NOTE TO USERS

This reproduction is the best copy available.

UMI®

FAULT DIAGNOSIS OF HYBRID SYSTEMS WITH APPLICATIONS TO GAS TURBINE ENGINES

Rasul Mohammadi

A Thesis

in

The Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of Doctor of Philosophy at

Concordia University

Montréal, Québec, Canada

April 2009

© Rasul Mohammadi, 2009

Library and Archives
Canada

Published Heritage
Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et
Archives Canada

Direction du
Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 978-0-494-63443-1
Our file Notre référence
ISBN: 978-0-494-63443-1

NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canada

ABSTRACT

FAULT DIAGNOSIS OF HYBRID SYSTEMS WITH APPLICATIONS TO GAS TURBINE ENGINES

Rasul Mohammadi, Ph.D.

Concordia University, 2009

Stringent reliability and maintainability requirements for modern complex systems demand the development of systematic methods for fault detection and isolation. Many of such complex systems can be modeled as hybrid automata. In this thesis, a novel framework for fault diagnosis of hybrid automata is presented. Generally, in a hybrid system, two types of sensors may be available, namely: continuous sensors supplying continuous-time readings (i.e., real numbers) and threshold sensitive (discrete) sensors supplying discrete outputs (e.g., level high and pressure low).

It is assumed that a bank of residual generators (detection filters) designed based on the continuous model of the plant is available. In the proposed framework, each residual generator is modeled by a Discrete-Event System (DES). Then, these DES models are integrated with the DES model of the hybrid system to build an Extended DES model. A "hybrid" diagnoser is then constructed based on the extended DES model. The "hybrid" diagnoser effectively combines the readings of discrete sensors and the information supplied by residual generators (which is based on continuous sensors) to determine the health status of the hybrid system.

The problem of diagnosability of failure modes in hybrid automata is also studied here. A notion of failure diagnosability in hybrid automata is introduced and it is shown that for the diagnosability of a failure mode in a hybrid automaton, it is sufficient that the failure mode be diagnosable in the extended DES model developed for representing the hybrid automaton and residual generators. The diagnosability of failure modes in the case that some residual generators produce unreliable outputs in the form of false alarm or false silence signals is also investigated. Moreover, the problem of isolator (residual generator) selection is examined and approaches are developed for computing a minimal set of isolators to ensure the diagnosability of failure modes.

The proposed hybrid diagnosis approach is employed for investigating faults in the fuel supply system and the nozzle actuator of a single-spool turbojet engine with an afterburner. A hybrid automaton model is obtained for the engine. A bank of residual generators is also designed, and an extended DES is constructed for the engine. Based on the extended DES model, a hybrid diagnoser is constructed and developed. The faults diagnosable by a purely DES diagnoser or by methods based on residual generators alone are also diagnosable by the hybrid diagnoser. Moreover, we have shown that there are faults (or groups of faults) in the fuel supply system and the nozzle actuator that can be isolated neither by a purely DES diagnoser nor by methods based on residual generators alone. However, these faults (or groups of faults) can be isolated if the hybrid diagnoser is used.

This thesis is dedicated to my wife Tanaz, who has supported me
unconditionally throughout the years of my study,
and our daughter Tina.

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my supervisor, Dr. S. Hashtrudi Zad, for his constant support, encouragement and guidance throughout the course of this thesis. Without his help, this work would not be possible. Special thanks to my co-supervisor, Dr. K. Khorasani, who shared some of his experience and offered valuable suggestions during my doctoral study.

I would like to extend my gratitude to all my thesis committee members, including Dr. A. Aghdam, Dr. A. Amer, Dr. B. Gordon, late Dr. P. Gohari, Dr. M. Lawford and Dr. S. Williamson. I sincerely appreciate the time, advice, and support they offered me throughout this endeavor.

Many thanks go to my good friends, especially, Nader Meskin, Mohsen Zamani Fekri, Ahmadreza Momeni, Sadegh Farzaneh, Amitabh Barua, Amin Mannani, Elham Semsar and Mohsen Azizi who helped make my stay in Montréal a very pleasant one.

Finally, I would like to thank my adorable wife, Tanaz, for all her love, support, patience, and encouragement. I am as ever, especially indebted to my parents for all of their never-ending support they have provided throughout my life.

TABLE OF CONTENTS

List of Figures

List of Tables

List of Abbreviations and Symbols

List of Abbreviations

AI	Artificial Intelligence
DES	Discrete-Event System
EDESA	Extended DES Abstraction
EFPRG	Extended Fundamental Problem in Residual Generation
EMA	Expectation Maximization Algorithm
FDI	Fault Detection and Isolation
FPRG	Fundamental Problem in Residual Generation
FSA	Finite-State Automata
FSM	Finite-State Machine
GEFPRG	Generalized Extended Fundamental Problem in Residual Generation
GLR	Generalized Likelihood Ratio
IMM	Interacting Multiple Model
I/O	Input/Output
LTI	Linear Time-Invariant
PLA	Power Level Angle
RPM	Revolution Per Minute
RPS	Revolution Per Second
RTS	Reachability Transition System
SMC	Sequential Monte Carlo

List of Symbols Used in Chapter 6

A_n	Nozzle area
A_n^{init}	Initial nozzle area
A_n^{max}	Maximum nozzle area
C	Heat capacity
C_a	Aircraft speed
C_n	Speed of the exiting gases
c_p	Specific heat capacity at constant pressure
c_{pa}	Specific heat capacity at constant pressure for air
c_p	Specific heat capacity at constant volume
c_{va}	Specific heat capacity at constant volume for air
c_{pg}	Specific heat capacity at constant pressure for the gases in the combustion chamber and afterburner
c_{vg}	Specific heat capacity at constant volume for the gases in the combustion chamber and afterburner
F	Thrust produced by the engine
h	Enthalpy
H	Aircraft altitude
H_u	Low calorific value of fuel
J	Rotor moment of inertia
M_a	Mach number in ambient temperature and pressure
m_g^{comb}	Mass of the gases in the combustion chamber
m_n	Mass of the gases in the nozzle
$\dot{m}_a$	Gas mass flow rate passing through the compressor
$\dot{m}_g$	Gas mass flow rate passing through the turbine
$\dot{m}_n$	Gas mass flow rate exiting the nozzle
$\dot{m}_f$	Main fuel mass flow rate

$\dot{m}_{fAB}$	Afterburner fuel mass flow rate
$\dot{m}_f^{max}$	Maximum of the main fuel flow rate
$\dot{m}_{fAB}^{max}$	Maximum of the afterburner fuel flow rate
N	Speed (RPS) of the shaft
p_a	Ambient pressure
p_{01}	Pressure at the compressor inlet
p_{02}	Pressure at the compressor outlet
p_{03}	Pressure at the turbine inlet
p_{04}	Pressure at the turbine outlet
p_{05}	Pressure at the nozzle inlet when the afterburner is in operation
PR_{comp}	Compressor pressure ratio
R	Specific gas constant
T_a	Ambient temperature
T_{01}	Temperature at the compressor inlet
T_{02}	Temperature at the compressor outlet
T_{03}	Temperature at the turbine inlet
T_{04}	Temperature at the turbine outlet
T_{05}	Temperature at the nozzle inlet when the afterburner is in operation
V_{comb}	Volume of the main combustion chamber
W	Power
W_c	Power consumed by the compressor
W_t	Power generated by the turbine
Δp_b	Percentage of pressure loss in the main combustion chamber
η_i	Isentropic intake efficiency

η_c	Isentropic compressor efficiency
η_t	Isentropic turbine efficiency
η_j	Isentropic nozzle efficiency
η_m	Mechanical transmission efficiency of the turbine
η_b	Efficiency of the main combustion chamber
η_{AB}	Efficiency of the afterburner combustion chamber
γ	Specific heat capacity ratio
γ_a	Specific heat capacity ratio for air
γ_g	Specific heat capacity ratio for the gases in the combustion chamber and afterburner
ρ_n	Density of the gases in the nozzle
ω	Angular velocity of the shaft

Chapter 1

INTRODUCTION

Modern complex systems demand high precision and reliability. Fault diagnosis is one of the important capabilities in complex systems for attaining high reliability. The use of extensive number of sensors with different types of signals in these systems requires systematic algorithms for fault monitoring and isolation. The large number of operational modes (e.g., 4^{80} modes of operation in Livingstone [109]) adds to the computational complexity of diagnosis algorithms and makes fault isolation a challenging problem. Therefore, one of the main issues in realizing modern complex systems is the development of systematic algorithms for fault detection and isolation. Using systematic methods for designing diagnosis systems not only increases the accuracy and reliability of diagnosis but also reduces the future costs of system revisions and maintenance. In addition, human errors are less likely in systematic diagnosis code generation than in manual code generation. Fault diagnosis systems play a very important role in aerospace, manufacturing and process industries. As a result, a large body of work has been conducted on fault diagnosis (see, e.g., [94, 47, 57]).

In order to develop systematic diagnosis algorithms for complex systems, a rigorous and precise model is very important. Many complex systems such as aircraft engines, satellites and spacecraft evolve both continuously and discretely, and hence, require modeling tools that take into account the interactions of continuous[1] and discrete-event dynamics. In the past two decades, a number of efforts have been made by researchers to develop modeling tools which are flexible to work with, and at the same time, reflect the complex behavior of modern complex systems. In particular, hybrid system models have been developed and used extensively for modeling complex systems [10].

Hybrid systems emerge from the interaction of discrete planning algorithms and continuous processes. The dynamics of a hybrid system in every mode evolves continuously until a transition takes the system to a different mode of operation. A transition in a hybrid system may take place autonomously as a result of the continuous evolution of system variables or because of a discrete event such as a supervisory command (a jump from one mode to another).

Hybrid systems have been employed extensively by researchers in different engineering fields, as well as by computer scientists. In engineering fields, hybrid systems have been used as a modeling tool for developing state-of-the-art algorithms in many domains such as control, verification, data management and fault diagnosis [11, 23, 64, 6, 115, 79].

Generally, in a hybrid system, two types of sensors are available, namely: **continuous sensors** supplying continuous-time readings (i.e., real numbers) and **threshold sensitive (discrete sensors)** supplying discrete outputs (e.g., level high and pressure low). Discrete outputs can be used for the diagnosis of drastic failures such as stuck-closed of a valve and continuous outputs can be used for the diagnosis of faults that slightly change the system dynamics such as a small loss-of-effectiveness in an actuator.

Purely discrete-event approaches for diagnosis in a hybrid system usually rely

[1] The term "continuous system" in this thesis refers to a system with continuous-time variables. For brevity, we use the term "continuous" instead of "continuous-time".

on an abstraction of continuous dynamics and do not have the detailed information supplied by continuous sensors. Diagnosis approaches based on DES models cannot isolate failures that manifest as small continuous variations in the system's behavior. Therefore, they are not suitable for diagnosis in many complex systems [45]. A purely continuous approach, on the other hand, may lead to very complex nonlinear relationships which are rather difficult to analyze in real-time. Moreover, due to the limitations on sensor implementation, some continuous variables may not be measurable and therefore, fault diagnosis based on purely continuous dynamics may not be always possible. Developing hybrid diagnosis algorithms which take advantage of both the high-level DES and the low-level continuous dynamics may be used to solve a larger number of problems.

In this thesis, we investigate fault diagnosis in systems that, for diagnostic purposes, can be modeled as hybrid systems. We are interested in hybrid systems because diagnosis problems not solvable in purely DES or purely continuous models may be solved using hybrid models. We develop a hybrid fault diagnosis framework in which the information available at the DES level is integrated systematically and efficiently with the information coming from the continuous dynamics through the continuous sensors. Since we do not only use abstracted models, we do not lose any information necessary for diagnosis due to abstraction. We assume that the system under supervision is operational and the fault detection and isolation system only uses the discrete outputs and continuous sensor readings in the system, and no test inputs are used for diagnosing faults. Thus, we only concentrate on **on-line passive** diagnosis.

We employ our hybrid diagnosis approach for investigating faults in the fuel supply system and the nozzle actuator of a single-spool turbojet engine with an afterburner. Faults in the fuel supply system and actuators have been the source of many failures in jet engines [104]. As an example of a fault in the fuel supply system, one can mention flight 236 of Air Transat [29]. Air Transat Flight 236 was an Air Transat route between Toronto, Canada and Lisbon, Portugal. On August 24, 2001, the flight ran out of fuel over the Atlantic Ocean with 306 people (293 passengers and 13 crew) aboard. The flight crew successfully landed the plane in the Azores with no loss of life. During the course of the flight, the pilots had noticed a fuel imbalance between the fuel tanks in the left and the right wings of the aircraft and had attempted to remedy this by opening a cross-feed valve between the tanks. This caused fuel from the operational tank to be wasted through the leak in the engine on the other side. After the engine flame-out, the airplane operated 19 minutes without engine power before landing.

