u_k\right) & -\frac{\partial g}{\partial y}\left(x_k,y_k^+,u_k\right) \end{bmatrix} \begin{bmatrix} \left(x_{k+1}-x_k\right) \\ \left(y_{k+1}-y_k^+\right) \end{bmatrix} = \begin{bmatrix} hf\left(x_k,y_k^+,u_k\right) \\ g\left(x_k,y_k^+,u_k\right) \end{bmatrix} \tag{20}$$

Note that if a mode transition takes place at t_k into a mode requiring a state reset to say x^*, then we would take $x_{k+1}^0 = x^*$. In summary the discrete time model is given by (17) and (19) or (20). Notice that by making appropriate correspondences, the model (17) and (20) does take the form of (15).

Ordinarily, the model is integrated by solving (17) for y_k^+ using a Newton-Raphson method, with termination dependent on an error check, or simply a fixed number of iterations. The latter is typical for DAE solvers. Then the linear Equation (20) is solved for x_{k+1}, y_{k+1}. If we choose to implement the computations in the most flexible manner, we should permit solution of (17) with any specified number of iterations, n_1 and then x_{k+1}, y_{k+1} should be obtained from (19) with any specified number of iterations, n_2.



Algorithm 1:

Given x_k, y_k, u_k and n_1, n_2 and $f, f_x, f_y; g, g_x, g_y$

Determine x_{k+1}, y_{k+1}

1. Compute y_k^+ from $g\left(x_k, y_k^+, u_k\right) = 0$ using n_1 iterations of Newton's method, starting with y_k

   a. Newton$\left(g\left(x_k, y, u_k\right), g_y\left(x_k, y, u_k\right), y, y_k^+, n_1\right)$

2. Compute x_{k+1}, y_{k+1} from

$$\begin{aligned} x_{k+1} - x_k - \tfrac{1}{2}h\left(f\left(x_k, y_k^+, u_k\right) + f\left(x_{k+1}, y_{k+1}, u_k\right)\right) &= 0 \\ g\left(x_{k+1}, y_{k+1}, u_k\right) &= 0 \end{aligned}$$

   using n_2 iterations of Newton's method starting with x_k, y_k^+

   a. Define $z = (x, y)$,

$F(z) = \left(x - x_k - \tfrac{1}{2}h\left(f\left(x_k, y_k^+, u_k\right) + f\left(x, y, u_k\right)\right), g\left(x, y, u_k\right)\right)$ and

$$F_z = \begin{bmatrix} I - \dfrac{h}{2} f_x\left(x, y, u_k\right) & -\dfrac{h}{2} f_y\left(x, y, u_k\right) \\ g_x\left(x, y, u_k\right) & g_y\left(x, y, u_k\right) \end{bmatrix}$$

   b. Newton$\left(F\left(x, y, u_k\right), F_z\left(x, y, u_k\right), (x, y), \left(x_k, y_k\right), n_2\right)$

The idea is that for any given power system in the form of (14) we will automatically create code for implementing (17) and (19), with n_1, n_2 as parameters. The computational routines will be targeted for numerical computation in Mathematica and SIMULINK. We developed Mathematica code to implement the following functions..

Function Newton:

Newton$\left(F(x), F_x(x), x, x_0, n\right)$

Given: function $F(x)$, Jacobian $F_x(x)$, initial estimate x_0 and number of iterations n

```
1. set $x_I = x_0$, $k = 1$
2. While $k \le n$
    a. solve for $x$, $F(x_I) + F_x(x_I)(x - x_I) = 0$ (use *Mathematica* function Solve)
    b. set $x_I = x$
    c. set $k = k + 1$
```

Function DAEDiscrete:

DAEDiscrete$\left(f, g, x, y, x_0, y_0, u, n_1, n_2, h\right)$

1. Given:
    a. Functions f, g
    b. Argument lists x, y, u, x_0, y_0
    c. Iteration integers n_1, n_2
    d. Time increment h

2. Implement Algorithm 1 to compute $F\left(z,u\right)$ such that

$$z_{i+1} = F\left(z_k, u_k\right), \quad z_k = \left(\hat{x}_k, \hat{y}_k\right) \tag{21}$$

Utilization:

We will provide in this paragraph some guidelines about how to select some of the parameters of DAE Discrete function described above.

If x_0, y_0, u are symbols, then (21) gives us the discrete time approximation suggested in (15). This is the primary intent of this model. Ordinarily, we take h small and expect that a relatively small number of iterations, i.e., n_1, n_2 are small. Once F is computed, it can be simplified using standard *Mathematica* functions. In view of the expected complexity of the expressions, it may also be desirable to truncate them, retaining only low orders of the small parameter h.

8.5 Conclusion

The concept of Integrated Power System was proposed by the Navy as replacement for the conventional power system existing in most naval ships, which have the disadvantage to be segmented, i.e. generator supplying power to lone propulsion and service loads. The integrated Power system has many advantages such reduce manning, the maintainability, and most importantly the survivability and the reconfigurabilty. In this chapter we had focused on the reconfiguration as a strategy for dealing with malfunctioning and regulation problems. In this chapter we introduced the concept of Integrated Power System for Shipboard Power System. A reconfiguration strategy was proposed for maintaining safety operation after a loss of generation. A transition diagram highlighted the reconfiguration scheme. The dynamic models of the Integrated Power System, as well as the algebraic equations were provided. Due to the presence of discrete components such as circuit breakers, battery switching and our hybrid modeling approach, a special numerical algorithm was developed for solving hybrid differential-algebraic equations.

CHAPTER 9:OPTIMIZATION AND SIMULATIONS

9.1 Introduction

The concept of reconfigurability was introduced in section 8.3. What is presented in this chapter is how to optimally determine the reconfiguration strategy. In this chapter we will define the optimal control problem and present the problem formulation for the DDG 1000 integrated power system. We will also apply the discrete DAE computation to the DDG model and will generate simulations plots follows by comments of simulation results.

9.2 Optimization

The optimal control theory was discussed in CHAPTER 4: and CHAPTER 7: for the 3-bus power system. In this section we will define the optimal control problem and present the problem formulation for the case of the Integrate Power System.

9.2.1 Control Problem Formulation

The integrated power system was described in section 8.2, it is mainly composed of two generators, two motors and transmission lines connecting them. The reconfiguration of an IPS consists of activating various circuit breakers to shed non vital load, or reduced propulsion, or switching vital loads to battery supply. All in order to maintain a favorable voltage profile throughout the network.

Our approach to optimal control design once again is based on finite, (receding horizon) dynamic programming. The state trajectory of the integrated power system is described by two linear discrete-time dynamic of the motor slips, and battery that evolves over a finite time period. This period is divided into *N=25* equally spaced intervals and *k= 0.5* is the discrete time index. The state space of the slips values are {*0, 0.1, 0.2, 0.3, 0.4, 0.5*}. The discrete state space is $\{q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9, q_{10}, q_{11}, q_{12}, q_{13}\}$. The control variables are the binary variables $\{s_1, s_2, s_3, s_4, s_5\}$ and the regulated variable is V_2. The goal is to keep the load voltage V_2 close to one, specifically, we require $0.95 \le V_2 \le 1.05$.

Differential-Algebraic Equations:

- **Differential Equations:**

$$\begin{bmatrix} \dot{s}_{m1} \\ \dot{s}_{m2} \\ \dot{\sigma} \end{bmatrix} = \begin{bmatrix} \dfrac{1}{I_m \omega_0^2}\left(P_{m1} - V_4^2 \dfrac{R_r s_{m1}(1-s_{m1})}{R_r^2 + s_{m1}^{\ 2} X_r^2} \right) \\ \dfrac{1}{I_m \omega_0^2}\left(P_{m2} - V_5^2 \dfrac{R_r s_{m2}(1-s_{m2})}{R_r^2 + s_{m2}^{\ 2} X_r^2} \right) \\ \dfrac{i}{C} \end{bmatrix} \tag{9.1}$$

- **Algebraic-Equations:**

The algebraic equations are the real and the reactive power flow equation in (8.11) and (8.12)

$$\begin{aligned} 0 &= g_1(V_1, V_2, V_3, V_4, V_5, V_6, V_7, \theta_2, \theta_{23}, \theta_{24}, \theta_{27}, \theta_{57}, \theta_{67}) \\ 0 &= g_2(V_1, V_2, V_3, V_4, V_5, V_6, V_7, \theta_2, \theta_{23}, \theta_{24}, \theta_{27}, \theta_{57}, \theta_{67}) \end{aligned} \qquad (9.2)$$

- **Differential-Algebraic Equations-Discrete**

The function DAEDiscrete is the descretized form the DAE. Due to space limitation we will not attempt to give descretized DAE results but, we will give only the operational form:

$$DAEDiscrete\begin{bmatrix} \dot{s}_{m1} = \dfrac{1}{I_m\omega_0^2}\left(P_{m1} - V_4^2\dfrac{R_r s_{m1}\left(1-s_{m1}\right)}{R_r^2 + {s_{m1}}^2 X_r^2}\right) \\ \dot{s}_{m2} = \dfrac{1}{I_m\omega_0^2}\left(P_{m2} - V_5^2\dfrac{R_r s_{m2}\left(1-s_{m2}\right)}{R_r^2 + {s_{m2}}^2 X_r^2}\right) \\ \dot{\sigma} = \dfrac{i}{C} \\ 0 = g_1(V_1, V_2, V_3, V_4, V_5, V_6, V_7, \theta_2, \theta_{23}, \theta_{24}, \theta_{27}, \theta_{57}, \theta_{67}) \\ 0 = g_2(V_1, V_2, V_3, V_4, V_5, V_6, V_7, \theta_2, \theta_{23}, \theta_{24}, \theta_{27}, \theta_{57}, \theta_{67}) \\ \\ \\ \end{bmatrix} \qquad (9.3)$$

Constraints:

A relatively large set of constraints obtained from the logic specifications presented in (8.2) and (8.3). We will only give the general form of constraints:

$$E_6\delta_{qq_i} + E_7 d \le E_0 + E_1 x + E_2\delta_{q_i} + E_3 y + E_4\delta_s + E_5\delta_e\ , \quad i=2,...,13 \qquad (9.4)$$

Cost Function

Consider the following cost function:

$$J=(((V_7 - 1)^2 + (V_2 - 1)^2 + .25(V_4 - 1)^2 + .25(V_5 - 1)^2 + r\,(\delta_{qq4} + \delta_{qq5} + \delta_{qq6} + \delta_{qq7} + \delta_{qq10} + \delta_{qq11} + \delta_{qq12} + 2(\delta_{qq8} + \delta_{qq9} + \delta_{qq13})))/m); \tag{9.5}$$

The cost function is a multi-objective cost function that includes a deviation of the reference voltage term: $||V_7 - 1||^2$, $||V_2 - 1||^2$, $||V_4 - 1||^2$, $||V_5 - 1||^2$ where V_7, V_2, V_4, V_5 are bus voltage at the port, starboard, motor on starboard, motor on port side. The two motors bus voltages are weighted less than the generator bus voltages. The other cost elements are indicators $\delta_{qq4} + \delta_{qq5} + \delta_{qq6} + \delta_{qq7} + \delta_{qq10} + \delta_{qq11} + \delta_{qq12}$ corresponding to the summation of all state where exactly one of the non-vital load is shed for a weighting constant, r =1/25 and $\delta_{qq8} + \delta_{qq9} + \delta_{qq13}$ corresponding to the summation of the state where all non-vital load are shed for a high penalty of 2. The cost function is a weighted average (i.e divided by m).

Summary:

$$J=(((V_7 - 1)^2 + (V_2 - 1)^2 + .25(V_4 - 1)^2 + .25(V_5 - 1)^2$$
$$+ r\,(\delta_{qq4} + \delta_{qq5} + \delta_{qq6} + \delta_{qq7} + \delta_{qq10} + \delta_{qq11} + \delta_{qq12} + 2(\delta_{qq8} + \delta_{qq9} + \delta_{qq13})))/m)$$

$$s.t.\ \ DAEDiscrete\begin{bmatrix} \dot{s}_{m1} = \dfrac{1}{I_m\omega_0^2}\left(P_{m1} - V_4^2\dfrac{R_r s_{m1}(1-s_{m1})}{R_r^2 + s_{m1}{}^2 X_r^2}\right) \\ \dot{s}_{m2} = \dfrac{1}{I_m\omega_0^2}\left(P_{m2} - V_5^2\dfrac{R_r s_{m2}(1-s_{m2})}{R_r^2 + s_{m2}{}^2 X_r^2}\right) \\ \dot{\sigma} = \dfrac{i}{C} \\ 0 = g_1(V_1, V_2, V_3, V_4, V_5, V_6, V_7, \theta_2, \theta_{23}, \theta_{24}, \theta_{27}, \theta_{57}, \theta_{67}) \\ 0 = g_2(V_1, V_2, V_3, V_4, V_5, V_6, V_7, \theta_2, \theta_{23}, \theta_{24}, \theta_{27}, \theta_{57}, \theta_{67}) \\ \\ \end{bmatrix}$$

$$E_6\delta_{qq_i} + E_7 d \le E_0 + E_1 x + E_2\delta_{q_i} + E_3 y + E_4\delta_s + E_5\delta_e\ ,\quad i=2,..,13 \tag{9.6}$$

9.2.2 Optimal Controller Synthesis

A cost function with no terminal cost which specified voltage deviations is proposed. The hybrid modes reflected the simulation through the transition structure which are embedded in the IP-formulas (inequality constraints). The dynamic of the system is the difference-Algebraic Equations based on the slips and the battery state of charge and the algebraic equations. An optimal controller is synthesized as a look up table in term of the status of switches that can be enabled or disabled over the entire time horizon. The code is described below.