Components in a fuel supply system and a nozzle actuator such as pumps and solenoid valves behave in a discrete-event manner. The status of these components varies when the operating regime of the engine changes. The discrete-event behavior of these components can be described by DES models. On the other hand, thrust generation in an engine is a continuous process, and operation of engine components such as compressor and turbine can be described by continuous static and dynamic thermodynamic relations (i.e., algebraic and differential equations). We show that there are cases when the faults in the fuel supply system and the nozzle actuator cannot be isolated by a purely DES diagnosis method or by continuous approaches based on residual generators (continuous fault diagnosis systems) alone. However, the fault can be isolated if our hybrid diagnosis framework is employed.

In the following, we briefly review the research conducted in the literature as related to the work in this thesis.

1.1 Literature Review

1.1.1 Hybrid Systems

Hybrid systems have been studied for a long time. Early models developed by system engineers for hybrid systems were mainly based on the *switched system* models [72, 111]. Development of results in control of discrete-event systems in 80's largely motivated the recent interest and activity in hybrid systems. Moreover, the development of adaptive control theory in 70's and 80's, digital control and the renewed interest in optimal control for sampled-data systems have a considerable impact on the recent trend of research in hybrid systems. Much of the work conducted on hybrid system modeling mostly concentrates on solving control problems (e.g., [10, 11, 22, 23, 51, 66]).

There are several approaches for modeling of hybrid systems. What makes these approaches different is the emphasis on the continuous and discrete-event dynamics. In general, hybrid systems have been studied extensively by computer scientists and system engineers. The models developed by computer scientists are extensions of Finite-State Machines (FSM) or Petri-nets to present more information of the system. On the other hand, system engineers are more interested in the continuous nature of hybrid systems. In the models developed by system engineers, the operation of a hybrid system is represented by a set of differential equations each corresponding to a mode of operation.

Hybrid Automata

In the early 90's, hybrid automata models [6, 9] were used in the computer science community for extending the traditional model checking of finite-state machines to real-time systems. Hybrid automata are generalizations of traditional FSMs. In a hybrid automaton, transitions among the states are conditioned on the value of logical propositions defined over a set of continuous dynamical processes. Hybrid automata models have been very influential on the research in the area of hybrid systems. In [77], Input/Output (I/O) hybrid automata are developed as an extension to hybrid automata. I/O hybrid automata can be combined into hybrid automata by synchronization.

Hybrid automata have been extensively used as a general modeling formalism for the analysis, verification and synthesis of hybrid systems [7, 52, 77]. They are the most conventional models being used by researchers for modeling hybrid systems [10]. We use hybrid automata as the modeling tool in our work. The finite-state automata model which is the discrete abstraction of hybrid automata is very close to the natural way a human describes discrete processes. Moreover, unified formal mathematical models are available for hybrid automata.

1.1.2 Fault Diagnosis

A **fault** refers to a non-permitted deviation in a system's behaviour from that required by the system specifications for a bounded or unbounded period of time [56]. In general, there are two types of faults: **permanent faults** and **nonpermanent faults**. Permanent faults are those faults that when they occur, the system remains in the faulty condition indefinitely. Nonpermanent faults are faults which have limited duration. After the occurrence of a non-permanent fault, the system may recover and return to the normal condition. These faults are usually caused by temporary malfunction of the system or some external interference. A broken valve can be an example of a permanent fault and a loose connection in an electric circuit may cause a non-permanent fault.

Fault diagnosis is the **detection** and **isolation** of faults after they occur (and before they possibly cause a catastrophe in the system). Fault diagnosis is very important in enhancing the reliability and productivity of complex systems. In the following, we briefly review the work done on fault diagnosis in the literature.

Hardware redundancy is one of the most commonly-used methods for fault diagnosis and fault tolerance. In this approach, multiple sensors are used for measuring each system variable. Then, a *voter* compares their outputs and determines the final value. If one of the sensors fails, the failure can be detected by comparing its value with other sensor values. This approach is also employed in diagnosing software code errors in the form of *N-version programming*. In N-version programming, multiple codes are provided for a critical part of the system. Usually, these codes are written in different programming languages by different programmers to avoid language, compiler and human related errors. Although, these techniques are simple and fairly reliable, they impose an overhead on the system, resulting in the increase of implementation cost. Moreover, they are only suitable for detecting sensor failures and programming errors and are not suitable for detecting common-cause failures.

Expert systems are also used for diagnosing failures. Expert systems are designed based on the experience and knowledge of *experts* (stored as a set of rules) and use an *inference engine* to diagnose failures. These systems are advantageous in cases that obtaining a model for the system is difficult. However, gathering the required expertise and information for building an expert system could be a hard and time consuming task. In addition, it may not be possible to evaluate the completeness of the expert data base. Hardware and software redundancy and expert systems are examples of diagnosis techniques which perform diagnosis without utilizing a model of the system and are therefore, known as **model-free** methods.

In addition to model-free methods, several **model-based** techniques for fault diagnosis have been proposed in the literature. In a model-based approach, the observed behaviour of the system is compared against the system model, and the condition of the system (normal/faulty) is inferred from this comparison.

In the following, we review the main directions of research on fault diagnosis that use model-based approaches.

Fault Diagnosis in Discrete-Event Systems

Fault diagnosis of discrete-event systems has been investigated in the context of automatic control systems (see, e.g., [73, 17, 98]) and in other areas such as Artificial Intelligence (AI) (e.g., [15]). In [73], F. Lin proposed a **state-based** approach for diagnosis failures in DES. In state-based approaches, it is assumed that the state set of the system can be partitioned according to the *condition* (failure status) of the system. The goal of the diagnosis process is to determine the current state of the system (or at least the block of the normal/faults partition the current state belongs to) using the available observations (sensor measurements) and then to determine the current condition of the system. In [73], the problems of *off-line* and *on-line active diagnosis* are addressed. An algorithm is presented for computing a sequence of test commands for diagnosing system failures. If the algorithm converges, the system will be *on-line diagnosable.*

In [98, 99], M. Sampath et al. present a systematic approach for *passive* on-line fault diagnosis in finite-state automata. In passive diagnosis, the diagnosis system does not generate any test inputs and relies on observations only. In [98], an extended observer for the system, called a **diagnoser**, is used to perform diagnosis. The issue of diagnosability is also addressed. The approach in [98] is **event-based**. In an event-based method, inference is made about *the occurrence of fault events* (based on the observed events). It is assumed that a fault is the result of an (unobservable) fault event. In [100], an integrated approach to fault diagnosis and supervisory control has been presented by generalizing the notion of diagnosis to active DES fault diagnosis.

In [71], algorithms for testing finite-state machines are reviewed. Although testing algorithms are related to the problem of fault diagnosis, the framework used in [71] is different: the finite-state machines are usually assumed to be deterministic with a fixed condition (failure status); also it is assumed that transitions can always be observed even if they do not result in a change in output. These assumptions often do not hold in fault diagnosis of control systems.

In [48], S. Hashtrudi Zad et al. study fault diagnosis in DES using a state-based approach. They proposed a passive on-line method for diagnosing failures in discrete event systems and construct a fault detection system (diagnoser). In this framework, the objective is to use the output sequence to determine the *current* normal/faulty condition of the system. The condition of the system does not have to be known at the time that the diagnoser is started. Assuming that a failure is diagnosable if it occurs before the diagnoser initialization, the proposed diagnoser can eventually detect and isolate the failure. A model reduction method has also been introduced in [48] to reduce the number of diagnoser states and the computational complexity of the diagnoser design.

All the above modelling approaches use automata to model DES. The complete system model can be generated using *synchronous* or *parallel composition* of component models (see e.g., [112]). An alternative approach based on a structured system description of DES models, called *causal network*, has been introduced in ([31, 84]) for fault diagnosis. This qualitative model seems to be suitable for process diagnosis in local power station plants [84]. This approach has a better computational efficiency for diagnosing discrete-event systems when compared to DES modeling techniques in [17, 73] and [98]. However, it also uses logical statements describing the status of each component, which is partly from human experience and expertise [84].

Timed models have been considered in fault diagnosis for cases where timing constraints can be used to improve the accuracy of diagnosis (see e.g., [91, 26, 36, 49]).

Fault Diagnosis in Continuous Systems

There is an extensive amount of research on fault detection and isolation (FDI) that uses models with continuous variables (see e.g., surveys [42, 110, 41, 44, 56], books [101, 94, 25] and the references therein). Most of the FDI approaches which employ continuous models rely on **analytical redundancy**. In these methods, the inherent redundancy existing in the static and dynamic relationships among the system inputs and measured outputs is used for fault detection and isolation. In other words, sensor measurements are compared with the values of the respective variables which are obtained analytically based on the mathematical model of the system. The resulting differences are called **residuals**. Then residuals are processed to determine which residuals can be considered normal and which ones indicate presence of a fault. When no fault is present in the system, the residual should be normally zero or very close to zero, and when a fault occurs, the residual should be distinguishably different from zero [25]. The algorithm or processor used to generate residuals is called a **residual generator**. The FDI methods based on analytical redundancy can be categorized as follows [41]:

1. Parity space approaches - In these approaches, the parity (consistency) of the mathematical equations of a dynamical system is verified by using sensor measurements (see e.g., [27, 95, 34]).

2. Dedicated observer approaches and innovation-based approaches - In these approaches, the outputs of the system are reconstructed from the measurements (or a subset of measurements) by using Luenberger observer(s) in a deterministic setting or Kalman filter(s) in a stochastic setting. The weighted output estimation error or innovations is used as the residual for the fault detection and isolation (see e.g, [94, 43, 93]).

3. Fault detection filter approaches - In these approaches, a special filter is constructed for detection and isolation of a fault or a set of faults. These approaches were first proposed by R. V. Beard [18] and H. L. Jones [61]. See also [81, 80] for fault detection and isolation filter design based on linear systems and [33] for design methods based on nonlinear models.

4. Parameter identification approaches - These approaches use the assumption that the faults of a dynamical system are reflected in the physical parameters such as mass, friction, resistance, etc. Estimation of the parameters of the mathematical model can be used for fault detection and isolation (see e.g., [56] for the first contributions and [101] for relatively more recent work).

1.1.3 Fault Diagnosis in Hybrid Systems

Diagnosis of hybrid systems, particularly diagnosis of hybrid automata, has been a subject of some research. However, the number of work done on fault diagnosis of hybrid automata is fewer than those in continuous systems and DES due to the complexity of hybrid systems and the relative novelty of the subject. In the following, we present some of the work done on diagnosis of hybrid systems and especially the work conducted on diagnosis of hybrid automata.

Fault diagnosis based on **discrete and/or temporal abstractions** of continuous dynamics is among one of the conventional approaches for fault diagnosis in hybrid systems. In [75], hybrid diagnosis based on timed discrete-event representations is studied. The continuous state of the system is quantized and discrete methods are used for diagnosis. The diagnosis method of [48] for FSMs has been extended to hybrid automata in [50]. In [50], the authors examine the question of whether a high-level DES contains enough information about the low-level hybrid model by introducing the notion of *consistency.* The *high-level* and *low-level* models are called consistent if the analysis and design based on the high-level model and the low-level model yield the same result. A set of sufficient conditions for consistency has also been provided. In [50], an output from a set of symbols is assigned to any state of the hybrid automaton. It is assumed that the output is constant at any discrete mode. Based on this assumption, the problem of diagnosis in hybrid automata has been reformulated as a DES diagnosis problem.

Diagnosis of hybrid system has been studied in [115] by abstracting the systems with a timed Petri-net model. A hybrid automaton model with linear first-order dynamics is considered. Using the model of the system, a fault symptom table is generated off-line by simulation and then compiled into a *decision tree*. The decision tree is used on-line for diagnosis. In this approach, the mode of the system is estimated by processing of the continuous variable signals. The Petri-net model is used to generate event predictions to focus the signal processing algorithms. Due to the use of decision tree, the approach is confined to the assumption of only one fault at a time.