Mathematica Code Formulation:

ContinuousS = Flatten[Outer[List, Range[0, 1, 0.25], Range[0, 0.5, 0.1],Range[0, 0.5, 0.1], 2];

h = 0.5; m = 25;r = 1/25;

DiscreteS = Range[1, 12];

S = Map[Flatten[#] &, Flatten[Outer[List, ContinuousS, DiscreteS, 1], 1]];

Γ*[BinaryVars_List, RealVars_List, StateVars_List, n_Integer] :=*

$(((V_7 - 1)^2 + (V_2 - 1)^2 + .25(V_4 - 1)^2 + .25(V_5 - 1)^2 + r\,(\delta_{qq4} + \delta_{qq5} + \delta_{qq6} + \delta_{qq7} + \delta_{qq10} + \delta_{qq11} + \delta_{qq12} +$

$2(\delta_{qq8} + \delta_{qq9} + \delta_{qq13})))\,/\,m);$

$\Gamma N[\,StateVars\_List\,] := 0;$

BinaryStateVars = $\{\delta_{q2},\delta_{q3},\ \delta_{q4},\delta_{q5},\ \delta_{q6},\delta_{q7},\delta_{q8},\delta_{q9},\ \delta_{q10},\delta_{q11},\ \delta_{q12},\delta_{q13}\};$

BooleanStateVars = {q2, q3, q4, q5, q6, q7,q8, q9, q10, q11, q12, q13,

qq2, qq3, qq4, qq5, qq6, qq7,qq8, qq9, qq10, qq11, qq12, qq13};

DiscreteState = BooleanStateVars;

NextBinaryState = $\{\delta_{qq2},\delta_{qq3},\delta_{qq4},\delta_{qq5},\delta_{qq6},\delta_{qq7},\delta_{qq8},\delta_{qq9},\delta_{qq10},\delta_{qq11},\delta_{qq12},\delta_{qq13}\};$

BinaryControlVars = $\{\delta_{s_1},\delta_{s_2},\delta_{s_3},\delta_{s_4},\delta_{s_5}\};$

BooleanControlVars = $\{s_1,s_2,s_3,s_4,s_5\};$

RealStateVars = $\{\sigma,sl1,sl2\};$

RealVars = $\{V_1,\ V_2,V_3,\ V_4,V_5,\ V_6,V_7,\ \theta_2,\ \theta_{23},\ \theta_{27},\ \theta_{24},\ \theta_{57},\ \theta_{67}\};$

AllRealVars = $\{\sigma,\ sl1,\ sl2,\ V_1,\ V_2,\ V_3,\ V_4,\ V_5,\ V_6,V_7,\ \theta_2,\ \theta_{23},\ \theta_{27},\ \theta_{24},\ \theta_{57},\ \theta_{67}\};$

BinaryVars = Complement[varBinary, BinaryStateVars];

y0 = {1,1,1,1,1,1,1,0,0,0,0,0,0}

{T, Control} = Timing[OptimalPolicyDAE[eqBinary, eqReal, BinaryVars,RealVars, { },

BinaryStateVars, BinaryStateVars, BooleanStateVars, NextBinaryState, RealStateVars,

DAERHS, y0, BinaryControlVars, ContinuousS, DiscreteS, ΓN*,* Γ*, m]];*

Look Up Table implementation:

In order to put the controller in a form that is readable from Matlab/SIMULINK, the following Mathematica code builds a table lookup and generates a mfile called TableDefsI that is map as a SIMULINK block.

Mathematica command for building lookup table:

BuildLookupTable[{5, 6, 6, 12}, BinaryControlVars, {“AAA”, “BBB”, “CCC”, “DDD”, “EEE”}, Control[[-1]], “TableDefs.m”]

- The first argument of BuildLookupTable, {5, 4, 7} is the list of statedimension, where 5 is the dimension of the discretized state of charge of the battery , 6, 6 is the dimension of the two slips, and 12 is the dimension of discrete mode (i.e. Dim[DiscreteS] = 12)
- The second argument is the list of Control Variables.
- The third argument, {“AAA”, “BBB”, “CCC”, “DDD”, “EEE”} is the list of Table names corresponding to the five control variables.
- The fourth argument is the Controller: Control[[-1]]
- The fifth argument is the Table look up file name: TableDefs.m

The optimal control was computed in about 33 hrs on a laptop with 1.1 Ghz Pentium M.

Simulation results will be analyzed to confirm the performance of the controllers.

Optimal Controller: Lookup Table

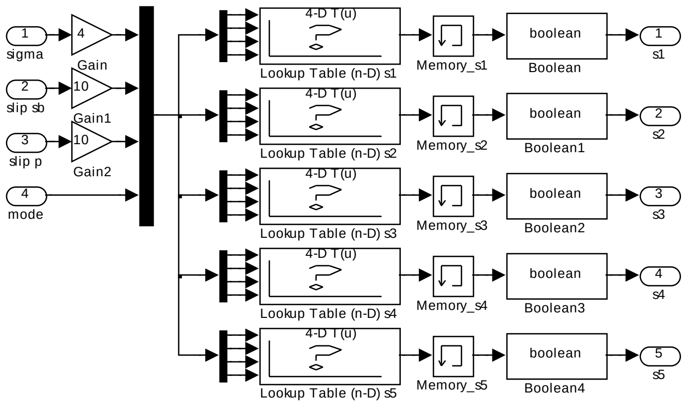


Figure 9.2-1:Table Look up

This lookup table has input {sigma, slip sb, slip p, mode}. The output is $\{s_1, s_2, s_3, s_4, s_5\}$.

9.3 Simulations

Simulations and the interpretation of their results are the ultimate steps in the conceptualization of any engineering or scientific research. Simulation provides a framework for validating assumptions made before simulation and an approximation of the behavior of a system before any attempt to a real-time simulation of that system.

Implementation of the Integrated Power System in SIMULINK using the SimPower Toolset:

SIMULINK provides a power system simulation tool called SimPower. Various power system components such as generator, exciter, motor, transformer, short transmission line are available for open loop power system design.

The SimPower model of the IPS is modeled based on the description of the IPS provided in Figure 8.2-2 with the modification that the system is a three phase system with the generator dynamics incorporated. The dynamic of the governor and prime mover are neglected so that the generators receive a constant mechanical input of 40 MW. An exciter is attached to the generator in order to regulate the generator terminal voltage to 13.8 kV. Two 13.8-4.16 kV transformers between the generators and the motors busses ;and two 13.8-0.450 kV transformers between the generators and non-vital loads busses. The mechanical power input to the induction motors is 36 MW. The transmission line and tie line are modeled as short transmission lines.

Faults are modeled by using circuit breakers. A generator fault on the port side is modeled by disconnecting the generator from the system.

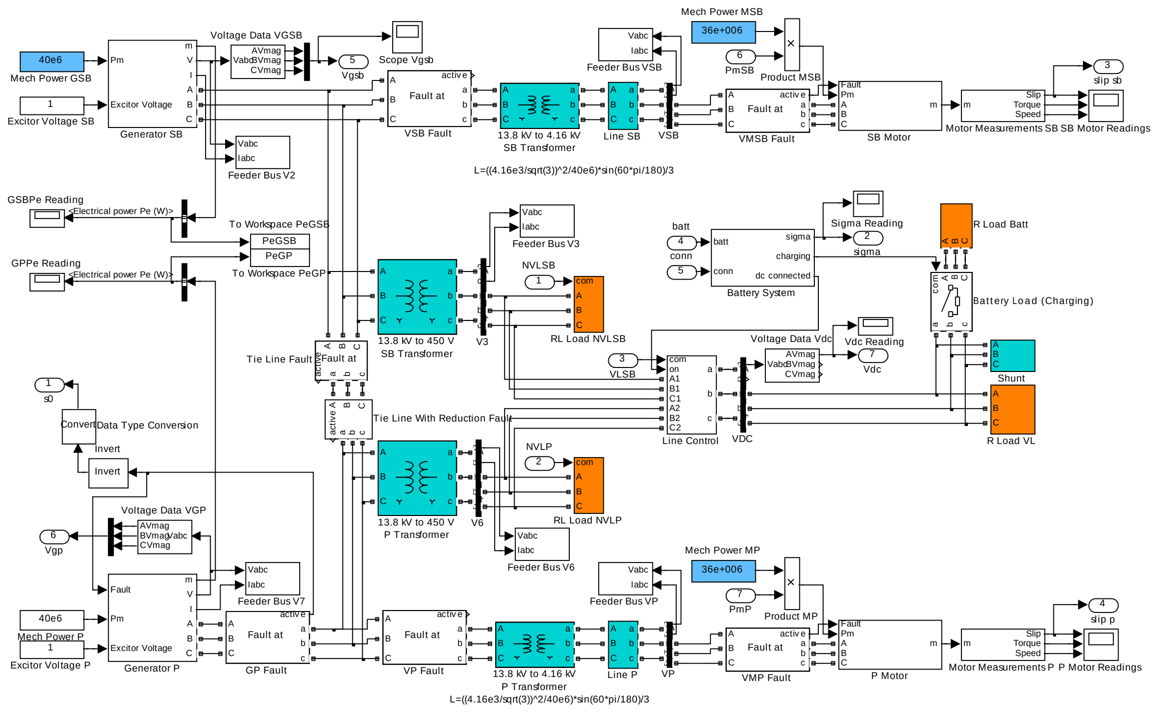


Figure 9.3-1: IPS in SimPower

In the DDG 1000s case study we will consider only with one fault, the generator fault on the port side. Other fault such as tie line fault was conducted but is reported in this thesis. The user can select the time of occurrence of a fault. We selected time of the fault as *t= 1 sec.*

9.3.1 Open Loop Simulation and Results:

An open loop simulation is simulation with some input to the controller fixed. All the circuit breakers are set constants except the circuit which activates the generator fault on the port side.

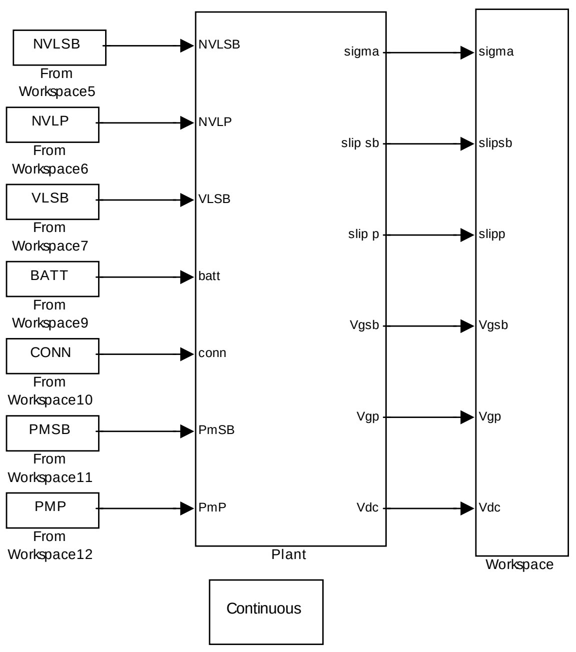


Figure 9.3-2:Open Loop System with circuit breakers set.

Figure 9.3-2 indicates how the circuit breakers feed into the plant.

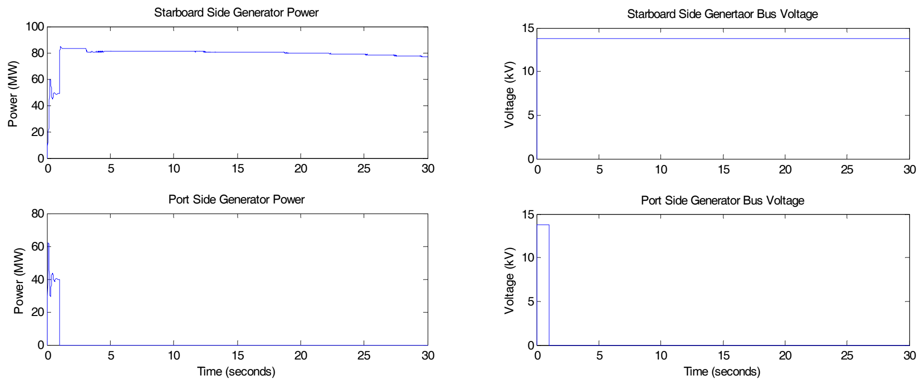


Figure 9.3-3: Open Loop Generator Powers and Bus Voltages

Figure 9.3-3 shows that when a generator fault occurs on the port side at time *t=1 sec*, the power on the port side generator drops to zero and the power on the starboard side jumps from 48 MW to 82 MW, then reaches 78 MW after *t =30 sec.* The bus voltage on the starboard side is maintained at 13.8 kV. The port side generator power and voltage immediately drop to zero after the fault on the port side generator.