The diagnosis method of [85] has been extended to hybrid systems in [78, 86]. *Temporal causal graphs* are used for modeling the abstractions of the continuous dynamics. Qualitative fault isolation algorithms are developed based on the temporal causal graphs. Faults are assumed to be *abrupt*; i.e., faults make instantaneous and persistent changes in the system (continuous) variables. A DES observer monitors the dynamics and notifies the diagnosis system of the fault occurrence. If qualitative diagnosis fails to isolate the fault, continuous parameter estimation techniques are used.

Fault diagnosis based on *Sequential Monte Carlo (SMC)* or *particle filtering* methods have been employed by researchers for diagnosis of hybrid systems, and mostly for isolating and identifying faults when the fault can be detected. In [79], a particle filtering approach is proposed to track multiple models of behavior of the system. An abstract model of the system dynamics is made in the form of a temporal causal graph. When a fault is detected, a (high-level) candidate qualitative diagnosis is made using the temporal causal graph. Using continuous model-fitting techniques such as the Expectation Maximization Algorithm (EMA) or Generalized Likelihood Ratio (GLR) method, the time of the fault occurrence and its severity are estimated. In [79], isolation and identification of faults are achieved by backtracking through the system operational modes using the system's observations. This makes storing all the observations necessary which could be problematic in complex systems due to the extensive number of sensors. In [79], it is also assumed that the hybrid system does not have any autonomous jump; all transitions between system modes are triggered by a control command. This assumption does not hold in many complex systems. Fault diagnosis in hybrid systems using particle filtering techniques has also been investigated in [68, 69].

In [39], a two level diagnosis mechanism is developed for diagnosis of faults whose occurrence can be sensed by measuring system variables at the time the faults occur. The occurrence of faults is signalled by observable events (actions) (different faults may yield the same events). When an event is observed, the diagnoser designed for the DES level performs a set of hypothesis tests and makes a diagnosis statement regarding the faults occurring in the DES level. Furthermore, the DES diagnoser generates a discrete state estimate of the system. Then, the diagnoser of the continuous level performs hypothesis tests (for example residual tests) of the discrete state estimate and generates sub-diagnosis statements regarding the faults at the continuous dynamics. A decision logic unit then produces the final diagnosis statement. A notion of diagnosability as related to the approach of [39] is presented in [38]. In [39, 38], it is implicitly assumed that discrete events occurring in the system are observable. This assumption cannot be held in many control applications. In addition, since occurrence of faults are signaled (for example by an ALARM signal) diagnosers are only responsible for the isolation of the faults.

A different approach for diagnosing a special class of hybrid systems is presented in [16]. Here, fault hypotheses are modeled using a Markov chain with a Gaussian residual associated with each state and a Viterbi-like algorithm is used to find the most likely state trajectory. This approach does not consider the event-driven dynamics that are present in hybrid systems.

1.1.4 Testing Diagnosability

Fault diagnosability is an important issue related to fault diagnosis. In the context of fault diagnosis using DES models, fault diagnosability has been investigated in [73, 98, 48, 114, 59]. The notion of diagnosability in DES has been introduced in [98]. A fault is considered **diagnosable** if it can be detected and isolated in a bounded number of events. Furthermore, a test for diagnosability based on the construction of a diagnoser has been presented in [98]. The number of diagnoser states, and hence the complexity of the above-mentioned test for diagnosability is in the worst case exponential in the number of the systems states. In [48], an approach similar to [98] is developed for diagnosability assuming that a faulty condition may be present when diagnosis starts. In [48], a fault is considered diagnosable if it can be detected and isolated after its occurrence or the start of diagnosis in a bounded number of events. The diagnosability conditions given in [48] are also in terms of the properties of a diagnoser and hence verifying them has a worst-case exponential complexity. In contrast to the exponential approaches of testing diagnosability in [98, 48], efficient polynomial algorithms for testing diagnosability (in the framework of [98]) are proposed in [114, 59].

Diagnosability of faults in diagnosis methods based on continuous dynamics of the system has also been a subject of research (see e.g., [80, 33]). Usually, in the context of diagnosis using continuous models, diagnosability is achieved by finding the conditions for the existence of a detection and isolation filter. In [80], the necessary and sufficient solvability conditions for the existence of residual generators for isolating faults in linear dynamical systems have been provided using a geometric approach. In linear systems, faults in the system can be modeled as *additive faults* that are added to the state-flow of the system as unknown inputs. In [80], it is shown that additive fault signals can be used to model the faults in actuators, sensors and the structure of systems. In [33], using a geometric approach, the necessary and sufficient solvability conditions for the existence of residual generators for isolating faults in nonlinear dynamical systems have been provided. In these systems, faults are modeled by additive signals.

In [35] and [105], diagnosability of hybrid systems is studied using a *timed automaton* abstraction of the hybrid system. Faults are only diagnosable if they change either the delay of the system in discrete states or the sequence of observed events.

1.1.5 Problems Related to Fault Diagnosis in Hybrid Systems

State estimation in hybrid systems - One of the approaches for monitoring and fault diagnosis of dynamical systems is based on the state estimation of the system using the observations. In [53], a hybrid estimation method is formulated as a k-best search using probabilistic hybrid automata models. The hybrid estimation technique is compared with the estimation methods based on Interacting Multiple Model (IMM) technique [20]. A state estimation approach based on banks of extended Kalman filters is presented in [54]. In this approach, only a limited number of trajectories that have high probabilities are traced. State estimation in hybrid systems has also been studied in [65, 19, 37].

Observer design and state observability in hybrid systems - Observer design and state observability are issues that are related to fault diagnosis. In

[13], observer design for *linear hybrid automata* is discussed. For the existence of the observer, it is required that the (affine) continuous system associated with each discrete state (location/mode) be observable. Therefore, observability reduces to identifying the current discrete state. The observer has two units: a *location-observer* which identifies the current discrete state (or a set of possible discrete states), and a continuous observer which gives the current continuous state in the current discrete state. In [14], a set of sufficient conditions is provided under which the hybrid system becomes observable. A hybrid system is called *observable* if for any initial continuous state, x_0, and initial discrete state, q_0, and any input, the continuous state and current discrete state can be identified asymptotically.

Observability in linear hybrid systems has also been discussed in [107]. Observability is studied for a class of hybrid systems called *jump-linear systems* with no inputs; the continuous system in each discrete state is *autonomous*. A set of necessary and sufficient conditions is provided under which the switching signal and the continuous state can be recovered uniquely from the output of the *jump* system. [28] extends the results of [107] to *affine hybrid systems* whose discrete transitions take place by enabling guard conditions on continuous states. In [87], building hybrid observers are discussed based on the proposed algorithms in [85, 78].

1.1.6 Applications of Diagnosis in Hybrid Systems

High-tech systems demand advanced systematic methods for health monitoring and diagnosis. In the following, we mention some real-world applications for diagnosis in hybrid systems.

In [79], NASA's Sprint AERCam is used as an application of diagnosis in hybrid systems. The AERCam is a small spherical robotic camera unit with 12 thrusters. Thrusters allow the camera to have linear and rotational motion. The system has been modeled as a hybrid system, and fault isolation has been studied using particle filtering approaches. Diagnosis of the propulsion system of an experimental rocket-powered vehicle called X-34 has been studied in [69]. X-34 is modeled by a 10^{th} order hybrid system with nonlinear dynamics containing both commanded and autonomous transitions. Fault diagnosis of a document processing factory modeled by a hybrid automaton has been studied in [115]. The authors use Xerox Document Center DC265 printer as the focus of their application. DC 265 is a multifunction system that can print 65 pages per minute. It is composed of a large number of moving components such as motors, solenoids, clutches, gears, rolls and belts. The system is modeled as a hybrid system with first-order linear dynamics in each mode. The diagnosis algorithm has been tested on a DC265 printer in the laboratory. Diagnosis in power systems modeled as hybrid systems is also studied in [40].

1.1.7 Fault Diagnosis in Gas Turbine Engine

Aircraft engines are complex systems that require high reliability to ensure flight safety and timely maintenance. There is a large body of research on health monitoring and fault diagnosis of aircraft engines (see e.g., surveys [74, 106, 58, 92] and the references therein). Fault diagnosis in gas turbine engines has been investigated using model-free data-driven methods as well as model-based approaches. Most of the model-based approaches for fault diagnosis in aircraft engines use the continuous dynamic models and rely on analytical redundancy (see e.g. [92, 63, 62, 82, 83, 30, 46]).

In [63] and [62], fault diagnosis in sensors and actuators of a gas turbine engine has been investigated using a *bank of Kalman filters.* In [82], a bank of Kalman filters was used for fault detection and isolation of failures in the sensors and actuators of the F-16 aircraft assuming that the faults are known. In [30], an observer-based method is developed for fault diagnosis of actuator and sensor faults in a gas turbine engine. In [63, 62, 82, 30], the authors do not try to identify the components (such as valves and pumps) responsible for the actuator faults. Fault detection and isolation of engine sensors has been studied in [83] using a bank of Kalman filters. A review of different model-based methods for fault diagnosis in engine sensors is reported in [92].

1.2 Thesis Objectives

In this work, we focus on the problem of fault detection and fault isolation in hybrid systems. The objective of this research is the development of a hybrid framework for fault diagnosis in hybrid systems modeled as hybrid automata. The motivation for this framework comes from the fact that in a complex hybrid system, there are discrete sensors that generate discrete outputs which are available at the DES representation of the system, and continuous sensors that generate continuous outputs which are present in the continuous model of the system. Discrete outputs can be used more efficiently for diagnosis of drastic faults such as valve stuck-closed, and continuous outputs can be used to diagnosis faults that slightly change the system dynamics such as the small loss-of-effectiveness in an actuator. Moreover, some faults may not be diagnosable if one uses purely DES abstracted models alone, and some faults may not be isolable if one uses only continuous sensor readings.

In this framework, we use a *bank of detection and isolation filters* (residual generators) for diagnosis of faults at the continous-variable level of the system. We develop a novel approach for systematically integrating the information coming from the low-level residual generators with the information available at the DES level to construct a **hybrid diagnoser**. The hybrid diagnoser can be used efficiently for diagnosing faults in the system.

Diagnosability of faults in our hybrid diagnosis framework is a challenging problem that we also investigate in this dissertation. The results obtained on diagnosability will be used to solve the problem of *residual generator selection* in hybrid systems. To study the applicability of our results in real-world applications, we model a gas turbine engine with hybrid system and apply our hybrid diagnosis framework to detect and isolate fault in the gas turbine engine.

1.3 Thesis Contributions

The thesis contributions are as follows.

Development of a hybrid framework for fault diagnosis of hybrid automata

In this thesis, we develop a hybrid framework for passive on-line fault diagnosis in systems modeled by hybrid automata. Generally, there are two types of sensors in the system: continuous sensors that generate a continuous output signal, and discrete sensors such as level sensors that are used for measuring different levels of a liquid in a tank. Diagnosis results based on a purely DES abstraction model will be conservative in the sense that a diagnosable *failure mode* may be rendered undiagnosable due to the abstraction and lack of sufficient information. On the other hand, fault diagnosis based on the purely continuous dynamics may not be possible at all times because some continuous variables may be unmeasurable due to the limitations on implementing continuous sensors. We develop a novel diagnosis approach in hybrid systems which systematically integrates the information at both DES and continuous levels for detecting and isolating faults. In our framework, all faults are modeled at the continuous level as unknown signals that are added to the system's dynamics similar to control input. Every fault signal corresponds to a fault or a set of faults in a component. Fault signals can model drastic failures such as stuck-open and stuck-closed faults of a valve and faults that slightly change the systems dynamics such as loss-of-effectiveness faults. We use a bank of residual generators that are designed based on the continuous-varibale dynamics of the system for detection and isolation of faults. We develop a systematic approach for modeling the residual generators by DES models and combining these DES models with the DES abstraction model of the hybrid system to construct an *extended DES* model. The extended DES model is used for fault diagnosis and diagnosability analysis. As opposed to approaches in [75, 50, 115], we do not lose a major amount of information that is available at the continuous-varible level due to the abstraction.

Using a bank of residual generators, an observer design methodology is investigated in [13]. In [13], a residual generator is designed for every discrete state. In our work, residual generators are not necessarily designed for all the discrete states of the system. We also develop a systematic method for merging the information at the DES level with the information generated by the residual generators.

Similar to our framework, the diagnosis approach of [86] performs diagnosis by using the information at both the DES and the continuous dynamics levels of the system. In our hybrid diagnosis framework, the information gathered from both the DES and the continuous-varibale dynamics levels is systematically integrated together. Therefore, our diagnosis approach is more efficient and general in the sense that it can be used in a convenient way for diagnoser design and diagnosability verification. Moreover, our framework provides a suitable tool to investigate, in a systematic way, the problem of residual generator selection for rendering faults diagnosable.

Investigation of the diagnosability of failures in hybrid automata

We develop a systematic approach for verifying the diagnosability of faults in our hybrid diagnosis framework. We introduce a notion of diagnosability in hybrid automata. Our diagnosability analysis can be viewed as a more general form of **δ-diagnosability** that is introduced in [35] and [105] where diagnosability in hybrid systems is studied using a timed automaton *abstraction* of the hybrid system. We show that if a fault is diagnosable in the extended DES model (developed for representing the hybrid automaton and residual generators), then it will be diagnosable for the hybrid system. The integrated information received from discrete and continuous sensors may contain redundant information which in turn can be used to detect and isolate faults in the presence of **false alarm** and **false silence** signals. We also investigate diagnosability of faults in the presence of false alarm and false silence signals.

Investigation of residual generator selection in hybrid automata

We investigate the problem of isolator (residual generator) selection and develop approaches for computing a *minimal set* of isolators to maintain the diagnosability of a failure mode. We develop necessary and sufficient conditions for isolator selection in a hybrid automaton so that a fault becomes diagnosable in the extended DES model of the hybrid system and isolators. The problem of isolator selection studied for hybrid automata can be considered as the counterpart of the sensor selection problem that is discussed in [90, 32, 5] for purely DES.

Application of the hybrid diagnosis framework to gas turbine engines

We employ our hybrid diagnosis approach for investigating faults in the fuel supply system and the nozzle actuator of a single-spool turbojet engine with an afterburner. First, we develop a hybrid automaton model for the gas turbine engine. We model components in the fuel supply system and the nozzle actuator using DES models. The status of these components varies when the operating regime of the engine changes. A DES abstract model can be developed for the engine by parallel composition of all the component models. We also develop a nonlinear system model for describing the continuous dynamics of the engine. We develop simpler linear system models for representing the continuous dynamics of the engine in different operating regimes by linearization of the nonlinear model about different operating points. A hybrid automaton model for the engine is constructed by combining the DES models of the fuel supply system and the nozzle actuator with the linear systems representing the engine dynamics in different operating regimes. A bank of residual generators is designed based on the linear system models. Each residual generator is modeled by a DES and an extended DES is developed by combining the DES models of the residual generators and the DES model of the engine. Based on the extended DES model, a hybrid diagnoser is constructed. We show that there are cases where the faults in the fuel supply system and the nozzle actuator cannot be isolated by a purely DES diagnoser or by methods that are based on residual generators alone. However, the faults can be isolated if the hybrid diagnoser is used. A number of simulation studies are conducted to demonstrate and verify the advantages of our proposed hybrid fault diagnosis approach.

Due to the criticality and complexity of gas turbine engines, employing more descriptive and efficient modeling tools and algorithms has always been the subject of research in control and fault diagnosis of gas turbine engines. To the best of our knowledge, the hybrid fault diagnosis in gas turbine engines presented in this thesis is the first work which models a gas turbine engine with hybrid automata models and investigates fault diagnosis in the gas turbine engine using hybrid diagnosis approaches.

1.4 Thesis Outline

In Chapter 2, we briefly review the background material which will be used in the following chapters. We review modeling of systems with DES and in particular Finite-State Automata (FSA). We also review fault diagnosis and diagnosability analysis in DES and fault detection and isolation in continuous linear systems. In Chapter 3, we describe our hybrid diagnosis framework. We study diagnosability of faults in Chapter 4. In Chapter 4, we assume that all the residual generators that can be designed based on the continuous dynamics of different discrete states are constructed and used for diagnosability analysis. In Chapter 5, we investigate the problem of isolator (residual generator) selection and develop approaches for computing a *minimal set* of isolators to maintain the diagnosability of a failure mode. In Chapter 6, we employ our hybrid diagnosis approach for investigating faults in the fuel supply system and the nozzle actuator of a single-spool turbojet engine with an afterburner. Finally, in Chapter 7, we present a summary of our results and discuss directions for future research.

Chapter 2

BACKGROUND

In this chapter, we present an overview of the background material related to our work. In this work, we study fault diagnosis in systems modeled by hybrid automata. In this chapter, we first explain Discrete-Event Systems (DES) and in particular Finite-State Automata (FSA) as a tool for modeling discrete-event systems. FSA are used in our work for modeling the DES level of hybrid systems. Next, we review diagnosis in FSA and provide the necessary and sufficient conditions for the diagnosability of faults in FSA. The diagnoser design method described here will be later used in Chapter 3 for designing a diagnoser for a DES representing a hybrid system and its residual generators. We also briefly explain model-based fault diagnosis in continuous systems and present the necessary and sufficient conditions for the existence of residual generators in linear systems.

2.1 Discrete-Event Systems

A **Discrete-Event System (DES)** is a dynamical system equipped with a discrete state set and an event driven state transition structure. An event in a DES occurs instantaneously causing transition from one state to another. Automata theory [55] provides one of the most comprehensive sets of mathematical tools for studying

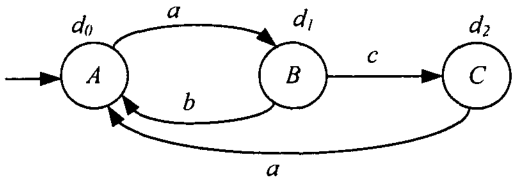


Figure 2.1: A simple FSA with three states.

DES. Many of the other models (such as Petri nets) for describing DES are rooted in the automata theory. In automata models, the system evolution is represented by transitions from one state to another. The reader is referred to [96, 97, 112] for details. In this dissertation, we use **Moore finite-state automata** to model DES. First we review some basic definitions and operations on automata.

2.1.1 Languages and Finite-state Automata

An **alphabet** Σ is a finite set of **symbols**. Symbols correspond to events in DES models. A symbol **sequence** over Σ has the form $\sigma_1\sigma_2\cdots\sigma_n$ for $n \geq 1$, where $\sigma_i \in \Sigma$ with $1 \leq i \leq n$. The set Σ^+ denotes the collection of all possible finite symbol sequences over Σ. The set $\Sigma^* = \{\epsilon\} \bigcup \Sigma^+$ represents the set of all strings sequences over Σ. Here, ϵ denotes the empty sequence (sequence with no symbols).

Definition 2.1.1. *A* ***language*** *over alphabet Σ is any subset of Σ^*.* ■

The empty language is shown by $\emptyset$.

A **finite-state Moore automaton (generator)** G is a 6-tuple:

$$G = (Q, \Sigma, T, D, \lambda, q_0)$$

where Q is the non-empty state set; q_0 is the initial state; $T : Q \times \Sigma \times Q$ is the set transitions; D is the set of discrete outputs and $\lambda : Q \longrightarrow D$ is the output map.

Example 2.1.1. *Fig. 2.1 shows a simple FSA consisting of three states. In a Moore FSA, the output is associated with state. Here, $Q = \{A, B, C\}$, $\Sigma = \{a, b, c\}$,*

$q_0 = A$, $T = \{(A,a,B),(B,b,A),(B,c,C),\ (C,a,A)\}$, $D = \{d_0,d_1,d_2\}$, $\lambda(A) = d_0$, $\lambda(B) = d_1$ *and* $\lambda(C) = d_2$. ■

For any $q_i \in Q$, $\sigma_i \in \Sigma$ with $i \in \{1,\cdots,n\}$, and $n \geq 2$, a path $x_1 \xrightarrow{\sigma_1} x_2 \xrightarrow{\sigma_2} \cdots \xrightarrow{\sigma_{n-1}} x_n$, is called a **cycle** if $x_1 = x_n$. In this thesis, sometimes, the states of the path of a cycle are also referred to as a cycle.

Definition 2.1.2. *[48] Suppose two states q and q' of Q satisfy $\lambda(q) \neq \lambda(q')$, and q' can be reached from q through a path along which the output is equal to $\lambda(q)$ (except at q'), then we say q' is **output-adjacent** to q and write $q \Rightarrow q'$.* ■

Let $G_1 = (Q_1,\Sigma_1,T_1,D_1,\lambda_1,q_{0,1})$ and $G_2 = (Q_2,\Sigma_2,T_2,D_2,\lambda_2,q_{0,2})$ be two FSA. Consider a **product** of G_1 and G_2 shown as $G_1 \otimes G_2$ in which the shared events of two FSA are synchronized. Specifically,

$$G_1 \otimes G_2 = (Q,\Sigma,T,D,\lambda,q_0)$$

where

$$\begin{aligned}
&Q = Q_1 \times Q_2\\
&\Sigma = \Sigma_1 \bigcup \Sigma_2\\
&q_0 = (q_{0,1},q_{0,2})\\
&D = D_1 \times D_2\\
&T = \{((q_1,q_2),\sigma,(q_1',q_2')) \mid \sigma \in \Sigma_1 \bigcap \Sigma_2 \text{ and } (q_1,\sigma,q_1') \in T_1 \text{ and } (q_2,\sigma,q_2') \in T_2\}\\
&\quad \bigcup \{((q_1,q_2),\sigma,(q_1',q_2)) \mid \sigma \in \Sigma_1 - \Sigma_2 \text{ and } (q_1,\sigma,q_1') \in T_1\}\\
&\quad \bigcup \{((q_1,q_2),\sigma,(q_1,q_2')) \mid \sigma \in \Sigma_2 - \Sigma_1 \text{ and } (q_2,\sigma,q_2') \in T_2\}
\end{aligned}$$

Function $\lambda : Q \longrightarrow D$ is the output map such that $\lambda((q_1,q_2)) = (\lambda(q_1),\lambda(q_2))$.

The **synchronous product** or **parallel composition** of G_1 and G_2, shown as **sync**(G_1,G_2), is defined to be the *reachable* sub-generator of $G_1 \otimes G_2$. Operator

sync models the joint operation of automata. Note that **sync** is a commutative and associative operation, namely:

Commutative Property:

$$\textbf{sync}(G_1, G_2) = \textbf{sync}(G_2, G_1)$$

Associative property:

$$\textbf{sync}(G_1, \textbf{sync}(G_2, G_3)) = \textbf{sync}(\textbf{sync}(G_1, G_2), G_3)$$

Consider three FSA G_1, G_2 and G_3. The synchronous product of G_1, G_2 and G_3, $\textbf{sync}(G_1, G_2, G_3)$, is defined as:

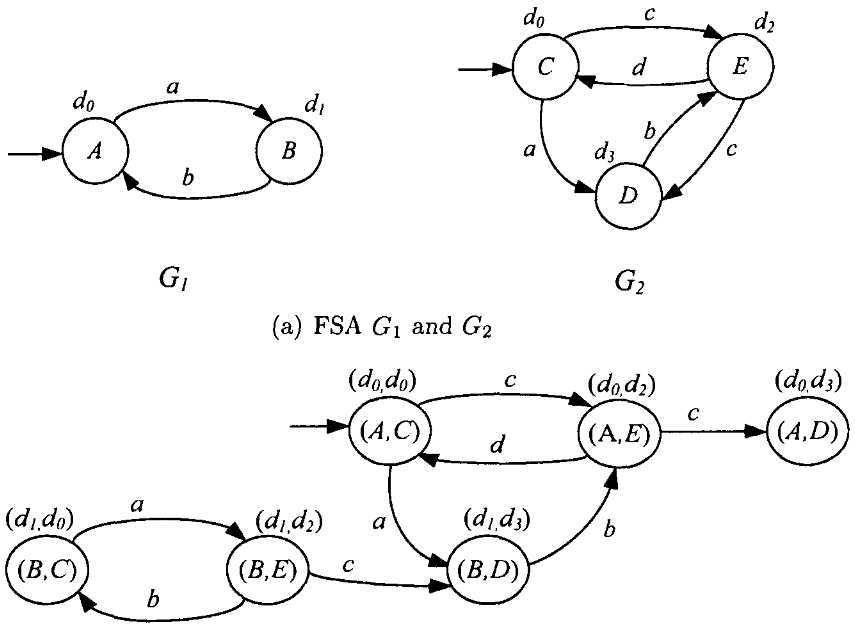
$$\textbf{sync}(G_1, G_2, G_3) = \textbf{sync}(G_1, \textbf{sync}(G_2, G_3)) = \textbf{sync}(\textbf{sync}(G_1, G_2), G_3)$$

The synchronous product of more than three FSA can be defined similarly.