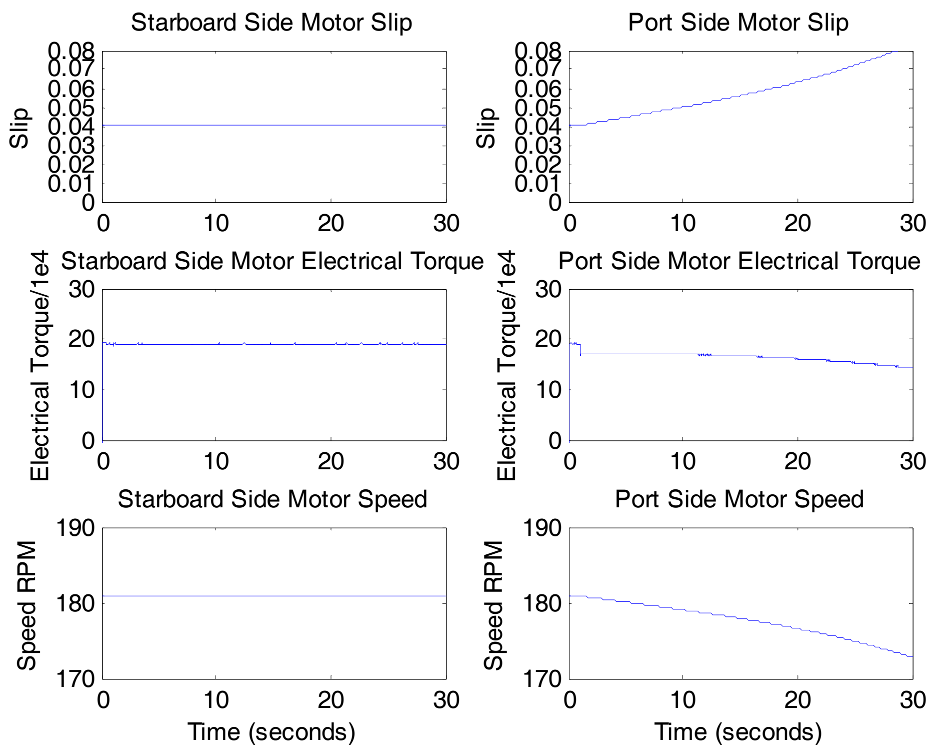


Figure 9.3-4:Open Loop Motor Slip, Electrical Torque, Speed

In Figure 9.3-4, when a fault occurs at time *t =1 sec,* on the port side, the motor slip increases and electrical torque and the speed decrease.

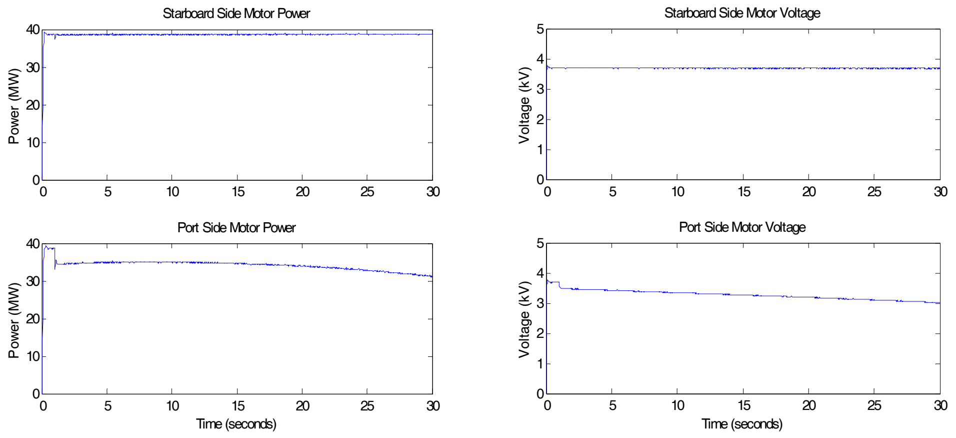


Figure 9.3-5:Open Loop Motor Power and Voltages

Figure 9.3-5 shows that with a generator fault on the port side at time *t = 1 sec*, without reduction of power (i.e. propulsion) on the motors, the power and the voltage on the port side will collapse.

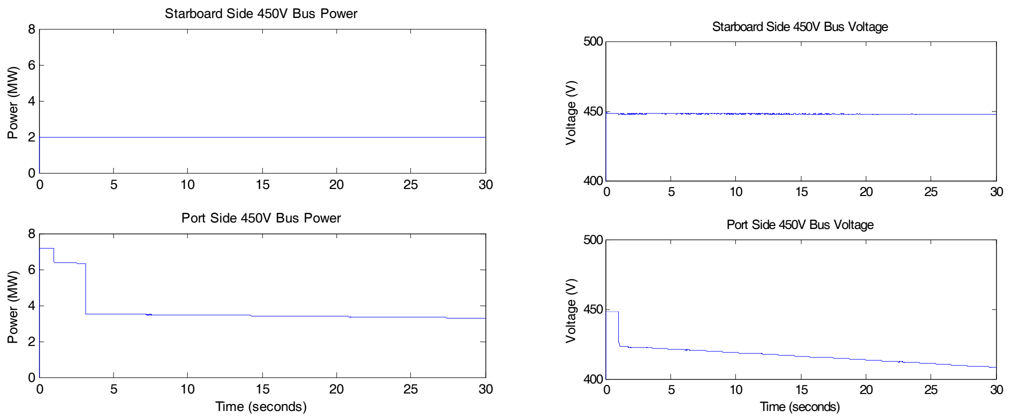


Figure 9.3-6: Open Loop Vital load Powers and Bus Voltages

In Figure 9.3-6 the Vital load bus power demand on the starboard side before and after the fault at *t =1sec,* is 2 MW. On the port side the power drops and the voltage is on the verge of collapsing after *t =30 sec.*

9.3.2 Closed Loop Simulation and Results:

A comparison can be made, between the open loop simulation and the closed loop simulation to evaluate the effectiveness of the optimal controller.

Simulation Framework:

The simulation framework describes the interconnection of different components of the integrated power system in SIMULINK environment.

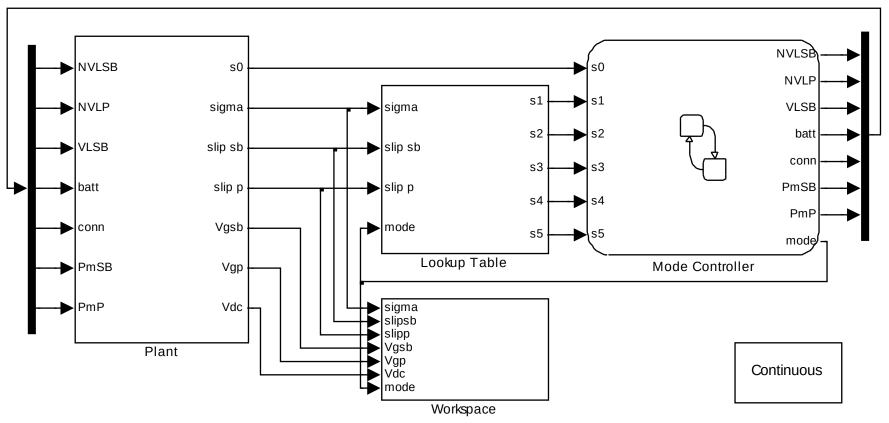


Figure 9.3-7:Simulation Framework

The plant represents the integrated power system with the switch breakers as inputs and as outputs s_0, slip sb, slip p and sigma representing the state of the switch s_0, the slip on the starboard, the slip on the port, the state of charge of the battery, and then *Vgsb, Vgp, Vdc*, corresponding to voltage on the generator starboard, generator port and DC vital load.

The Lookup Table receives as input the state of charge sigma, the slip on the starboard slipsb, the slip on the port, slipp and the mode of the system

The workspace contains the graphical representation of inputs signals indicated in the workspace block.

The mode controller is the stateflow representation of the thirteen states of the integrated power system. Its input signals are initial switch s_0 and the five switches s_1, s_2, s_3, s_4, s_5. The outputs are the status of the circuit breakers.

Let us analyze how the components in the simulation framework feed to each other.

- The plant feeds into {mode controller, Lookup Table, Workspace}.
- The mode controller feeds back into the {Plant, Workspace, Lookup Table}.
- The Lookup Table feeds into the {Mode Controller}.

Stateflow diagram:

As indicated before stateflow is a graphical tool in SIMULINK for representing a hybrid automaton. Each state contains the circuit breakers setting of a configuration of the integrated power system. The changes in the circuit breakers status are dictated by the optimal controller that was derived using the mixed-integer dynamic programming.

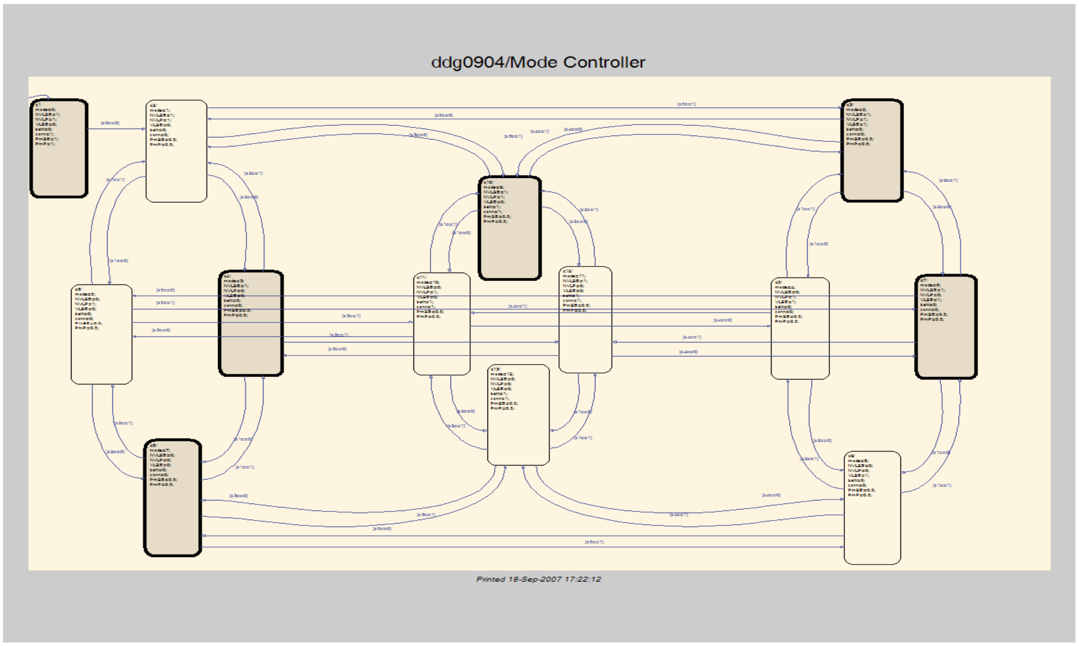


Figure 9.3-8:Stateflow Diagram for the DDG 1000

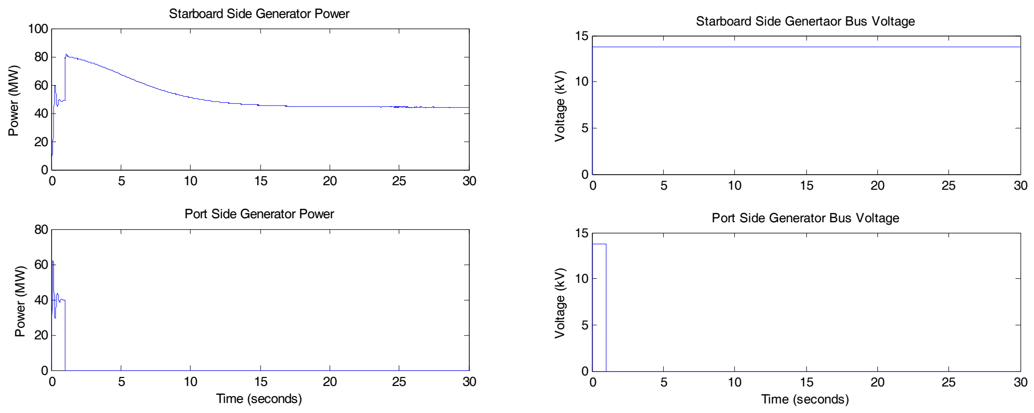


Figure 9.3-9: Closed Loop Generator Powers and Bus Voltages

Figure 9.3-9 Shows that when a generator fault occurs on the port side at time *t=1 sec*, the power on the port side generator drop to zero and the power on the starboard side at *t=1 sec,* jumps from 48 MW to 82 MW, then it starts to slowly decrease and reaches a steady-state value 42 MW after *t =17 sec*. The bus voltage on the starboard side is maintained at 13.8 kV.