Example 2.1.2. *Two FSA, G_1 and G_2 are shown in Figure 2.2(a) The synchronous product of G_1 and G_2 is depicted in Figure 2.2(b).* ■

Another operation on FSA is **meet**. The result of the meet operation of G_1 and G_2, shown as $\textbf{meet}(G_1, G_2)$, is a reachable generator in which only the common events may occur and in synchrony. In the case that $\Sigma_1 = \Sigma_2$, $\textbf{meet}(G_1, G_2) = \textbf{sync}(G_1, G_2)$.

The hybrid diagnoser that we develop for a hybrid system is constructed based on a DES model representing the hybrid system and the residual generators. In the next section, we briefly review diagnosis in FSA.



(a) FSA G_1 and G_2

(b) **sync**(G_1, G_2)

Figure 2.2: Example 2.1.2- a) Two FSA G_1 and G_2, b) Synchronous product of G_1 and G_2.

2.2 Diagnosis in FSA

We use the state-based method of [48] for diagnoser design in DES which is briefly reviewed below.

Consider a non-deterministic Moore finite-state automaton $G = (Q, \Sigma, T, D, \lambda, q_0)$. It is assumed that $\Sigma = \Sigma_o \bigcup \Sigma_{uo}$, where Σ_o represents the observable event set and Σ_{uo} consists of unobservable events.

Suppose there are m **failure modes** $F^1, F^2, \cdots, F^m$ in G. Each failure mode corresponds to a failure (or a set of failures) in the system. A valve stuck-closed or a motor stuck-off are examples of failure modes. Let

$$\mathcal{K} = \{N, F^1, \cdots, F^m, F^{1,2}, \cdots, F^{m-1,m}, \cdots, F^{1,\cdots,m}\}$$

denote the **condition set**. The system can be in the normal condition (N) or a

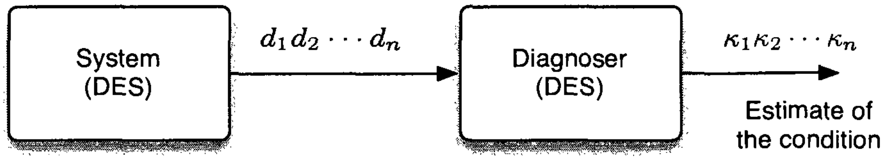


Figure 2.3: Diagram of a DES and diagnoser.

condition corresponding to a combination of failure modes. Thus, for example, for $p = 2$, the condition set will be $\mathcal{K} := \{N, F^1, F^2, F^{1,2}\}$, where $F^{1,2}$ corresponds to the case where both failure modes F^1 and F^2 have occurred.

The event set can be partitioned as $\Sigma = \Sigma_f \dot{\bigcup} \Sigma_N$, where $\Sigma_f = \{\hat{f}^1, \cdots, \hat{f}^m\}$ is the set of **fault events** and Σ_N is the set of non-fault events. Each fault event corresponds to a failure mode. We assume, without loss of generality, that all fault events are unobservable, i.e., $\Sigma_f \subseteq \Sigma_{uo}$. That is, the occurrence of a failure mode does not result in an output change that identifies the failure mode. The system enters faulty conditions as the consequence of the occurrence of fault events. We assume that the state set can be partitioned according to the condition of the system:

$$Q = Q_N \dot{\bigcup} (Q_{F^1} \dot{\bigcup} \cdots \dot{\bigcup} Q_{F^m}) \dot{\bigcup} (Q_{F^{1,2}} \dot{\bigcup} \cdots \dot{\bigcup} Q_{F^{m-1,m}}) \dot{\bigcup} \cdots \dot{\bigcup} Q_{F^{1,\cdots,m}}$$

The **condition map** $\kappa : Q \to \mathcal{K}$ is defined such that for every $q \in Q$, $\kappa(q)$ is the condition of the system at the state q. The definition of κ is extended to the subsets of Q: $\kappa(z) = \{\kappa(q) | q \in z\}$, for any $z \subseteq Q$.

As discussed in the previous chapter, Sampath et al. developed the concept of diagnoser to perform diagnosis [98]. In the event-based framework of [98], the diagnoser can be viewed as an extended observer for the system which gives an estimate of the current state of the system and information on potential past occurrences of failure events. In the state-based framework [48], however, the diagnoser is a dynamical system that generates an estimate for the condition of the system by using the output sequence $(d_1 d_2 \cdots d_n)$ generated by the system. This is done by calculating a set $z_n \subseteq Q$ to which q must belong at the time that d_n was generated (see Figure 2.3); $\kappa(z_n)$ will be the estimate of the system condition. After the generation of the next output (d_{n+1}), the diagnoser updates z_n.

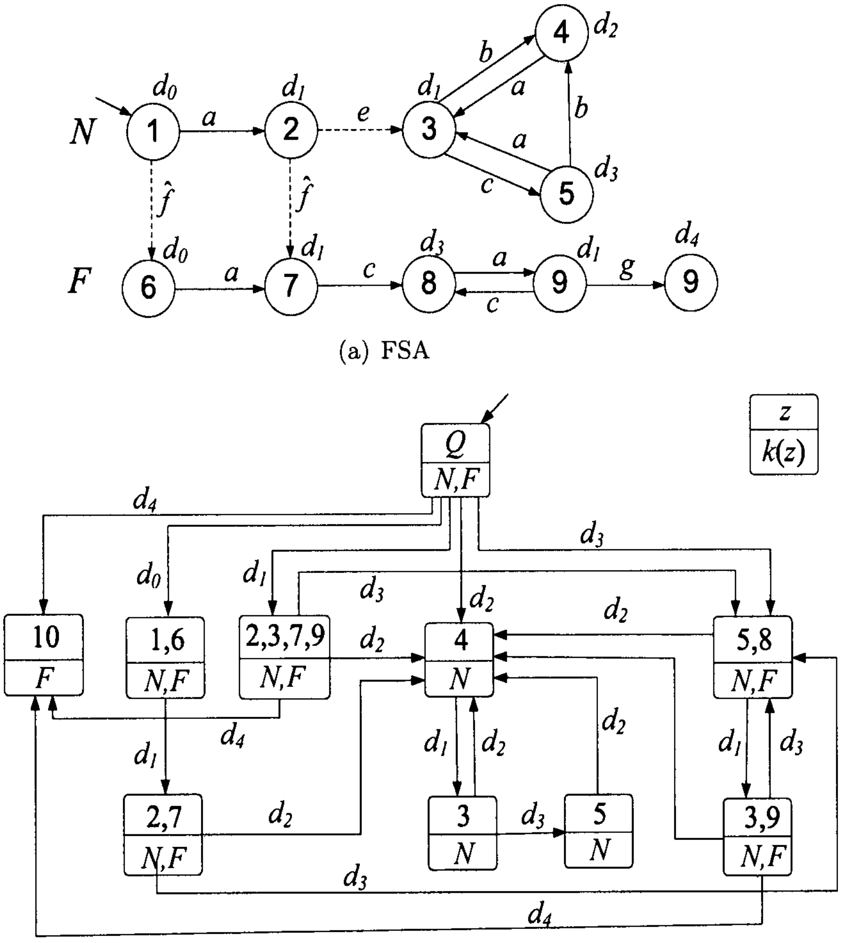
In [48], the **diagnoser** designed for G is defined to be a finite-state Moore automaton $D_G = (Z \bigcup \{\underline{z}_0\}, D, \xi, z_0, \hat{\mathcal{K}}, \kappa)$, where $Z \bigcup \{\underline{z}_0\}$, D and $\hat{\mathcal{K}} \subseteq 2^{\mathcal{K}} - \{\emptyset\}$ are the state, event and output sets of D; $\underline{z}_0 := (z_0, 0)$ is the initial set with $z_0 \in 2^Q - \{\emptyset\}$; $Z \subseteq 2^Q - \{\emptyset\}$, and $\xi : Z \bigcup \{\underline{z}_0\} \times D \to Z$ represents the transition function; $\kappa : Z \bigcup \{\underline{z}_0\} \to \hat{\mathcal{K}}$ denotes the output map. Given the state estimate z_n and upon observing d_{n+1}, the state estimate is updated according to

$$z_1 = z_0 \bigcap \lambda^{-1}(\{d_1\}) \ (n = 0)$$

$$z_{n+1} = \xi(z_n, d_{n+1}) = \{q \mid \lambda(q) = d_{n+1} \ \& \ (\exists q' \in z_n : q' \Rightarrow q)\} \ (n \geq 1)$$

The state estimate z_0 holds the information available about the state of the system at the time that the diagnoser is started.

Example 2.2.1. *Consider the FSA in Figure 2.4(a) with the set of fault events $\Sigma_f = \{\hat{f}\}$ and the failure mode F. The event set is $\Sigma = \{a, b, c, e, f, g\}$ with $\Sigma_o = \{a, b, c, g\}$ and $\Sigma_{uo} = \{e, f\}$. The unobservable events are shown by dashed lines in Figure 2.4(a). The condition set is $\mathcal{K} = \{N, F\}$. Here, for example, we have $1 \Rightarrow 7$, $2 \Rightarrow 8$ and $2 \Rightarrow 4$. Figure 2.4(b) shows the diagnoser designed for the system. Initially, the state of the system is assumed unknown. Therefore, $z_0 = Q$, and the condition of the system is uncertain, i.e., $\{N, F\}$. Suppose after the initialization of the diagnoser, output "d_1" is observed. Thus, the state of the system must be "2", "3", "7" or "9", and the condition of the system is $\{N, F\}$. Similarly, if "d_3" is observed first, then the state of the system must be "5" or "8" and the condition of the system is $\{N, F\}$. Now, suppose output "d_2" is observed after "d_3". The state*



(a) FSA

(b) Diagnoser

Figure 2.4: Example 2.2.1- A finite-state automaton and its diagnoser.

of the system must be "4" and the condition of the system is $\{N\}$. This means that the fault has not occurred. ■

This diagnoser has a cycle $\{5,8\} \rightarrow \{3,9\} \rightarrow \{5,8\}$ corresponding to the output sequence $d_3d_1d_3$, which is called an **"F-indeterminate"** cycle because the condition estimate in the diagnoser cycle is $\{N, F\}$ and hence **uncertain**. It should be noted that there are two cycles in the system in Figure 2.4(a) with the same output sequence $d_3d_1d_3$, namely: the states $5-3-5$ in the normal mode (N) and the states $8-9-8$ in the failure mode F. The diagnoser cannot distinguish between these two cycles. Therefore, if the system is trapped in the faulty cycle $8-9-8$, the diagnoser will not be able to detect the failure. Fault diagnosability is reviewed in Section 2.3.

The fault diagnosis method developed in [48] relies on the output sequence and output-adjacent states. Therefore, it is useful to store the information about the output-adjacent states of G in a **Reachability Transition System (RTS)** [48]. The RTS (corresponding to G) is defined to be the transition system $\overline{G} = (Q, R, D, \lambda)$ which has Q, D and λ as the state set, output set and output map, respectively; $R \subseteq Q \times Q$ is a binary relation, and $(q_1, q_2) \in R$ if and only if $q_1 \Rightarrow q_2$.

The number of states of the diagnoser in the worst case is exponential in the number of system states $|Q|$. Therefore, for fault diagnosis, instead of constructing the diagnoser D_G, it is computationally more practical to compute the RTS of G off-line and use it later for **online implementation** of the diagnosis algorithm [48]. In other words, having z_n and the observation d_{n+1}, use $\overline{G}$ to compute z_{n+1} and $k(z_{n+1})$.

The RTS (in the form of a table) corresponding to the system in Figure 2.4(a) is shown in Table 2.1. From a computational point of view, RTS can be computed in $\mathcal{O}(|Q|^2 + |Q||T|)$ time because a breadth-first search reachability analysis for each $q \in Q$ can be done in $\mathcal{O}(|Q| + |T|)$ [48].

State	Output	Output-adjacent States
1	d_1	2,7
2	d_2	4
2	d_3	5,8
3	d_2	4
3	d_3	5
4	d_1	3
5	d_2	4
5	d_3	3
6	d_1	7
7	d_3	8
8	d_1	9
9	d_3	8
9	d_4	10

Table 2.1: Reachability transition system of the FSA in Figure 2.4(a).

The DES diagnoser discussed in [48] only uses the discrete outputs of the system for diagnosis. In general some of the events are observable. If the occurrence of these observable events cannot be inferred from the output sequence, then the information about the occurrence of the observable events can be included in the output map [48]. In the following, we discuss diagnosability of faults in the framework of [48].

2.3 Diagnosability of Faults in DES

The example in Section 2.2 illustrates the importance of **diagnosability** in DES.

Let $\mathcal{F} = \mathcal{K} - \{N\}$ denote the set of **faulty conditions**. Also let $\mathcal{F}^i$ be the set of faulty conditions in which the failure mode F^i is present, and let $\overline{\mathcal{F}^i} = \mathcal{F} - \mathcal{F}^i$. For example, consider a system with three failure modes F^1, F^2 and F^3 and assume that $\mathcal{K} = \{N, F^1, F^2, F^3, F^{1,2}, F^{1,3}, F^{2,3}, F^{1,2,3}\}$. We have $\mathcal{F} = \{F^1, F^2, F^3, F^{1,2}, F^{1,3}, F^{2,3}, F^{1,2,3}\}$ and $\mathcal{F}^1 = \{F^1, F^{1,2}, F^{1,3}, F^{1,2,3}\}$.

Let $Q_{\mathcal{F}^i}$ be the set of all discrete states corresponding to the condition set $\mathcal{F}^i$. Similarly, let $Q_{\overline{\mathcal{F}^i}}$ be the set of all discrete states corresponding to the condition set

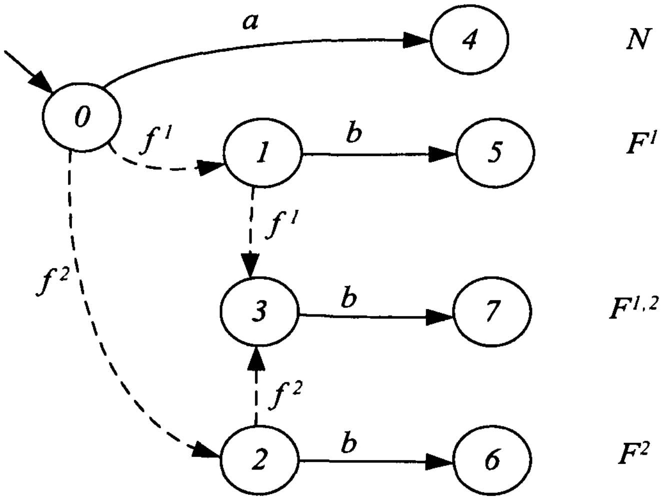


Figure 2.5: A FSA with two failure modes.

$\mathcal{F} - \mathcal{F}^i$. For example, in the FSM shown in Figure 2.5, we have $Q_{\mathcal{F}^1} = \{1,3,5,7\}$, $Q_{\mathcal{F}^2} = \{2,3,6,7\}$ and $Q_{\overline{\mathcal{F}^1}} = \{0,2,4,6\}$. Define $G_{\mathcal{F}^i}$ as the sub-generator of G consisting of the states of $Q_{\mathcal{F}^i}$ only. Similarly, define G_N, $G_{\mathcal{F}}$ and $G_{N,\overline{\mathcal{F}^i}}$ as the sub-generators of G consisting only the states of Q_N, $Q_{\mathcal{F}}$ and $Q_N \bigcup Q_{\overline{\mathcal{F}^i}}$, respectively.

In the following, we provide a set of necessary and sufficient conditions for failure diagnosability in FSA. First, we bring in some related definitions.

Definition 2.3.1. *[48] If the occurrence of a failure mode F^i can be directly concluded from the generation of an output symbol 'd' $\in D$, then 'd' is called* **Fⁱ-*indicative*.** ■

For example, in the FSA shown in Figure 2.4(a), discrete output "d_4" is F-indicative.

A state z of the diagnoser corresponding to a state estimate for the system is called **$\mathbf{F^i}$-certain** if $\kappa(z)$, the corresponding estimate of the system's condition, indicates that the failure has occurred [48]. A state z of the diagnoser is defined as **$\mathbf{F^i}$-uncertain** if $\kappa(z)$, the corresponding estimate of the system's condition is consistent with the occurrence of F^i but doesn't conclusively indicate that the failure has occurred. For example, in the DES diagnoser shown in Figure 2.4(b), the states with the condition estimate $\{N, F\}$ are F-uncertain, and the state with the condition estimate $\{F\}$ is F-certain.

The **output language** $L_o(G, q)$ generated by G from the state $q \in Q$ is defined as

$$\begin{aligned} L_o(G,q) :=& \{d_1 d_2 \cdots d_m \in D^+ \text{ such that } d_1 = \lambda(q) \text{ and} \\ & [\exists q_i \in Q \ (1 \le i \le m): \ q_1 = q, q_{i-1} \Rightarrow q_i, d_i = \lambda(q_i), (2 \le i \le m)]\} \end{aligned}$$

The output languages $L_o(G_{\mathcal{F}^i}, q)$, $L_o(G_{N,\overline{\mathcal{F}^i}}, q)$, $L_o(G_N, q)$ and $L_o(G_{\mathcal{F}^i}, q)$ are defined similarly.

The diagnosability of a failure mode F^i is defined as follows [48].

Definition 2.3.2. *[48] A permanent failure mode F^i of G is said to be* ***diagnosable*** *if F^i can be detected and isolated in a bounded number of events in G following both the occurrence of F^i and the initialization of diagnosis.* ■

Theorem 2.3.1. *[90] Assume that $z_0 = Q$. A permanent failure mode F^i is diagnosable in G if and only if:*

1. *For any $q \in Q_{\mathcal{F}^i}$, if there is no transition out of q, then $\lambda^{-1}(\lambda(q)) \bigcap (Q - Q_{\mathcal{F}^i}) = \emptyset$;*

2. *If there is a cycle in $Q_{\mathcal{F}^i}$ consisting of discrete states having the same output, say d, then $\lambda^{-1}(d) \bigcap (Q - Q_{\mathcal{F}^i}) = \emptyset$;*

3. *For any $q \in Q_{\mathcal{F}^i}$ and $q' \in (Q_N \bigcup Q_{\overline{\mathcal{F}^i}})$ satisfying $\lambda(q) = \lambda(q')$, we have:*

$$\{s | s \in L_o(G_{N,\overline{\mathcal{F}^i}}, q') \bigcap L_o(G_{\mathcal{F}^i}, q), |s| \ge |Q|^2\} = \emptyset$$

■

Condition (1) in Theorem 2.3.1 states that there should be no deadlock state in $Q_{\mathcal{F}^i}$ with no transition out of the state unless the output in that state can be generated only when F^i has occurred. In other words, such an output has to be F-indicative. Similarly, condition (2) states that there should be no cycles with constant output in $Q_{\mathcal{F}^i}$ unless the constant output is F-indicative. Finally, condition (3) states that there should be no cycle in $Q_{\mathcal{F}^i}$ having a sequence of outputs that can also be generated by a cycle in $Q_N \bigcup Q_{\overline{\mathcal{F}^i}}$ (otherwise, F^i cannot be distinguished and hence will be undiagnosable).

In [90], it is shown that testing diagnosability for nondeterministic FSA has complexity $\mathcal{O}(|Q|^4)$. In the next section, we briefly describe fault diagnosis in continuous systems and present the conditions for the existence of residual generators for linear systems.

2.4 Model-based Fault Diagnosis in Continuous Systems

As explained in the previous chapter, model-based approaches for fault diagnosis in continuous systems rely on **analytical redundancy**. In these approaches, there are residual generators which generate residual signals from the difference between actual measurements and their estimates obtained using the system's model. Faults are detected by setting fixed or variable thresholds on residual signals. A number of residual generators can be designed with each being sensitive to only one of the faults. Fault isolation is achieved by analyzing each residual once the threshold is exceeded.

A general diagram of the model-based fault diagnosis in continuous system is shown in Figure 2.6. It has two stages, namely: residual generation and residual evaluation [101].

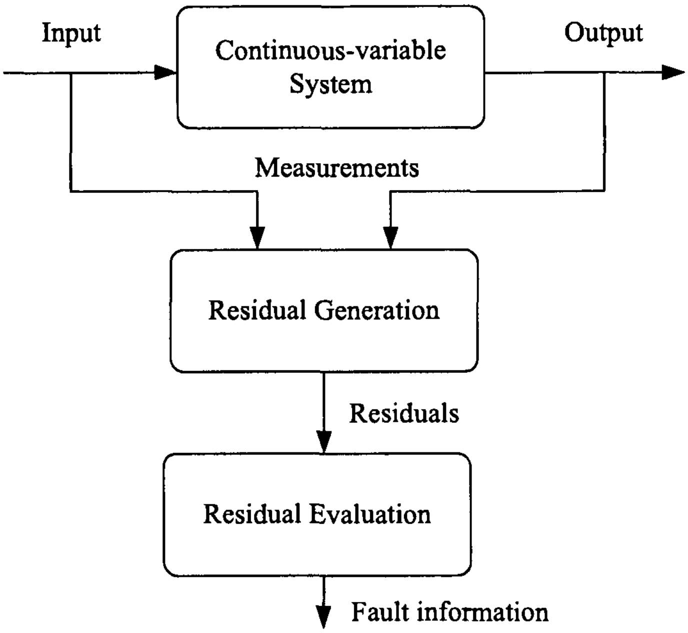


Figure 2.6: Diagram of fault diagnosis in continuous systems adopted from [101].

Residual generation: This block generates residual signals by using the available inputs and measured outputs of the system. The residual must contain information regarding the occurrence of a fault. Normally, it should be zero or close to zero if no fault is present in the system. In contrast, it has to be distinguishably different from zero when a fault occurs. The residual generation can be designed based on any method that is described in the previous chapter.

Residual evaluation: This block consists of a decision rule and examines the residuals to determine if any fault has occurred. It may perform a simple threshold test on the instantaneous values or moving averages of the residuals. It may also use statistical methods, for example, generalized likelihood ratio testing or sequential probability ratio testing to detect and isolate faults.

Residual generators used in our work can be designed using any method that is mentioned in the previous chapter. However, without loss of generality, we assume a simple threshold passing test for the evaluation of residual signals. In the next

Chapter, we will describe how the information contained in the value of the residuals can be incorporated in our framework for isolating faults.

Diagnosability of a fault in a continuous system is related to the existence of residual generators for the system to detect and isolate the fault. Using geometrical approaches, the existence of residual generators has been studied in [80, 81] for linear systems and in [33] for nonlinear systems. In our work, the continous-variable dynamics of the system can be linear or nonlinear. However, in the following, we present the well-known **Fundamental Problem in Residual Generation (FPRG)** as discussed in [80, 81] for Linear Time-Invariant (LTI) systems. The solvability conditions for the existence of residual generators provide the solutions to the FPRG.

2.4.1 Fundamental Problem of Residual Generation (FPRG)

Let $\mathcal{X}$ and $\mathcal{Y}$ be linear spaces over the field of real numbers R. Let $C : \mathcal{X} \longrightarrow \mathcal{Y}$ denote a *linear transformation* (or *map*) from $\mathcal{X}$ to $\mathcal{Y}$. The vector space $\mathcal{X}$ is called the *domain* of C, and $\mathcal{Y}$ is the *codomain*. The *kernel* (or *null-space*) of C is the subspace

$$Ker\ C := \{x \mid x \in \mathcal{X} \text{ and } Cx = 0\} \subseteq \mathcal{X}$$

The *image* of C is the subspace

$$Im\ C := \{y \mid y \in \mathcal{Y} \text{ and } \exists x \in \mathcal{X} :\ y = Cx\} \subseteq \mathcal{Y}$$

Consider an LTI system

$$S : \begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) \end{cases} \tag{2.1}$$

where $A \in \mathbb{R}^{nxn}$, $B \in \mathbb{R}^{nxp}$ and $C \in \mathbb{R}^{rxn}$ are the system matrices. The system S sometimes is represented by the triple (C, A, B). In the following, $< Ker\ C|A >$ denotes the unobservable subspace of the pair (C, A) and is defined as

$$< Ker\ C|A >= Ker\ C \bigcap A^{-1} Ker\ C \bigcap \cdots \bigcap A^{-n+1} Ker\ C$$

Definition 2.4.1. *[113] A subspace $S \subseteq \mathbb{R}^n$ is a* ***(C,A)-unobservability subspace*** *if $S =< Ker\ HC|A + DC >$ for some maps $D : \mathbb{R}^l \longrightarrow \mathbb{R}^n$ and $H : \mathbb{R}^l \longrightarrow \mathbb{R}^l$.* ■

Definition 2.4.2. *[113] The system (C, A, B) is called* ***input-observable*** *if B is monic and $< Ker\ C|A > \bigcap Im\ B = 0$.* ■

Now, consider the LTI system

$$S_f^m : \begin{cases} \dot{x}(t) = Ax(t) + Bu(t) + \sum\limits_{i=1}^{m} L_i f_i(t) \\ y(t) = Cx(t) \end{cases} \tag{2.2}$$

where $u(t)$ and $y(t)$ are the input and output of the system, respectively, and are assumed to be known. They are referred to as the *observables.* Functions $f_i(t) \in \mathbb{R}^{n \times s_i}$, for $i \in \{1, \cdots, m\}$, are arbitrary and unknown functions of time and called **fault type signals**. For simplicity, these functions in our hybrid diagnosis framework discussed in Chapter 3 are called **fault types**. Note that the fault functions are not necessarily scalars. The matrices L^i are called **fault signatures**. Without loss of generality, assume that the system has two faults and is in the form:

$$S_f^2 : \begin{cases} \dot{x}(t) = Ax(t) + Bu(t) + L_1 f_1(t) + L_2 f_2(t) \\ y(t) = Cx(t) \end{cases} \tag{2.3}$$

Given the system S_f^2 in Equation 2.3, we want to design an LTI residual generator which is sensitive to $f_1(t)$ but insensitive to $f_2(t)$. This problem is called the FPRG [80, 81].