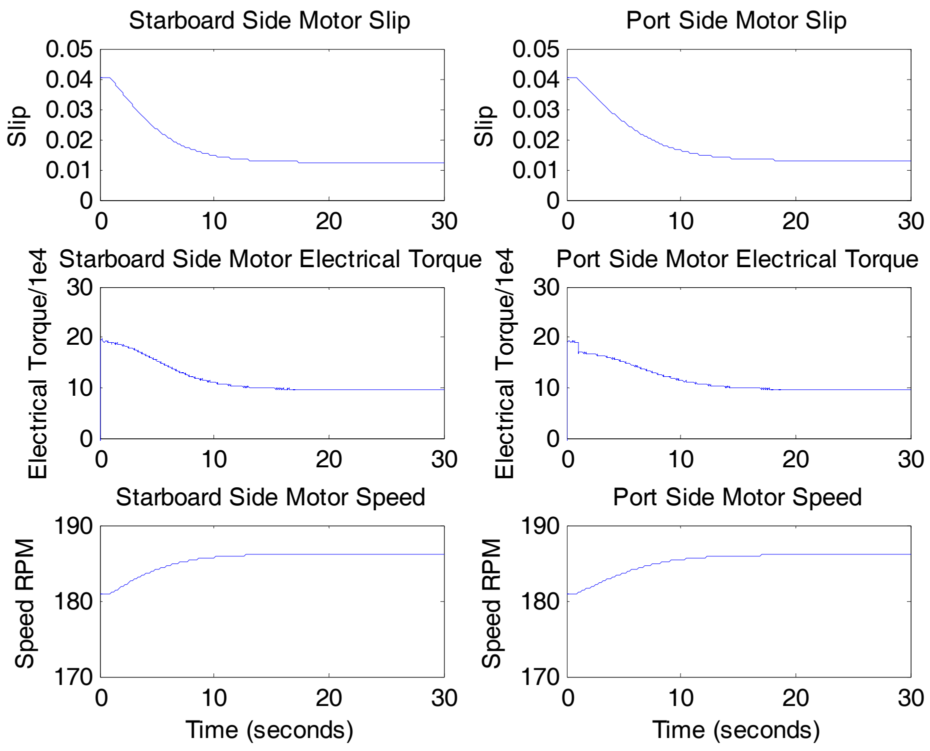


Figure 9.3-10:Closed Loop Motor Slip, Electrical Torque, Speed

In Figure 9.3-10, when a fault occurs at time *t =1 sec* the motor slip and electrical torque decrease to a steady state values of 0.012 and 9.8 respectively, whereas the speeds increase to a steady-state values of 186 RPM.

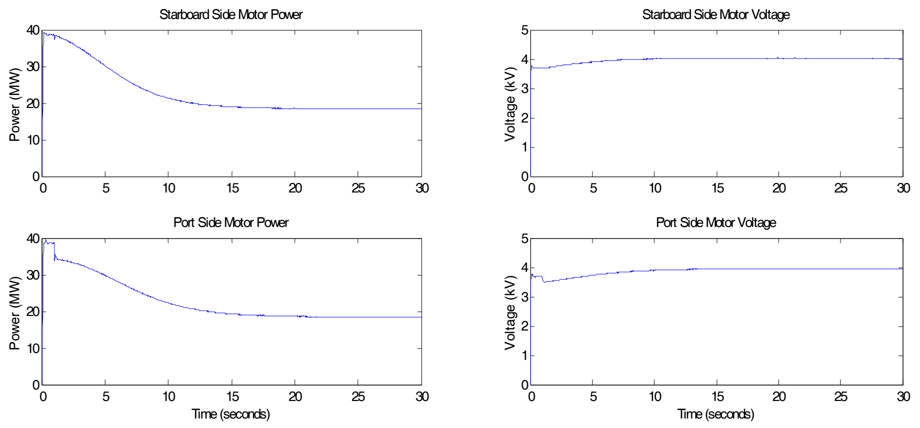


Figure 9.3-11:Closed Loop Motor Power and Voltages

Figure 9.3-11 shows that with a generator fault on the port side at time *t* = *1 sec*, a reduction of power on both motors will prevent the voltage on both motors to collapse and to maintain the bus voltages of the motors within a small deviation from the reference voltage of 4kV.

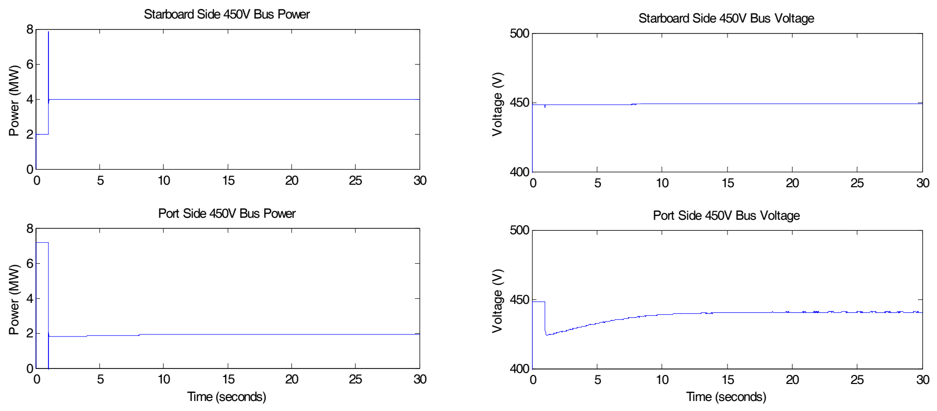


Figure 9.3-12 : Closed Loop Vital load Powers and Bus Voltages

In Figure 9.3-12 the Vital load bus power demand on the starboard side before the fault at t=1sec, is 2 MW, after the fault the power demand is 4 MW, whereas on the port side, before the fault the power demand is 7.2 MW, after the fault the power demand is 2 MW.

9.4 Conclusion

The design of the Integrated Power System for future Navy’s Ships is a challenging task; we have proposed a reconfiguration approach based on the emerging field of optimal hybrid control design. The approach is sound and constitutes a step toward a global optimal design. The problem of reconfigurations of the Integrated Power System was formulated as an optimal hybrid control. An Optimal controller was synthesized in a form a lookup table. Simulation was conducted on the open loop plant followed by the simulation of the closed loop plant. A comparative study of the open loop and the closed showed that the closed loop plant achieved the reconfiguration goal and prevented voltage collapse.

CHAPTER 10: CONCLUSION AND FUTURE RESEARCH

10.1 Conclusion

The objective of this thesis was to design a power management system capable of dynamically reconfiguring the shipboard power systems in order prevent loss of power at critical buses when damage conditions are encountered. A salient aspect of power management for mission critical power systems is the presence of an uninterruptible power supply that provides power to vital loads in case of in extremis situations such as imminent voltage collapse.

In order to tackle this challenging problem we proposed a new design paradigm based on hybrid control systems theory. Hybrid systems theory provides an ideal design framework for dealing with complex decision problems that involve discrete as well as continuous actions. Discrete subsystems can be associated with logical conditions. We had expressed that notion of discreteness by making logical specification an integral part of the hybrid modeling and design process.

The main contributions of this thesis involved the development of:

- The transition dynamics of the discrete event system are expressed in terms of a logical specification

- A Mathematica package has been developed that converts any logical specification into a set of inequalities in binary and possibly real variables

- A Mathematica package that computes optimal feedback switching strategies based on mixed integer dynamic programming has been developed

- Continuous dynamics were modeled as sets of differential-algebraic equations. The latter are essential for modeling realistic power systems. Two approaches to computation with DAE models have been addressed.

- The control problem was viewed as a problem of 'bifurcation control' where discrete actions were taken to prevent the occurrence of bifurcations.

On the application viewpoints, two examples were investigated. The first example was a 3-bus power system with UPS, in that example we studied the feasibility of our approach. A bifurcation control analysis was conducted on this system. The significance of the bifurcation analysis was that it showed the stability limits of the system for parameter variation and gave a clear picture of how discrete actions could prevent bifurcation. It indicated what actions would be most effective and provided insight into the resultant stability margins. An optimal controller was generated using our mixed-integer dynamic programming tools. A simulation framework was built using the SIMULINK/Stateflow and simulation results agreed with expectations. The system was described by a system of differential-algebraic equations. However, it was possible to solve the algebraic equations on multiple domains of the state space. These domains being algebraic sets, a strategy for solving problems reducible to this form is easily implemented, because the logic identifying solution corresponding to a particular domain is easily incorporated into the logical specification.

The second example is more complex; the notional DDG 1000 integrated power system. In that example the emphasis was on the reconfiguration of the integrated power system after a loss of generation. Because the IPS is a relatively larger system, its model contained differential-algebraic equations in which the algebraic equations are too complex to be solved by present day elimination or quantifier elimination tools. As a result the continuous dynamics were numerical solved during the optimization process. After designing an optimal controller, various simulations were run.

10.2 Future Research

We identified four areas of research that should be pursued in the future

- **Hamilton-Jacobi Bellman equation (HJB)**

In our approach we used a discrete time scale and set up a temporal recursion using Bellman's Principle of Optimality. It would be interesting to set up the continuous time Hamilton-Jacobi-Bellman equation and study the impact of constraints in the form of IP formulas. Solution of even very simple problems of this type could provide valuable insight into the structure of optimal controllers for hybrid systems.

- **Variable Structure Control**

Variable structure control consists on designing switching surfaces and using discontinuous control to maintain the state system on the switching surface. Optimal switching surface can be designed by using the Pontryagin Maximum Principle. An optimal variable structure control can be cast as a logic-Based optimal control. The idea emanates from the fact that existing variable structure theory based on stabilization of the sliding manifolds provides essentially local controllers. If we consider *m* as the number of switching surfaces there corresponds 2^m regions of the state-space. A logic-based optimal control of variable structure control problem could be formulated with 2^m modes hybrid automaton. Therefore a logic-based optimal control could provide an ideal framework for solving variable structure system where global behavior is important.

- **Bifurcation control analysis for the IPS**

A study of bifurcation control was conducted for the 3-bus power system to investigate the effect of switching on voltage collapse. A bifurcation control study could also be conducted on the integrated power system. Critical buses could be identified and a performance index could be structured using this insight. Weaknesses in controllability could also be identified so that more effective switching options could be incorporated into the system.

- **Prime mover and governor dynamics for frequency analysis.**

In the integrated power system example with a loss of the port generation, the assumption made was to neglect the effect of prime mover dynamics by assuming an infinite inertia on the starboard generator, thereby establishing a fixed system frequency of 60 Hz. In reality the generator inertia is not infinite and there would be a frequency swing. Ordinarily, frequency variations are limited in order to protect equipment. In small systems frequency excursions could be as important as voltage collapse.

List of References

[1] Q. Li, Y. Guo, and T. Ida, "*Transformation of Logical Specification into IP-formulas,*" presented at 3rd International Mathematica Symposium (IMS '99), Hagenburg, Austria, 1999.

[2] K. McKinnon, H. Williams," *Constructing Integer Programming Models by Predicate Calculus*", Annals of Operation Research, 1989

[3] J. Hooker "*Logic-Based Methods for Optimization: Combining Optimization Constraint Satisfaction*" , Wiley-Intersciences, 2000

[4] T. Geyer, M. Larsson, M. Morari, "*Hybrid Emergency Voltage Control in Power System*" European Control Conference, 2003

[5] S. Hedlund, A. Rantzer , "*Optimal Control of Hybrid Systems*" , CDC, 1999

[6] H.K. Kwatny, E. Mensah, D. Niebur, C. Teolis, " *Optimal Shipboard Power System Management via Dynamic Mixed Integer Programming*", IEEE Electric Ship Technologies Symposium", Philadelphia, PA, 2005

[7] H.K. Kwatny, E. Mensah, D. Niebur, C. Teolis, G. Bajpai " *Symbolic Construction of Dynamics Mixed Integer Programs for Power System Management*" , PESC, Atlanta, GA, Feb 2007

[8] H.K. Kwatny, E. Mensah, D. Niebur, C. Teolis, " *Optimal Control of Hybrid System via Dynamic Programming using Mathematica*" IFAC Symposium on Power Plant and Power System, Kannaskis, Canada June 2006

[9] H.K. Kwatny, E. Mensah, D. Niebur, C. Teolis, J.E. Dongmo G. Bajpai " *Model for Optimal Dynamic Reconfiguration and Simulation of Ship Power System in Simulink with stateflow*" IEEE Electric Ship Technologies Symposium", Arlington, VA, Mai 2007

[10] H.K. Kwatny, E. Mensah, D. Niebur, C. Teolis, G. Bajpai " *Logic-Based Design of Optimal Reconfiguration Strategies for ship Power Systems*" IFAC, Nonlinear Control Symposium", Pretoria, South Africa, Aug 2007

[11] G.K. Fourlas, K.J. Kyriakopoulos, and C.D. Vournas, " *Hybrid Systems Modeling for Power Systems*" IEEE Cuircuits and Systems Magazine, 3rd Quarter 2004.

[12] Kazuro Tsuda, D. Mignone, G. Ferrari-Trecate, M. Morari," *Reconfiguration Strategies for Hybrid Systems*", Preceedings of the American Control Conference, Arlington, VA, June 2001.

[13] S.A. Attia, M. Alamir and C. Canudas De Wit, " *Voltage Collapse Avoidance in Power Systems a Receding Horizon Approach*", Intelligent Automation and Soft Computing, 2006.

[14] Chan-Chiao Lin, Huei Peng, J. W. Grizzle and Jun-Mo Kang," *Power management Strategy for a Parallel Hybrid Electric Truck*", IEEE Transactions on Control Systems Technology , Nov 2003

[15] Bo Lincoln, Anders Rantzer, " *Relaxing Dynamic Programming*" IEEE Trans on Automatic Control,Vol. 51,No. 8 August 2006.