Consider the residual generator RG of the form

$$RG : \begin{cases} \dot{w}(t) = Fw(t) - Ey(t) + Gu(t) \\ r(t) = Mw(t) - Hy(t) + Ku(t) \end{cases} \tag{2.4}$$

where F, E, G, M, H, and K are the matrices with appropriate dimensions. Residual generator RG takes observables $u(t)$ and $y(t)$ as inputs, and generates residual $r(t)$, with the following properties.

- $r(t) \longrightarrow 0$ if $f_1(t) = 0$, i.e., transfer matrices $u \longrightarrow r$ and $f_2 \longrightarrow r$ are zero, and modes observable from $r(t)$ are asymptotically stable.
- The transfer matrix $f_1 \longrightarrow r$ is input-observable.

Theorem 2.4.1. *[81] The FPRG has a solution if and only if*

$$\mathbb{S}^* \bigcap Im\ L_1 = 0 \tag{2.5}$$

where $\mathbb{S}^*$ is the infimal (C, A)-unobservability subspace containing $Im\ L_2$. ■

The FPRG can be extended to the case of multiple faults as follows [81]. Assume that k faults are present in the system. The objective is to design a filter which generates k residuals, $r_i(t)$ $(i \in \{1, \cdots, k\})$, such that the failure of the i-th component, i.e, nonzero $f_i(t)$, can only affect the i-th residual $r_i(t)$ and no other residuals $r_j(t)$ $(j \neq i)$. This problem is called the Extended Fundamental Problem in Residual Generation (EFPRG).

Theorem 2.4.2. *[81] The EFPRG has a solution if and only if*

$$\mathbb{S}_i^* \bigcap Im\ L_i = 0 \tag{2.6}$$

where $\mathbb{S}_i^*$ is the infimal (C, A)-unobservability subspace containing $\sum_{j \neq i} Im\ L_j$. ■

The geometric relations in the above theorems provide a general solution to the problem of residual generation in LTI systems. In [81] and [80], a procedure is described for calculating the matrices F, E, G, M, H, and K of the residual generator for the case that FPRG and EFPRG have solutions.

In the following, we extend the results of [81] to study the existence of residual generators which are sensitive to a set of fault types and insensitive to the rest. Consider the LTI system in Equation 2.2 with m fault types. Let $FT = \{f^1, \cdots, f^m\}$ be the set of fault types, and let $\Phi = \{f^{k_1}, \cdots, f^{k_j}\}$ be a subset of FT with $\Phi \neq \emptyset$. We want to design an LTI residual generator which is sensitive to all $f^i(t) \in \Phi$ but insensitive to all $f^j(t) \in \overline{\Phi}$, where $\overline{\Phi} = FT - \Phi$. We call this problem the **Generalized Extended Fundamental Problem in Residual Generation (GEFPRG)**. Let RG in the form of Equation 2.4 be the residual generator that is designed. Residual generator RG takes observable signals $u(t)$ and $y(t)$ as inputs, and generates residual $r(t)$, with the following properties:

1. The transfer matrices $u \longrightarrow r$ and $f^i \longrightarrow r$ are zero for all $f^i(t) \in \overline{\Phi}$, and the modes observable from $r(t)$ are asymptotically stable. Therefore, $r(t) \longrightarrow 0$ if $f^j(t) \equiv 0$, for all $f^j(t) \in \Phi$ and
2. The transfer matrix $[f^{k_1}, \cdots, f^{k_j}] \longrightarrow r$ is input-observable.

Corollary 2.4.3. *(Corollary of the EFPRG Theorem in [81]) The GEFPRG has a solution if and only if $\mathbb{S}^* \bigcap \sum_{f^i \in \Phi} Im\ L^i = 0$, where $\mathbb{S}^*$ is the infimal (C, A)-unobservability subspace containing $\sum_{f^j \in \overline{\Phi}} Im\ L^j$, and L^i is the fault signature of f^i.* ■

If $\Phi = FT$, the fault isolation problem becomes equivalent to the fault detection problem. In this case, the solvability condition is that the transfer matrix $[f^{k_1}, \cdots, f^{k_j}] \longrightarrow y$ is input-observable.

In the next chapter, we describe our hybrid diagnosis framework.

Chapter 3

FAULT DIAGNOSIS IN HYBRID SYSTEMS

In this chapter, we develop a hybrid framework for fault diagnosis in hybrid systems. As described in Chapter 1, normally there are two types of sensors in the system: continuous sensors that generate a continuous output signal, and discrete sensors such as level sensors that are used for measuring different levels of a liquid in a tank. Diagnosis results based on purely DES abstract model and the outputs of discrete sensors will be conservative in the sense that a diagnosable failure mode may be rendered undiagnosable due to the abstraction and lack of sufficient information. On the other hand, fault diagnosis based on purely continuous dynamics and the outputs of continuous sensors may not be possible at all times. An efficient diagnosis approach in hybrid systems must be able to use the information at both DES and continuous levels for detecting and isolating faults. We motivate the need for this approach through the following examples.

Consider the system of two cascade tanks and three valves in Figure 3.1. A chemical with temperature T_1 flows into Tank1 through valve V_1. Another chemical with temperature T_2 flows into Tank1 through the valve V_2. The chemicals are mixed in Tank1 and then flow to Tank2, where they are drained through pipe d. When the chemicals are mixed, the temperature of the resultant is governed by a differential equation in terms of T_1 and T_2. Some of the mixed chemicals are pumped back by pump P and through the valve V_3 to Tank1. The chemical flowing into Tank1 through valve V_3 has the same temperature as in Tank1. When all three valves are open and the pump is working, the height of the chemical in Tank1 is at H2. Assume that all the three valves are open, and pump P is working. Also assume that the valves may have a loss-of-effectiveness fault when they are commanded to open fully. Pump P is assumed fault free. Assume that only one fault may be present at a time in the system. Suppose that a discrete level sensor with two readings H1 and H2 monitors the liquid level and a temperature sensor measures the temperature of the liquid in the tank and generates a continuous signal. The readings of the discrete sensor will represent the discrete outputs in the hybrid automaton model of the system.

The continuous output of the hybrid automaton model will be a function of the temperature sensor readings. First, assume that only the readings of the level sensor are available. In this case, if one of the valves fails, the level of the chemical in Tank1 reaches H1 and the fault can be detected. However, it is not possible to determine the valve in which the fault has occurred, i.e, the fault cannot be isolated. Now assume that only the signal generated by the temperature sensor is at hand. Thus, if one of the valves V_1 or V_2 fails, the temperature of the liquid changes and the corresponding fault can be detected and isolated. However, if valve V_3 fails, the temperature of the liquid does not change and the fault of valve V_3 cannot be detected and isolated. Now assume that the readings of both sensors are available. Using the reading of the level sensor the faults can be detected. Faults in V_1 and V_2 change the temperature and therefore, can be isolated using the temperature sensor. A fault in V_3 can be isolated when the temperature does not change, but the level of

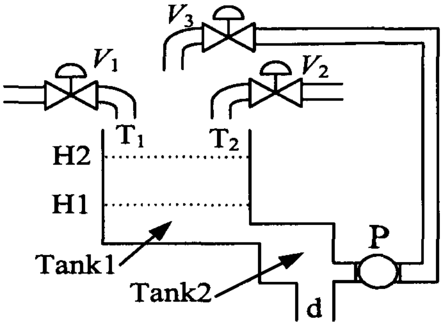


Figure 3.1: A system of tank and two valves.

the liquid reaches H1. Therefore, if any of the valves fails, the fault can be detected and isolated in the system.

As another example demonstrating the advantages of integrating the information at the DES level with the readings of the continuous sensors for fault diagnosis, consider the system shown in Figure 3.2. The system consists of a gas turbine engine and its fuel supply system and nozzle actuator. Next, we briefly describe the system. The details of modeling and fault diagnosis in the gas turbine engine are provided in Chapter 6.

The gas turbine engine is composed of six sections: intake duct, compressor, combustion chamber, turbine, afterburner and nozzle. The air is taken into the engine through the intake duct and compressed with the compressor. Fuel is then added to the air, and the mixture is burned in the combustion chamber. The high-pressure and high-temperature gases produced turn the turbine. Then they are reheated using the afterburner and expanded through the nozzle to produce thrust. A fuel supply system provides the fuel to the engine. The fuel supply system has two branches: one controlling the fuel mass flow rate entering the combustion chamber and the other controlling the fuel entering the afterburner. The pumps P_M and P_{AB} are in charge of pumping the fuel to the engine. The governors G_M and G_{AB} are servo-valves that control the mass flow rate of the fuel passing through them. The shut-off valves V_S is used to ensure that no fuel is supplied to the engine when the

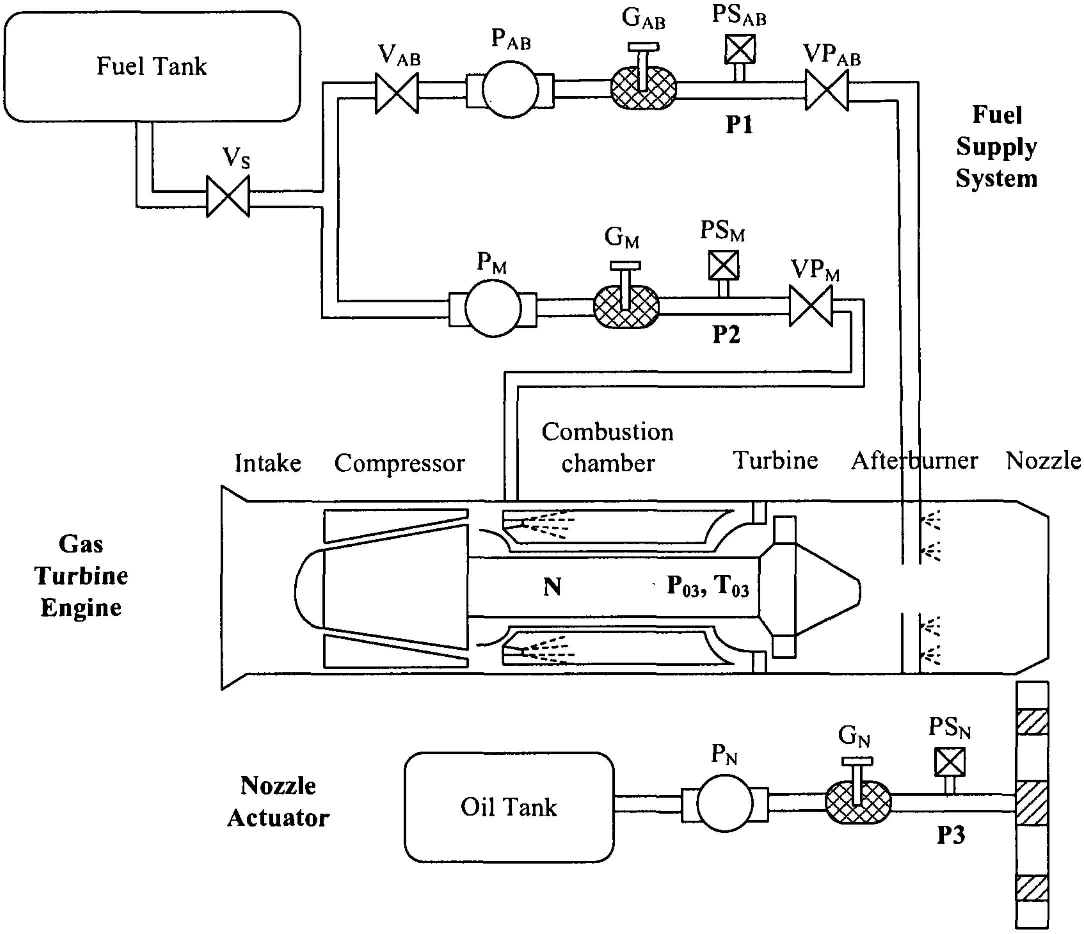


Figure 3.2: A gas turbine engine and its fuel supply system and nozzle actuator.

engine is off.