[16] H.S. Witsenhausen. "*A class of hybrid-state continuous-time dynamics systems".* IEEE Trans. On automatic Control, AC-11(2):161-167, 1966.

[17] A. Bemporad, M. Morari " *Control System Integrating Logic, Dynamics and Constraints",* Automatica, 1999

[18] Rajeev Alur, Thomas A. Henzenger , Howard Wong-Toi "*Symbolic Analysis Of Hybrid Systems*" Proceeding of the 37th IEEE,Conference on Decision and Control,1997

[19] M.S. Branicky, V.S. Borkar, S.K. Mitter " *A Unified Framework for Hybrid Control: Model and Optimal Control Theory*" IEEE Transaction On Automatic Control, Vol 43, pp 31-45, 1998

[20] Sven Hedlund, Anders Rantzer " *Hybrid Control Laws From Convex Dynamic Programming*", CDC, Dec 2000

[21] Peter J.G. Ramadge , W. Wonham, "*The Control of Discrete Event Systems*" Proceeding of The IEEE,Vol 77, No 1, January 1989

[22] I.A. Hiskens, M.A. Pai, " *Hybrid Systems View of Power Systen Modelling*", , IEEE, ISCAS, 2000, Switzerland

[23] M.-S. Lee, J.-T. Lim " *Restoration Strategy for Power distribution network using Optimal Supervisory Control"* ,IEE Proc.-Gener. Transn. Distrib. Vol 151, No. 3, May 2004

[24] Karen L. Butler, N.D.R. Sarma, V. Ragendra Prasad,"*Network Reconfiguration for Service Restoration in Shipboard Power Distribution Systems*", IEEE, Transn. On Power Systems, Vol. 16, No. 4,Nov 2001

[25] A. Bemporad, G. Ferrai-Trecate, M. Morari,"*Observability an Controllability of Piecewise Affine an Hybrid Sysyems*", IEEE, Transn. On Automatic Control, Vol 45,No. 10, Oct 2000

[26] G. Ferrai-Trecate, D. Mignone, M. Morari,"*Moving Horizon Estimation for Hybrid Sysyems*", IEEE, Transn. On Automatic Control,Vol 45,No. 10, Oct 2000

[27] A. Bemporad, W.P.M.H. Heemels, B. De Schutter, "On Hybrid Systems and Closed-Loop MPC Systems", Proc, of 40th IEEE, CDC,Orlando,2001

[28] T. Geyer, G. Papafotiou, M. Morari, "*On the Optimal Control of Switched-Mode DC-DC Converters*", Springer-Verlag, HSCC,pp. 342-356, 2004

[29] A. Bemporad N. Giorgetti,"*A Logic-Based Hybrid Solver for Optimal Control of Hybrid Systems*", Proc. Of the 42nd IEEE, CDC, Maui, Hawaii, Dec. 2003

[30] J.M. Davoren, A. Nerode, " *Logics for Hybrid Systems*" Prec. Of The IEEE, Vol. 88 No. 7,July 2000

[31] C. Tomlin, G. Pappas,"*Conflict Resolution for Air Traffic Management :A Study in Multiagent Hybri Sytems*", IEEE Transc. On Aut Cont, Vol.43 No.4, April 1998

[32] M. Rodrigues, D. Theilliol, D. Sauter,"*Fault Tolerant Control Design for Switched Systems* ", 2n IFAC Conf. on Analysis and Design of Hybri Systems, Alghero,Italy,June 2006

[33] M. Zefran, J.W. Burdick,"*Design of switching controllers for systems with changing dynamics*" Proc. Of the 37th IEEE, CDC, Tampa, Florida, Dec. 1998

[34] S. Helund, A. Rantzer, "*Hybrid Control Laws From Convex Dynamic Programming*"

[35] I. Dobson, "*Computing a Closest Bifurcation Instability in Multidimensional Parameter Space*" Springer-Verlag,J. Nonlinear Sci. Vol. 3,pp. 307-327,1993

[36] I. Dobson, "*Observations on the Geometry of Sadle Node Bifurcation and Voltage Collapse in Electrical Power Sysytems*", IEEE Transc. On circuits and systems I: Fund. Theory and Appl. Vol. 39,No. 3, March 1992

[37] Mireille Brouke, "*A geometric Approach to Bisimulation and Verification of Hybrid Systems*" Springer-Verlag, Berlin Heisdelberg, 1999

[38] W.J. Bencze, G.F. Franklin, "*A separation Principle for Hybrid Control System Design*", IEEE, IFAC Symposium on CACSD, Tucson, AZ, 1994

[39] E.C. Karrigan, A. Bemporad, M. Mignone,,M. Morari,. J.M. Maciejowski , "*Multi-Objective Prioritisation and Reconfiguration for the Control of Constrained Hybrid Systems*", IEEE, ACC, Chicago,Il, 2000

[40] S. Ayasun, C.O. Nwankpa, H.G. Kwatny "*Computation of Singular an Singularity Induced Bifurcation Points of Differential-Algebraic Power System Model*", IEEE Transc. On circuits and systems I: Vol. 51, No. 8, Aug. 2004

[41] O. Stein, J. Oldenburg, W. Marquardt, "*Continuous reformulations of discrete – continuous optimization*", Elsevier, Comp. and Chem. Eng. 28, 2004

[42] H. Ohtsuki, A. Yokoyama, Y. Sekine, "*Reverse Action of on-load Tap Changer in Association with voltage collapse*" IEEE Transc. On Power Systems, Vol. 6, No,1, Feb. 1991

[43] P.E Caines, S. Wang, "*Classical and Logic Based Regular Regulator Design and its Complexity for partially Observed Automata*", Prec. Of the 28th CDC,Tampa,Floria, Dec. 1989

[44] P.E Caines, R. greiner, S. Wang, "*Dynamical Logic Observers for Finte Automata*", Prec. Of the 27th CDC, Austin, Texas, Dec. 1988

[43] P.E Caines, S. Wang, "*A Conditional Observer and Controller Logic For Finite Machines*", Prec. Of the 29th CDC,Honolulu, Hawaii, Dec. 1990

[44] S. Artemov, J. avoven, A. Nerode, "*Logic, Topological Semantics and Hybrid Systems*", Proc. On the 36th CDC, San Diego, CA, Dec. 1997

[45] J. H. Prosser, Moshe Kam, H.G. Kwatny,"*Online Supervisor Synthesis for Partially Observed Discrete-Event Systems*" IEEE Transc. On Aut. Cont. Vol. 43, No.11,Nov. 1998

[46] R.A. Decarlo, M. Branicky, S. Pettersson, B. Lennartson,"*Perspectives and Results on the Stability and Stabilizability of Hybrid Systems*",Proc. Of the IEEE, Vol. 88, No.7, July 2000

[47] R. Brocket, "*Hybrid Models of motion Control Systems*", Birkhauser, Book, Perpectives in Control, 1993

[48] H. P. Williams, H. Yan" *Representations of all-different Predicate of Constraint Satisfaction in Integer Programming*" Informs Journal on Computing,No.13, Vol.2 ,Spring 2001

[49] H.K. Kwatny, E. Mensah, D. Niebur, C. Teolis, "*Symbolic Construction of Dynamic Mixed Integer Programs for Power System Management*", IEEE, PES ", San Francisco, CA, June 2005

[50] SYNTEK, "*CAPS DD IPS Electrical Distribution One-line Diagram*" 2003

[51] SYNTEK, "*DD(X) Notional Baseline Modeling and Simulation Development*" SYNTEK Technologies, Arlington, VA, AUGUST 1, 2003

[52] H. Ohtsuki, A. Yokoyama, and Y. Sekine,"*Reverse Action On-load Tap Changer in Association with Voltage Collapse*" IEEE Transactions on Power System,vol.6, pp. 300-306, 1991.

[53] M.K. Pal,"*Voltage Stability: Analysis Needs, Modelling, requirement, and Modelling Adequacy*",IEE Proceedings –Generations-C, vol. 140, pp. 279-286. 1993

[54] L. Bao, X. Duan, and T. He,"*Analysis of Voltage Collapse Mechanism in State Space*" IEE Proceedings –Generations-Transmissions and Distribution, vol. 147, pp. 395-400. 2000

[55] B. Wu, R. Dougal, and R.E. White, "*Resistive Companion Battery Model for Electric Circuit Simulations*" Journal of Power Sources, vol. 93, pp. 186-200, 2001.

[56] H.P. Williams, "*Model Building in Mathematical Programming*", John Wiley & Sons, 2nd ed.

[57] Mats Jirstrand, "*Constructive Methods for inequality Constraints in Control"*, Thesis, 1998

[58] I.A. Hiskens and B. Gong ,"*Voltage Stability Enhancement via Model Predictive Control of Load*", Intelligent Automation and Soft Computing, Vol 12, No, pp 1-8,2006

[59] Kwatny G.H, G. Blanknship, "*Nonlinear Control and Analytical Mechanics: A Computational Appraoch*", Birkhauser, 2000

[60] Bergen, A. R Vittal, V. "*Power Systems Analysis*", Prentice Hall, 2000, 2nd ed.

[61] Kwatny G.H., Fischl R.F.,Nwankpa, C,"*Local Bifurcation in Power Systems:Theory, Computation and Application"*, Proceeding of the IEEE,special issue in nonlinear phenomenon in Power systems theory and Practical Application,vol 83, No 11, p9 1456-1483,Nov. 1995

[62] Guanrong Chen, David Hill, Xinghuo Yu,"*Bifurcation Control: Theory and Application"*, Springer, LNCIS, 2003

[63] David Hill, Yi Guo, Mats Larson, And Youyi Wang," *Global Control of Complex Power Systems*", in " Bifurcation Control: Theory and Application", Springer, LNCIS, 2003

[64] Thierry Van Cutsem, Costas Vournas,"*Voltage Stability of Electric Power Systems*",

Kluwer Academic Publishers, Power Electronic and Power Systems Series, 1998

[65] M. K. Pal, "*Voltage Stability Conditions Considering Load Characteristics*", Transactions on Power Systems, Vol 7, No 1 Feb 1992

[66] N. Mohan, T. M. Undeland, W. P. Robbins,"P*ower Electronics*" John, Wiley 3rd Ed 2003

[67] Steven H. Strogatz,"*Nonlinear Dynamics and Chaos*",Westview, 1994

[68] Shankar Sastry,"*Nonlinear Systems*", Springer ,1999

[69] J.Hale, H. Kocak, "*Dynamics and Bifurcations*", Springer-Verlag,1991

[70] Program Esecutive Office, Ships, "*What is DDG 1000 History*"

http://peos.crane.navy.mil/DDG1000/What_is_DDG1000.htm

[71] SPG, Media, PLC," *Naval Technology*"

http://www.naval-technology.com/projects/dd21/

[72] GlobalSecurity.org," *Integrated Power System*"

http://www.globalsecurity.org/military/systems/ship/systems/ips.htm

[73] Program Esecutive Office, Ships, "*DDG 1000 History*"

http://peos.crane.navy.mil/DDG1000/Requirements_history.htm

[74] Christos G. Cassandras, Stephane Lafortune, "Introduction to Discrete Event Systems", Kluwer Academic Publishers, 1999

Appendix A: Per Unit Normalization

Generator 1:

Power Base: $S = 40MW$

Voltage Base: Vbase = 13.6 kV

Resistance: R=0.2236*0.01 Ω

Inductance: L=0.2236/2π60 H

$$Zbase = \frac{Vbase^2}{Sbase} = 4.761,\ Zactual = \ 0.002236 + 0.2236i,\ Zpu \ = \frac{Zactual}{Zbase} = 0.000469 + 0.0469i$$

$$Ypu = \ \frac{1}{Zpu} = 0.2129 - 21.29i,\ g_{12} = 0.2129,\ b_{12} = -21.29$$

Transformer 1 and 2:

$Sbase \ = \ 40\ MW$

$Vbase \ = 13.8\ kV$

$V_1 = 13.8\ kV($ primary), $V_2 = 4.16\ kV$ (secondary)

$R_1 = 0.001\,pu,\ R_2 = 0.001\,pu$

$L_1 = 0.04\,pu,\ L_2 = 0.04\,pu$

$R_m = 500,\ X_m = 500,\ (m = \text{magnetizing})$

Because the resistance and the inductance are in per unit we can write:

$Z_{2t} = (R_1 + R_2) + (L_1 + L_2)i = 0.02 + 0.08i$

Transmission lines:

The line model used is the short line model.

We need to specify the model transmission line parameters. The maximum power capacity of a line of admittance $Y_{jk} = ib_{jk}$ connecting busses with voltage magnitudes V_j, V_k is :

$$P_{\max} = V_j V_k b_{jk} \sin\frac{\pi}{3} = \frac{V_j V_k}{X_{jk}} \sin\frac{\pi}{3} \qquad (10.1)$$

Furthermore assume that the power base is 40 MW and that each of the two sides of the ideal transformer have a base as specified by the nominal bus voltages. Assuming that lines 2-4 and 7-9 are sized to carry 40 MW at nominal conditions, we have a basis for estimating the line admittances. Then, we have:

$$X_{jk} = \frac{V_j V_k}{P_{\max}} \sin\frac{\pi}{3} \qquad (10.2)$$

Since the transmission line is between the transformer and the voltage bus 4 we use X_{t4} as reactance where it defines the reactance of the combined three phase so we divide by a factor of 3 and multiply by a corrective factor 0.65 (i.e. to increase the reactance)

$$X_{t4} = \frac{\frac{V_t}{\sqrt{3}} \frac{V_4}{\sqrt{3}}}{9P_{\max}} \sin\frac{\pi}{3} \times 0.65 = \frac{V_t V_4}{27P_{\max}} \sin\frac{\pi}{3} \times 0.65 = 0.009 \quad , Z_{t4}=0.009i \text{ (SI)} \qquad (10.3)$$

$$Zbase = \frac{Vbase^2}{Sbase} = \frac{(4.16kV)^2}{40MW} = 0.4326,\ \mathrm{Z}_{t4} = \frac{Z}{Zbase} = 0.0208 \text{ p.u} \qquad (10.4)$$

The total admittance of the transformer and line is

$$Y_{24} = Y_{57} = \frac{1}{Z_{2t} + Z_{t4}} = 0.1966 - 9.912i,$$

$$g_{24} = 0.1966\ p.u,\ b_{24} = -9.912\ p.u. \qquad (10.5)$$

$$g_{57} = 0.1966\ p.u,\ b_{57} = -9.912\ p.u.$$

Transmission line 2-7: Tie line

$$X_{27} = (1/9)\frac{\frac{V_2}{\sqrt{3}}\frac{V_7}{\sqrt{3}}}{P_{\max}}\sin\frac{\pi}{3}\times 0.65 = \frac{V_2V_7}{27P_{\max}}\sin\frac{\pi}{3}\times 0.65 = 0.0992\ ,\ \mathrm{Z}_{t4}\text{=}0.0992i\ \text{(SI)} \qquad (10.6)$$

$$Zbase = \frac{Vbase^2}{Sbase} = \frac{(13.8\ kV)^2}{40MW} = 4.761,\ Z_{27} = \frac{Z_{27}}{Zbase} = -47.9645\ p.u.$$

$$(10.7)$$

$$g_{27} = 0,\ b_{27} = -47.9645\ p.u.$$

Constant admittance load:

Now, consider the loads. For the non vital loads at buses 3 and 6, we assume constant admittance load with a:

Lagging PF = 0.8

Real power: P = 2 MW

Vbase = 4.16 kV

Sbase = 40 MW

$$S = \frac{P}{\cos\theta} = 2.5\times10^{6}$$

$$S_{nvl} = S(\cos\theta + j\sin\theta) = 2\times10^{6} + j1.5\times10^{6}\ \text{(SI)}$$

$$S_{nvl} = \frac{S_{nvl}}{Sbase} = 0.05 + j0.0375\ p.u. = P + jQ$$

The power absorbed by the non vital load is:

$$P - jQ = |V|^{2}\,Y_{nvl}\quad ,$$

$$\mathrm{Y}_{\mathrm{nvl}} = \frac{S^{*}_{nvl}}{V_{3}^{2}} = \frac{0.05 - j0.0375\ p.u.}{1\ p.u} = 0.05 - j0.0375\ \text{p.u.}$$

- **Non vital load at bus 3:**

$g_{nvl1} = 0.05\ p.u.$, $\mathrm{b}_{\mathrm{nvl}} = \text{-}0.0375\ p.u.$

- **Non vital load at bus 6:**

$g_{nvl2} = 0.05\ p.u.$, $b_{nvl2} = \text{-}0.0375\ p.u.$

- **Vital load :**

The vital loads, fed from the DC bus are assumed to be power unity factor corrected.

$$P_{v} = 2(\cos + i\sin\theta) = 2MW\ (SI)$$

$$P_{vu} = \frac{P_{v}}{Sbase} = \frac{1}{20}\ p.u.$$

Vital load at bus 3 and 6:

Appendix B: OptimalPolicy

Dynamic Programming Functions

The main function is OptimalPolicy which calls the function Findmin. Given the current state, FindMin computes the optimal control that will be applied over the next time interval.

FindMin

Define binary and real variables: BinaryVars, RealVars
Split Inequalities into eqBinary, eqReal, binary equations contain only binary variables, real equations can contain both binary and real variables
GIVEN RealStateVars, BinaryStateVars, BinaryVars, RealVars, eqBinary, eqReal, J = cost as (interpolation) function of (RealStateVars, BinaryVars, RealVars, AltBinaryState)
Use Reduce to get all feasible binary combinations. Note: In this case there are 2^8=256 possible combinations of binary values, but only 8 are feasible
Use Reduce to solve for real variables for each feasible combination of binary variables - the result will be unique for each feasible combination of binary variables
Evaluate J for each pair of binary & real & select minimum.

FindMin comes in two flavors, the first is defined above. The second has all solutions, both binary and real as input.

```
 Clear[FindMin];
FindMin[ListSolsBinary_List,EqReal_List,BinaryVars_List,RealVars_List,RealStateVars_List,BinaryStateVars_List,x_List,AltBinaryState_,NextBinaryState_,J_]:=Module[{StateReplace,SolsBinary,SolsReal,Sols,Cost,Ind,A,RR,SolAlt},
    StateReplace=Inner[Rule,RealStateVars,Drop[x,-1],List];
    RR=True;
    Map[(RR=RR&&#)&,EqReal];
SolsReal=Map[Resolve[Exists[RealVars,Rationalize[(RR//.StateReplace)]//.#],Reals]&,ListSolsBinary[[x[[-1]]]]];
    (* SolsBinary is solved and the result is substituted in the EqReal. Similarily EqReal now contained only real variable is ratinalized and reduced *)
    Ind=Position[SolsReal,True];
    (* Identify positions where there are real solution. *)
```

```
SolsReal=N[Map[Reduce[Rationalize[(EqReal//.StateReplace)]//.#,RealVars,Reals]&,ListSolsBinary[[x[[-1]]]][[Flatten[Ind]]]]];
        A=ListSolsBinary[[x[[-1]]]][[Flatten[Ind]]];
    (*  Take the corrsponding binary solutions *)
Sols=MapThread[Join[#1,#2]&,{A,Map[Solve[#,RealVars][[1]]&,SolsReal]}];
     (* Pair up the valid binary and real solutions *)
    (* Set up a rule to switch to the alternate binary state representation *)
SolAlt=Map[{AltBinaryState→Position[NextBinaryState/.#,1][[1]]}&,Sols];
    (* SolAlt select the NextBinaryState (if any) for which a transition occurs, ie it binary value is 1  *)
Cost=Flatten[MapThread[(((J//.StateReplace)//.#1)//.#2)&,{Sols,SolAlt}]];
    (* A mapping of J (cost) with the Sols, after SolAlt is substituted  *)
        Ind=Position[Cost,Min[Cost]]; (* assign a position to the minimum cost*)
       (*  If[Length[Ind]>1,Print["Warning: Multiple minima found! Indices = "<>ToString[Ind]]; *)
        {Cost[[Ind[[1]]]],Sols[[Ind[[1]]]]}
    (* Select the cost value and the solution Sols corresponding to the minimum cost*)
    ];

FindMin[ListSolsAll_List,x_List,AltBinaryState_,NextBinaryState_,J_]:=Module[{StateReplace,Cost,Ind,SolAlt},
    StateReplace=Inner[Rule,RealStateVars,Drop[x,-1],List]; (* Pull out real state from x *)
    (* Set up a rule to switch to the alternate binary state representation *)
SolAlt=Map[{AltBinaryState→Position[NextBinaryState/.#,1][[1]]}&,ListSolsAll];
    (* Compute cost associated with each admissible mode change *)
Cost=Flatten[MapThread[(((J//.StateReplace)//.#1)//.#2)&,{ListSolsAll,SolAlt}]];
    Ind=Position[Cost,Min[Cost]]; (* identify positions of minimum cost*)
    Print["Cost ="];
    Print[Cost];
    (*  If[Length[Ind]>1,Print["Warning: Multiple minima found! Indices = "<>ToString[Ind]]; *)
        {Cost[[Ind[[1]]]],ListSolsAll[[Ind[[1]]]]}
    (* Select the cost value and the solution Sols corresponding to the minimum cost*)
    ];
```

```
General::spell : Possible spelling error: new symbol name "Cost" is similar to existing symbols {Cos, Cosh, Cot, Most}. More…
```

```
Clear[FindMinDAE];
FindMinDAE[ListSolsAll_List,x_List,AltBinaryState_,NextBinaryState_,DaeDynamics_,y0_,G_,J_]:=Module[{StateReplace,Cost,Ind,SolAlt,BinaryStateRules,NextRules,CurrentRules},
    (* Set up rule to define current real state variables. *)
    StateReplace=Inner[Rule,RealStateVars,Drop[x,-1],List]; (* Pull out real state from x *)
    (* Set up rules to define the next binary state in the alternate binary state representation *)
    SolAlt=Map[{AltBinaryState→Position[NextBinaryState/.#,1][[1]]}&,ListSolsAll];
    (* Set up rules that define next binary state *)
    BinaryStateRules=Map[Inner[Rule,NextBinaryState,NextBinaryState/.#,List]&,ListSolsAll];
    (* Set up rules to define all next real variables. *)
    CurrentRules=Map[Flatten[ReleaseHold[DaeDynamics[Drop[x,-1],y0,#,0]]]&,BinaryStateRules];
    NextRules=Map[Flatten[ReleaseHold[DaeDynamics[Drop[x,-1],y0,#,1]]]&,BinaryStateRules];
    (* Compute cost associated with each admissible mode change *)
    Cost=Flatten[MapThread[(((((G//.StateReplace)//.#1)//.#2)//.#3)+((((J//.StateReplace)//.#1)//.#2)//.#4))&,{ListSolsAll,SolAlt,CurrentRules,NextRules}]];
    Ind=Position[Cost,Min[Cost]]; (* identify positions of minimum cost*)
    (*  If[Length[Ind]>1,Print["Warning: Multiple minima found! Indices = "<>ToString[Ind]]; *)
      {Cost[[Ind[[1]]]],ListSolsAll[[Ind[[1]]]]}
    (* Select the cost value and the solution Sols corresponding to the minimum cost*)
    ];
```

Auxilliary Functions

The problem real variables are always known from the problem definition. However, the bianry variables need to be identified after the IP fromulas are determined. Here are two functions that generate the binary variables, the binary equations and the real equations and a simple one that generates the real equations after binary equations are identified.

```
BinaryEquations[IPFormulas_List,RealVars_List]:=Module[{DropReals,AA=IPFormulas},
DropReals[AA_List,i_]:=AA[[Flatten[Position[Map[(Cases[Variables[Level[#,1]],i])&,AA],{}]]]];
      Map[(AA=DropReals[AA,#])&,RealVars];
      AA
      ];
RealEquations[IPFormulas_,EqBinary_]:=Complement[IPFormulas,EqBinary];
BinaryVariables[IPFormulas_,RealVars_]:=Module[{AA=IPFormulas},Complement[Union[Flatten[Map[(Variables[Level[#,1]])&,AA]]],RealVars]];
ControlRules[ControlLogic_,StateReplaceList_,DiscreteState_,InputEvents_]:=Module[{ControlIP,F,vars,ControlEqns,CombinedList,ControlReplacements},
ControlIP=Map[GenIP[#,Join[DiscreteState,InputEvents]]&,ControlLogic];
     F[x_]:=Union[Flatten[Map[Variables[Level[#,1]]&,x]]];
vars=Map[F[#]&,MapThread[#1/.#2&,{ControlIP,StateReplaceList}]];
     (* MapThread[Print[#1/.#2,#3]&,{ControlIP,StateReplaceList,vars}]; *)
ControlEqns=MapThread[Reduce[#1/.#2,#3,Integers]&,{ControlIP,StateReplaceList,vars}];
     CombinedList=MapThread[{#1,#2}&,{ControlEqns,vars}];
ControlReplacements=Map[Solve[#[[1]],#[[2]]]&,CombinedList];
     ControlReplacements
     ]
 Clear[BinarySolutions];
BinarySolutions[EqBinary_,StateReplaceList_,ControlReplacements_]:=Module[{MyEqns,F,Ans,G,Sols,H},
MyEqns=MapThread[(EqBinary//.#1)//.#2&,{StateReplaceList,ControlReplacements}];
F[x_]:=Map[Reduce[#,Union[Flatten[Map[Variables[Level[#,1]]&,#]]],Integers]&,x];
     Ans=Map[F[#]&,MyEqns];
     (* x corresponds to discrete state results (Ans), y to control (ControlReplacements) *)
```

```
    G[x_,y_]:=Module[{},
        H[k_]:=Module[{DDD,EEE},
DDD=Map[Flatten[Solve[#,Variables[Level[#,2]]]]&,Level[x[[k]],1]];
            EEE=Map[Join[#,y[[k]]]&,DDD]
            ];
        Map[H[#]&,Range[Length[y]]]
        ];
MapThread[Flatten[G[#1,#2],1]&,{Ans,ControlReplacements}]
    ];
BinarySolutions[EqBinary_,StateReplaceList_]:=Module[{MyEqns,F,Ans,G,Sols,H},
    MyEqns=Map[(EqBinary/.#1)&,StateReplaceList];
Ans=Map[Reduce[#,Union[Flatten[Map[Variables[Level[#,1]]&,#]]],Integers]&,MyEqns];
    Map[Solve[#,Variables[Level[#,2]]]&,Ans]
    ];
 Clear[RealSolutions]
RealSolutions[ListSolsBinary_List,EqReal_List,BinaryVars_List,RealVars_List,RealStateVars_List,BinaryStateVars_List,x_List]:=Module[{StateReplace,SolsReal,Sols,Ind,A,RR},
    (* Given full state x, and list of binary solutions, find real solutions. RealSolutions will be mapped over S. *)
    StateReplace=Inner[Rule,RealStateVars,Drop[x,-1],List]; (* Pull out real state from x *)
    RR=True;
    Map[(RR=RR&&#)&,EqReal]; (* Convert real equation list to "&&" *)
SolsReal=Map[Resolve[Exists[RealVars,Rationalize[(RR//.StateReplace)]//.#],Reals]&,ListSolsBinary[[x[[-1]]]]];
    (* For each binary solution associated with given discrete state, substitute current real state into EqReal and check if next real state exists using Resolve *)
    Ind=Position[SolsReal,True];
    (* Identify positions where there are real solutions. *)
SolsReal=N[Map[Reduce[Rationalize[(EqReal//.StateReplace)]//.#,RealVars,Reals]&,ListSolsBinary[[x[[-1]]]][[Flatten[Ind]]]]];
    (* use reduce to find real solutions if where they exist *)
       A=ListSolsBinary[[x[[-1]]]][[Flatten[Ind]]];
    (*  Take the corrsponding binary solutions *)
Sols=MapThread[Join[#1,#2]&,{A,Map[Solve[#,RealVars][[1]]&,SolsReal]}];
     (* Pair up the valid binary and real solutions *)
    Sols
    ]
Clear[AllSolutions]
AllSolutions[ListSolsBinary_List,EqReal_List,BinaryVars_L
ist,RealVars_List,RealStateVars_List,BinaryStateVars_List,
S_List]:=Map[RealSolutions[ListSolsBinary,EqReal,BinaryVar
s,RealVars,RealStateVars,BinaryStateVars,#]&,S]

AllSolutions[ListSolsBinary_List,EqReal_List,BinaryVars_Li
st,RealVars_List,RealStateVars_List,BinaryStateVars_List,C
ontinuousS_List,DiscreteS_List]:=Module[{RSIndex,ShortEqua
tions,SBar,S2SBar,Sols,ExSols={}},

{RSIndex,ShortEquations}=ReducedState[EqReal,RealStateVars
];

{SBar,S2SBar}=StateSpaceProjection[ContinuousS,DiscreteS,R
SIndex];

Sols=Map[RealSolutions[ListSolsBinary,ShortEquations,Binar
yVars,RealVars,RealStateVars[[RSIndex]],BinaryStateVars,#]
&,SBar];
    ExSols=Map[Join[ExSols,Sols[[#]]]&,S2SBar];
    ExSols
    ]
```

State Space Projection

Identify the Required States

ReduceState has two arguments: 1) the list of the real equations, and 2) the list of real state variables. It returns a list of indices for the state variables that appear in the real equations.

```
 Clear[ReducedState];
ReducedState[RealEquations_,RealStateVariables_]:=Module[{
 ConstraintIndexList,ShortEquations,TempVars,RS,RSIndex,Dro
 pState,DropIndex,DropPosition},

 ConstraintIndexList=Flatten[Position[Map[ConstraintQ[#]&,R
 ealEquations],True]];
    ShortEquations=RealEquations[[ConstraintIndexList]];
    TempVars=Variables[Level[ShortEquations,{-1}]];
    RS=Intersection[TempVars,RealStateVars];
    DropState=Complement[RealStateVars,RS];
    RSIndex=Flatten[Map[Position[RealStateVars,#]&,RS]];
    DropList=Map[_≤#≤_&,DropState];

 DropPosition=Flatten[Map[Position[RealEquations,#]&,DropLi
 st]];

 {Flatten[Map[Position[RealStateVars,#]&,RS]],Drop[RealEqua
 tions,DropPosition]}
    ]
```

Projection

StateSpaceProjection has three arguments: 1) continuoues state space, 2) discrete state space, and 3) the reduced continuous state indices. It returns two quantities: 1) the reduced state space, and 2) and index list that maps elements of the state space to the corresponding elements of the reduced state space.

```
Clear[StateSpaceProjection];
StateSpaceProjection[ContinuousS_,DiscreteS_,ReducedStateIndex_]:=Module[{ReducedContinuousS,SPre,SBar},
ReducedContinuousS=Map[#[[ReducedStateIndex]]&,ContinuousS];
SPre=Map[Flatten[#]&,Flatten[Outer[List,ReducedContinuousS,DiscreteS,1],1]];
    SBar=Union[SPre];
    IndexList=Flatten[Map[Position[SBar,#]&,SPre]];
    {SBar,IndexList}
    ]
```

OptimalPolicy

For now, we assume that we seek binary-valued controls, the optimal policy is the control vector, ustar, determined as a tabular function over the state space, *S*. The state is composed of the real state variables, RealStateVars, and the discrete state variables. RealStateVars is simply a vector of length *n*, where *n* is the dimension of the real state space, *X*. We use two different representations of the discrete state space. Suppose there are m modes of the system, labeled . One representation of the discrete state is a list (vector) δ={ , } where each is either 0 if the system is not in that state or 1 if it is. Thus there is precisely one element equal to one and all of the others are zero. We denote the space of all δ's, Δ. The state space is $\Sigma=X\times\Delta$.

The other representation is simply {i} where i is the index such that =1. Let *Z* denote the integers 1,2,... The state space *S* is the product space $S=X\times Z$. It is easy to see that $S\Leftrightarrow\Sigma$. Clearly, given δ, i=Position[δ,1]. or given i, construct δ={ , }.

To use OptimalPolicy it is necessary to define the following arguments:

eqBinary, eqReal, BinaryVars, RealVars, BinaryStateVars, NextBinaryState, RealStateVars, DynamicsRHS, BinaryControlVars, S, GN, G, N

ContinuousS = Outer[List, Range[0.1, 1.9, 0.01]];
DiscreteS = IdentityMatrix[2];
DiscreteS = Range[2];
State space, $S = X \times Z$ = Map[Flatten[#] &, Flatten[Outer[List, ContinuousS, DiscreteS, 1], 1]];
Integrated cost, G[BinaryVars_List, RealVars_List, StateVars_List, n_Integer] := (RealVars[[1]] / (1 / n))^10 / m;
Terminal cost, GN[StateVars_List] := (0.5 (StateVars[[1]] − qbar)^2);
BinaryStateVars = {δ_{q1}, δ_{q2}};
NextBinaryState = {δ_{qq1}, δ_{qq2}};
BinaryControlVars = {δ_{s}};
RealStateVars = {q};
RealVars = {i, z};
AllRealVars = {i, q, z};
BinaryVars = Complement[BinaryVariables[AA, AllRealVars], BinaryStateVars];
DynamicsRHS = {z};
Number of steps, N

Control=OptimalPolicy[eqBinary,eqReal,BinaryVars,RealVars,BinaryStateVars,NextBinaryState,RealStateVars,DynamicsRHS,BinaryControlVars,S,GN,G,N];

```
Options[OptimalPolicy]={PrecomputeRealSolutions→True};
Clear[OptimalPolicy];
Clear[OptimalPolicy1];
Clear[OptimalPolicy2];
OptimalPolicy[EqBinary_List,EqReal_List,ControlLogic_,BinaryVars_List,RealVars_List,BinaryStateVars_List,BooleanStateVars_List,NextBinaryState_List,RealStateVars_List,DynamicsRHS_,BinaryControlVars_List,BooleanControlVars_List,S_,GN_,G_,n_Integer,opts___]:=If[PrecomputeRealSolutions{/.opts}/.Options[OptimalPolicy],OptimalPolicy2[EqBinary,EqReal,ControlLogic,BinaryVars,RealVars,BinaryStateVars,BooleanStateVars_List,NextBinaryState,RealStateVars,DynamicsRHS,BinaryControlVars,BooleanControlVars,S,GN,G,n],OptimalPolicy1[EqBinary,EqReal,ControlLogic,BinaryVars,RealVars,BinaryStateVars,BooleanStateVars_List,NextBinaryState,RealStateVars,DynamicsRHS,BinaryControlVars,BooleanControlVars,S,GN,G,n]];
OptimalPolicy1[EqBinary_List,EqReal_List,ControlLogic_,BinaryVars_List,RealVars_List,BinaryStateVars_List,BooleanStateVars_List,NextBinaryState_List,RealStateVars_List,DynamicsRHS_,BinaryControlVars_List,BooleanControlVars_List,S_,GN_,G_,n_Integer]:=Module[{SR,Data,Jstar,StateVars,NextState,AltBinaryState,sol1,OptSol,Delta,del,StateReplaceList,ControlReplacements,ListSolsBinary,ustar,ustarList={}},
    (* Construct data table in the form {{x1,x2,..,xn,Jstar} .....} *)
Data=Map[Join[S[[#]],{Map[GN[#]&,S][[#]]}]&,Range[Length[S]]];
    (* Data = a list of the states S paired with corresponding values of GN (the terminal cost) *)
    (* Create function Jstar(x1,x2,...,xn) *)
    Jstar=Interpolation[Data,InterpolationOrder → 1];
    (* BINARY Solutions - solve binary IP's for all discrete states *)
    (* Create an array, Delta, of elements del[i] of the length of BinaryStateVars*)
     Delta=Array[del,Length[BinaryStateVars]];
    (* initialization of all elements del to 0 *)
    Map[(del[#]=0)&,Range[Length[Delta]]];
StateReplaceList=Map[Inner[Rule,BinaryStateVars,ReplacePart[Delta,1,#],List]&,Range[Length[BinaryStateVars]]];
ControlReplacements=ControlRules[ControlLogic,StateReplaceList,BooleanStateVars,BooleanControlVars];
ListSolsBinary=BinarySolutions[EqBinary,StateReplaceList,ControlReplacements];
    (* MAIN Loop, Iterate Backward in Time *)
    For[r=n,r≥1,r--,
         StateVars=Join[RealStateVars,BinaryStateVars];(* The set of all state variables *)
         NextState=Join[DynamicsRHS,{AltBinaryState}];
       (* The set of dynamic variable and NextBinaryState (if any) for which a transition occurs, ie it binary value is 1*)
         sol1[x_]:=FindMin[ListSolsBinary,EqReal,BinaryVars,RealVars,RealStateVars,BinaryStateVars,x,AltBinaryState,NextBinaryState,G[BinaryVars,RealVars,StateVars,r]+Apply[Jstar,NextState]];
       (* the function sol1 call the function FindMin given it full state x *)
         OptSol=Map[sol1[#]&,S];(* map soll - FindMin- over S( continuous and discrete) and find the minimum  at each state *)
   Print["Finished FindMin. r = "];Print[r];
Data=Map[Join[S[[#]],{OptSol[[Range[Length[S]],1]][[#]][[1]]}]&,Range[Length[S]]];
         Jstar=Interpolation[Data,InterpolationOrder → 1];
ustar=Map[Inner[Rule,BinaryControlVars,Flatten[BinaryControlVars//.OptSol[[#,2]]],List]&,Range[Length[S]]];
       Print[ustar];
       ustarList=Join[ustarList,{ustar}];
       ];
    Simplify[ustarList]
    ]
```

Syntax :: sntxf : "{" cannot be followed by "/. opts }". More…

```
OptimalPolicy[EqBinary_List,EqReal_List,ControlLogic_,BinaryVars_List,RealVars_List,BinaryStateVars_List,BooleanStateVars_List,NextBinaryState_List,RealStateVars_List,DynamicsRHS_,BinaryControlVars_List,BooleanControlVars_List,S_,GN_,G_,n_Integer,opts___]:=If[PrecomputeRealSolutions{/.opts}/.Options[OptimalPolicy],OptimalPolicy2[EqBinary,EqReal,ControlLogic,BinaryVars,RealVars,BinaryStateVars,BooleanStateVars_List,NextBinaryState,RealStateVars,DynamicsRHS,BinaryControlVars,BooleanControlVars,S,GN,G,n],OptimalPolicy1[EqBinary,EqReal,ControlLogic,BinaryVars,RealVars,BinaryStateVars,BooleanStateVars_List,NextBinaryState,RealStateVars,DynamicsRHS,BinaryControlVars,BooleanControlVars,S,GN,G,n]];
Clear[OptimalPolicy2];
```

```
OptimalPolicy2[EqBinary_List,EqReal_List,ControlLogic_,BinaryVars_List,RealVars_List,BinaryStateVars_List,BooleanStateVars_List,NextBinaryState_List,RealStateVars_List,DynamicsRHS_,BinaryControlVars_List,BooleanControlVars_List,S_,GN_,G_,n_Integer]:=Module[{SR,Data,Jstar,StateVars,NextState,AltBinaryState,sol1,OptSol,Delta,del,StateReplaceList,ControlReplacements,ListSolsBinary,ustar,ustarList={},ListSolsAll},
    (* Construct data table in the form {{x1,x2,..,xn,Jstar} .....} *)
Data=Map[Join[S[[#]],{Map[GN[#]&,S][[#]]}]&,Range[Length[S]]];
    (* Data = a list of the states S paired with corresponding values of GN (the terminal cost) *)
    (* Create function Jstar(x1,x2,...,xn) *)
    Jstar=Interpolation[Data,InterpolationOrder → 1];
    (* BINARY Solutions - solve binary IP's for all discrete states *)
    (* Create an array, Delta, of elements del[i] of the length of BinaryStateVars*)
     Delta=Array[del,Length[BinaryStateVars]];
    (* initialization of all elements del to 0 *)
    Map[(del[#]=0)&,Range[Length[Delta]]];
StateReplaceList=Map[Inner[Rule,BinaryStateVars,ReplacePart[Delta,1,#],List]&,Range[Length[BinaryStateVars]]];
ControlReplacements=ControlRules[ControlLogic,StateReplaceList,BooleanStateVars,BooleanControlVars];
ListSolsBinary=BinarySolutions[EqBinary,StateReplaceList,ControlReplacements];
ListSolsAll=AllSolutions[ListSolsBinary,EqReal,BinaryVars,RealVars,RealStateVars,BinaryStateVars,S];
    Print["Toatal number of solutions = "<>ToString[Length[Flatten[ListSolsAll]]]];
    (* MAIN Loop, Iterate Backward in Time *)
    For[r=n,r≥1,r--,
          StateVars=Join[RealStateVars,BinaryStateVars];  (* The set of all state variables *)
          NextState=Join[DynamicsRHS,{AltBinaryState}];
       (* The set of dynamic variable and NextBinaryState (if any) for which a transition occurs, ie it binary value is 1*)
       (* sol1[x_]:=FindMin[ListSolsBinary,EqReal,BinaryVars,RealVars,RealStateVars,BinaryStateVars,x,AltBinaryState,NextBinaryState,G[BinaryVars,RealVars,StateVars,r]+Apply[Jstar,NextState]]; *)
```

```
        (* In sol1[x,y], x is the full state and y is the corresponding solution list from which FindMin seeks the one with least cost *)
           sol1[x_,y_List]:=FindMin[y,x,AltBinaryState,NextBinaryState,G[BinaryVars,RealVars,StateVars,r]+Apply[Jstar,NextState]];
        (* OptSol pairs up states with corresponding solutions and applies sol1 to each pair *)
           OptSol=MapThread[sol1[#1,#2]&,{S,ListSolsAll}];
   Print["Finished FindMin. r = "];Print[r];
Data=Map[Join[S[[#]],{OptSol[[Range[Length[S]],1]][[#]][[1]]}]&,Range[Length[S]]];
           Jstar=Interpolation[Data,InterpolationOrder → 1];
ustar=Map[Inner[Rule,BinaryControlVars,Flatten[BinaryControlVars//.OptSol[[#,2]]],List]&,Range[Length[S]]];
        (* Print[ustar]; *)
        ustarList=Join[ustarList,{ustar}];
        ];
      Simplify[ustarList]
      ]
```

```
General::spell1 : Possible spelling error: new symbol name "Data" is similar to existing symbol "Date". More…
```

```

Clear[OptimalPolicy];
OptimalPolicy[EqBinary_List,EqReal_List,ControlLogic_,BinaryVars_List,RealVars_List,BinaryStateVars_List,BooleanStateVars_List,NextBinaryState_List,RealStateVars_List,DynamicsRHS_,BinaryControlVars_List,BooleanControlVars_List,ContinuousS_,DiscreteS_,GN_,G_,n_Integer]:=Module[{SR,Data,Jstar,StateVars,NextState,AltBinaryState,sol1,OptSol,Delta,del,StateReplaceList,ControlReplacements,ListSolsBinary,ustar,ustarList={},ListSolsAll,S2SBar,S},S=Map[Flatten[#]&,Flatten[Outer[List,ContinuousS,DiscreteS,1],1]];
       (* Construct data table in the form {{x1,x2,...,xn,Jstar} .....} *)
Data=Map[Join[S[[#]],{Map[GN[#]&,S][[#]]}]&,Range[Length[S]]];
       (* Data = a list of the states S paired with corresponding values of GN (the terminal cost) *)
       (* Create function Jstar(x1,x2,...,xn) *)
       Jstar=Interpolation[Data,InterpolationOrder → 1];
       (* BINARY Solutions - solve binary IP's for all discrete states *)
        (* Create an array, Delta, of elements del[i] of the length of BinaryStateVars*)
          Delta=Array[del,Length[BinaryStateVars]];
        (* initialization of all elements del to 0 *)
        Map[(del[#]=0)&,Range[Length[Delta]]];
StateReplaceList=Map[Inner[Rule,BinaryStateVars,ReplacePar
t[Delta,1,#],List]&,Range[Length[BinaryStateVars]]];
ControlReplacements=ControlRules[ControlLogic,StateReplace
List,BooleanStateVars,BooleanControlVars];
ListSolsBinary=BinarySolutions[EqBinary,StateReplaceList,C
ontrolReplacements];
ListSolsAll=AllSolutions[ListSolsBinary,EqReal,BinaryVars,
RealVars,RealStateVars,BinaryStateVars,ContinuousS,Discret
eS];
        (* MAIN Loop, Iterate Backward in Time *)
        For[r=n,r≥1,r--,
             StateVars=Join[RealStateVars,BinaryStateVars];
(* The set of all state variables *)
             NextState=Join[DynamicsRHS,{AltBinaryState}];
          (* The set of dynamic variable and
NextBinaryState (if any) for which a transition occurs, ie
it binary value is 1*)
          (* sol1[x_
]:=FindMin[ListSolsBinary,EqReal,BinaryVars,RealVars,RealS
tateVars,BinaryStateVars,x,AltBinaryState,NextBinaryState,
G[BinaryVars,RealVars,StateVars,r]+Apply[Jstar,NextState]]
; *)
          (* In sol1[x,y], x is the full state and y is the
corresponding solution list from which FindMin seeks the
one with least cost *)
             sol1[x_,y_List
]:=FindMin[y,x,AltBinaryState,NextBinaryState,
G[BinaryVars,RealVars,StateVars,r]+Apply[Jstar,NextState]]
;
          (* OptSol pairs up states with corresponding
solutions and applies sol1 to each pair *)
OptSol=MapThread[sol1[#1,#2]&,{S,ListSolsAll}];
   Print["Finished FindMin. r = "];Print[r];
Data=Map[Join[S[[#]],{OptSol[[Range[Length[S]],1]][[#]][[1
]]}]&,Range[Length[S]]];
             Jstar=Interpolation[Data,InterpolationOrder →
1];
ustar=Map[Inner[Rule,BinaryControlVars,Flatten[BinaryContr
olVars//.OptSol[[#,2]]],List]&,Range[Length[S]]];
          (* Print[ustar]; *)
          ustarList=Join[ustarList,{ustar}];
          ];
        Simplify[ustarList]
        ]
```

Null[2]

```
Clear[OptimalPolicyDAE];OptimalPolicyDAE[EqBinary_List,EqReal_List,BinaryVars_List,RealVars_List,RealVarsIP_List,BinaryStateVars_List,BooleanStateVars_List,NextBinaryState_List,RealStateVars_List,DaeDynamics_,y0_,BinaryControlVars_List,BooleanControlVars_List,ContinuousS_,DiscreteS_,GN_,G_,n_Integer]:=Module[{SR,Data,Jstar,StateVars,NextState,AltBinaryState,sol1,OptSol,Delta,del,StateReplaceList,ControlReplacements,ListSolsBinary,ustar,ustarList={},ListSolsAll,S2SBar,S},
S=Map[Flatten[#]&,Flatten[Outer[List,ContinuousS,DiscreteS,1],1]];
    (* Construct data table in the form {{x1,x2,..,xn,Jstar} .....} *)
Data=Map[Join[S[[#]],{Map[GN[#]&,S][[#]]}]&,Range[Length[S]]];
    (* Data = a list of the states S paired with corresponding values of GN (the terminal cost) *)
    (* Create function Jstar(x1,x2,...,xn) *)
    Jstar=Interpolation[Data,InterpolationOrder → 1];
    (* BINARY Solutions - solve binary IP's for all discrete states *)
    (* Create an array, Delta, of elements del[i] of the length of BinaryStateVars*)
     Delta=Array[del,Length[BinaryStateVars]];
    (* initialization of all elements del to 0 *)
    Map[(del[#]=0)&,Range[Length[Delta]]];
StateReplaceList=Map[Inner[Rule,BinaryStateVars,ReplacePart[Delta,1,#],List]&,Range[Length[BinaryStateVars]]];
ListSolsBinary=BinarySolutions[EqBinary,StateReplaceList];
ListSolsAll=AllSolutions[ListSolsBinary,EqReal,BinaryVars,RealVarsIP,RealStateVars,BinaryStateVars,ContinuousS,DiscreteS];
    (* MAIN Loop, Iterate Backward in Time *)
    For[r=n,r≥1,r--,
         StateVars=Join[RealStateVars,BinaryStateVars]; (* The set of all state variables *)
         (* NextState=Join[DynamicsRHS,{AltBinaryState}]; *)
       (* The set of dynamic variable and NextBinaryState (if any) for which a transition occurs, ie it binary value is 1*)
```

```
        (* In sol1[x,y], x is the full state and y is the corresponding IP solution list from which FindMin seeks the one with least cost *)
        sol1[x_,y_List]:=Module[{},
FindMinDAE[y,x,AltBinaryState,NextBinaryState,DaeDynamics,y0,G[BinaryVars,RealVars,StateVars,r],Apply[Jstar,Join[RealStateVars,{AltBinaryState}]]]
            ];
           (* OptSol pairs up states with corresponding solutions and applies sol1 to each pair *)
           OptSol=MapThread[sol1[#1,#2]&,{S,ListSolsAll}];
    Print["Finished FindMin. r = "];Print[r];
Data=Map[Join[S[[#]],{OptSol[[Range[Length[S]],1]][[#]][[1]]}]&,Range[Length[S]]];
           Jstar=Interpolation[Data,InterpolationOrder → 1];
ustar=Map[Inner[Rule,BinaryControlVars,Flatten[BinaryControlVars//.OptSol[[#,2]]],List]&,Range[Length[S]]];
        (* Print[ustar]; *)
        ustarList=Join[ustarList,{ustar}];
        ];
      Simplify[ustarList]
      ]
```

Controller Maps

The following functions are used to generate the lookup table that implements the optimal feedback controller. The main function is BuildLookupTable which opens and closes a textfile in which the lookup arrays are written. BuildLookupTable calls the function LookupTable that actually constructs and writes the arrays.

```
MyToString[X_List]:=Module[{B,BB},
    B=X[[1]];
    Map[(B=ToString[B]<>","<>ToString[#])&,Drop[X,1]];
    B
    ]
LookupTable[StateDimensions_,ControVar_,TableName_,Control_,ChannelName_]:=Module[{A,a,B,dRule,A1,Levels,TransLevels,ATest,MatrixNames,channel,ArrayStrings,OnesList},
    A=Array[a,StateDimensions];
dRule=MapThread[(#1→ControVar/.#2)&,{Flatten[A],Control}];
    A1=A/.dRule;
     OnesList=Position[A1,1];
ArrayStrings=Map[(ToString[TableName]<>"("<>MyToString[#]<>") = 1;")&,OnesList];
```

```
    WriteString[ChannelName,ToString[TableName]<>" = zeros("<>MyToString[StateDimensions]<>");\n"];
    Map[WriteString[ChannelName,#<>"\n"]&,ArrayStrings];
    ]
General ::spell1 : Possible spelling error : new symbol name "dRule " is similar to existing symbol "Rule ". More…
General ::spell1 : Possible spelling error : new symbol name "Levels " is similar to existing symbol "Level ". More…
BuildLookupTable[StateDimensions_List,BinaryControlVars_List,TableNameList_List,Control_,FileName_String]:=Module[{channel},
    channel=OpenWrite[FileName];
MapThread[LookupTable[StateDimensions,#1,#2,Control,channel]&,{BinaryControlVars,TableNameList}];
    Close[channel]
    ]
```

Vita

Edoe F. MENSAH

EDUCATION

Ph.D, Mechanical Engineering: Systems & Control 03/08

M.S., Math and Electrical Engineering: Option Dynamical systems & Control 06/99

Drexel University, Philadelphia PA

Thesis Title: Logic-Based Optimal Control for Shipboard Power System Management

Advisor: Prof Harry G. Kwatny

EMPLOYMENT AND RELATED EXPERIENCES

Instructor: Mathematics 09/99-06/01

Drexel University, Philadelphia, PA, Dept of Math and Computer Science

Courses: Calculus I, II, III and IV, Linear algebra, discrete mathematics

Teaching assistant

Dept of Mechanical Eng and Mechanics, Dept of Math and Computer Science

Courses

- Control Systems, Aircraft Dynamics, Engineering Reliability 09/04-03/08
- Calculus I, II, III for engineering students 09/96-06/99

AWARD and Fellowship

Joseph Carleone Endowed Fellowship in recognition of academic contributions 02/08

College of Engineering, Drexel University

The Albert Herr teaching assistants award for excellence in teaching 06/98

Mathematic and Computer Science Dept, Drexel University