The shut-off valve V_{AB} is used to ensure that no fuel is supplied to the afterburner when the afterburner is not in operation. The pressurizing valves VP_M and VP_{AB} ensure that the fuel entering the engine has a high pressure for efficient atomization. The area of the nozzle is variable and changes with a hydraulic actuator. The pump P_N pumps the oil to the hydraulic actuator. The governor G_N controls the area by controlling the amount of oil passing through it. There are three discrete sensors PS_{AB}, PS_M and PS_N installed in the fuel supply system and the nozzle actuator generating "low" and "high" symbols at P_1, P_2 and P_3. There are also continuous sensors installed in the engine to measure the turbine inlet temperature T_{03}, turbine inlet pressure p_{03} and shaft speed N. It can be verified that the effects of the afterburner fuel mass flow rate and the nozzle area on the continuous sensors are opposite and proportional. Therefore, the failure modes in the components of the afterburner fuel supply system cannot be distinguished from the failure modes in the nozzle actuator.

Moreover, discrete sensors cannot sense small deterioration of the components. For example, consider the following faults in the components: small loss-of-effectiveness of G_M and stuck-closed of V_S. Suppose a small loss-of-effectiveness fault in G_M occurs. This fault cannot be picked up by using discrete sensors alone. Also, it can be verified that by using residual generators, the loss-of-effectiveness fault in G_M cannot be distinguished from the stuck-closed in V_S. However, integratg the information coming from the discrete sensors and the information provided by the residual generators, the fault in G_M can be isolated.

Undiagnosable failure modes in the engine may become diagnosable if more sensors are used. However, it should be noted that installing sensors in the engine may be difficult or very costly due to the very high temperature and pressure in the engine.

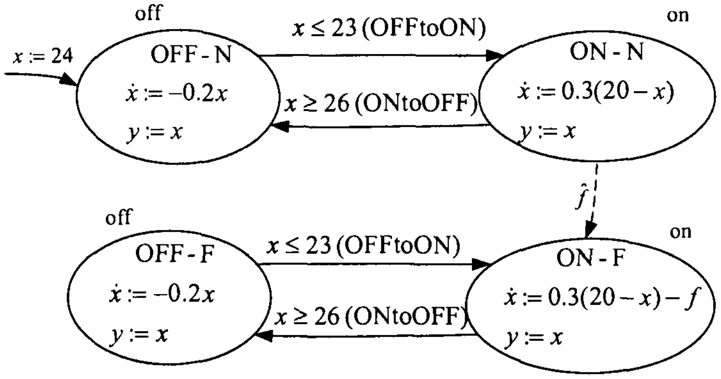


Figure 3.3: A hybrid automaton modeling a heating system with a fault.

We develop a systematic approach for fault diagnosis in hybrid systems in which information at both DES and continuous levels is used for detecting and isolating faults. The rest of this chapter is organized as follows. In Section 3.1, we describe the modeling of the system in our framework. Our diagnosis method developed for hybrid automata is described in Section 3.2. We conclude the chapter by presenting a summary of the chapter.

3.1 System Model

In this section, we describe the modeling of a hybrid automaton with faults in our framework.

3.1.1 Hybrid Automata

Figure 3.3 shows a hybrid automaton modeling a heating system consisting of an electrical heater and a thermostat. The heater is energized with an electrical power supply unit. Suppose that the heating system is in charge of keeping a room's temperature in the range of 23^o to 26^o. The temperature is denoted by x. The continuous output is denoted by y. There are two discrete states for the normal behavior of the system: ON-N and OFF-N. Assume that initially, the temperature is 24^o; the temperature is in the desired range and therefore, the heater is off. When the heater is off, the temperature of the room falls according to $\dot{x} = -0.2x$. When the temperature reaches 23^o, the power supply unit energizes the heater with maximum power.

The heater turns on and the system transitions to the discrete state ON-N. A discrete event (OFFtoON) is associated with this autonomous transition. While the system is in the discrete state ON-N, the heater may have a loss-of-effectiveness fault. If the heating system fails, the system transitions to the discrete state ON-F with an unobservable transition labeled with $\hat{f}$ shown by a dashed line. The loss-of-effectiveness fault is modeled by a fault signal f that is added to the state flow dynamics of the system in the discrete state ON-F. There are two sensors available in the system: a temperature sensor that generates a continuous signal, and a discrete sensor that by using the status of the power supply unit, generates the symbols "on" and "off" at the discrete states of the system. The system has two operational modes: the normal mode and the failure mode. The discrete states OFF-N and ON-N belong to the normal mode and the discrete states OFF-F and ON-F belong to the failure mode.

We assume that the system under control, i.e., the system along with low-level continuous controllers and DES supervisors, can be modeled as a hybrid automaton. In the following, we present a formal definition of hybrid automata in our work which is a modified form of the definition in [60]. Our definition models the faulty behavior of hybrid automata. Moreover, in our definition, there is a discrete output associated with each discrete states of the hybrid automata.

Definition 3.1.1. *A* ***hybrid automaton*** *is defined to be a 14-tuple*

$$H = (Q, \mathcal{X}, \mathcal{U}, \mathcal{Y}, FT, Init, S, \Sigma, T, G, \rho, D, \lambda, q_0) \tag{3.1}$$

where Q is the set of finite discrete states; $\mathcal{X} \subseteq \mathbb{R}^n$, $\mathcal{U} \subseteq \mathbb{R}^p$ and $\mathcal{Y} \subseteq \mathbb{R}^r$ are the *set of vector spaces of continuous state, control input and output, respectively;* FT *is the set of* m *fault types* $f^1, \cdots, f^m$ *with* $f^i(t) \in \mathbb{R}$ *for* $1 \leq i \leq m$; $Init \subseteq \mathcal{X}$ *is the set of initial continuous states;* $S = \{S_q \mid q \in Q\}$ *is the set of dynamic models defining the continuous dynamics of the system;* Σ *is a set of symbols representing the discrete events labeling the transitions among discrete states;* $T \subseteq Q \times \Sigma \times Q$ *is the set of discrete transitions;* $G : T \times \mathcal{X} \times \mathcal{U} \longrightarrow \{True, False\}$ *is the set of guard conditions;* $\rho : T \times \mathcal{X} \longrightarrow \mathcal{X}$ *is a reset map;* D *is the set of discrete output symbols;* $\lambda : Q \longrightarrow D$ *is the discrete output map and* q_0 *is the initial discrete state.* ■

For instance, in the heating system example, we have:

$Q = \{OFF - N, ON - N, OFF - F, OFF - N\}$;

$\mathcal{X}, \mathcal{U}, \mathcal{Y} \subseteq \mathbb{R}$;

$FT = \{f\}$;

$Init = \{x = 24\}$;

$$S = \{S_{OFF-N}, S_{ON-N}, S_{OFF-F}, S_{ON-F}\} \text{ with } S_{OFF-N} = S_{OFF-F} = \begin{cases} \dot{x} = -0.2x \\ y = x \end{cases},$$

$$S_{ON-N} = \begin{cases} \dot{x} = 0.3(30 - x) \\ y = x \end{cases}, S_{ON-F} = \begin{cases} \dot{x} = 0.3(30 - x) - f \\ y = x \end{cases};$$

$\Sigma = \{OFFtoON, ONtoOFF, \hat{f}\}$;

$T = \{T_1, T_2, T_3, T_4, T_5\}$ with $T_1 = (OFF - N, OFFtoON, ON - N)$,

$T_2 = (ON - N, ONtoOFF, OFF - N), T_3 = (ON - N, \hat{f}, ON - F)$,

$T_4 = (OFF - F, OFFtoON, ON - F), T_5 = (ON - F, ONtoOFF, OFF - F)\}$

$G(T_1, \{x \mid x < 23\}, \mathbb{R}) = True, G(T_1, \{x \mid x \geq 23\}, \mathbb{R}) = False$,

$G(T_2, \{x \mid x > 26\}, \mathbb{R}) = True, G(T_2, \{x \mid x \leq 23\}, \mathbb{R}) = False$,

$G(T_3, \mathbb{R}, \mathbb{R}) = True$,

$G(T_4, \{x \mid x < 23\}, \mathbb{R}) = True, G(T_4, \{x \mid x \geq 23\}, \mathbb{R}) = False$,

$G(T_5, \{x \mid x > 26\}, \mathbb{R}) = True, G(T_5, \{x \mid x \leq 23\}, \mathbb{R}) = False$;

$\rho(tr, x) = x$ for any $tr \in T$ and $x \in \mathcal{X}$; $D = \{off, on\}$;

$\lambda(OFF - N) = \lambda(OFF - F) = off, \lambda(ON - N) = \lambda(ON - F) = on$;

$q_0 = OFF - N$

In the above definition of hybrid automata, a discrete event is associated with any transition between two discrete states. In other words, in addition to the guard conditions, every arc in a hybrid automaton is labeled by a discrete event. Every transition $tr \in T$ can be represented by a triple $tr = (q_1, \sigma, q_2)$, with $q_1, q_2 \in Q$ and $\sigma \in \Sigma$. Here, q_1 is the *source* of t, q_2 is the *destination* of tr and σ is the *label* of tr. The map $label : T \longrightarrow \Sigma$ gives the label of the transitions. In general, we assume two types of transitions:

- Transitions which have a discrete nature such as commands from a DES supervisor. The guard conditions for these transitions are always true.

- Autonomous transitions that depend on the continuous state and control input of the system. We assume that when the guard conditions of these transitions become true, the system may stay in the source state or have a transition to the destination state.

For any discrete state q, S_q is the dynamical model

$$S_q := \begin{cases} \dot{x}(t) = J_q(x(t), u(t), f^1(t), \cdots f^m(t)) \\ y(t) = M_q(x(t), u(t)) \end{cases} \tag{3.2}$$

where $x(t) \in \mathcal{X}$, $u(t) \in \mathcal{U}$ and $f^i \in FT$ for $0 \leq i \leq m$ are the continuous state, control input and fault types respectively. Functions

$$J_q : \mathcal{X} \times \mathcal{U} \times FT \rightarrow \mathbb{R}^n$$

and

$$M_q : \mathcal{X} \times \mathcal{U} \rightarrow \mathcal{Y}$$

define the state flow and the output map of the continuous dynamics at q, respectively.

The dynamical system of equation (3.2) can model linear or nonlinear dynamics with faults. For example, the continuous dynamics at the discrete states of the hybrid automaton model of the heating system in Figure 3.3 is represented by a linear system.

As described earlier, observations in the system come from the readings of the discrete and continuous sensors. The discrete output at each discrete state is generated by using the readings of discrete sensors. The continuous output at each discrete state, on the other hand, is a function of the signals generated by continuous sensors.

It can be seen from Def. 3.1.1 that the tuple $(Q, \Sigma, T, D, \lambda, q_0)$ defines a Moore FSA representing a DES level abstraction of the system. We refer to the tuple $H_{abs} = (Q, \Sigma, T, D, \lambda, q_0)$ as the **DES abstraction** of H. We assume that $\Sigma = \Sigma_o \dot{\bigcup} \Sigma_{uo}$, where Σ_o represents the observable event set and Σ_{uo} consists of unobservable events. The commands that are generated by a supervisor or the events generated by discrete sensors are the examples of observable events. The set $\Sigma_f \subseteq \Sigma_{uo}$ is the set of fault events.

Next, we describe the modeling of faults in hybrid automata and describe the faulty behavior of the system.

3.1.2 Fault Modeling in Hybrid Automata

In general, we have two kinds of faults in the system: faults whose effect on the system's dynamics is drastic such as a stuck-on fault of a heater or a stuck-open fault of a valve, and faults that slightly change the system dynamics such as a small bias in a sensor reading or a small drift in the output of a valve. In this thesis, we study fault diagnosis in hybrid systems with both kinds of faults.

The system can be in normal mode of operation or in a failure mode corresponding to a fault. In our work, faults are represented by **fault types**. Each fault type corresponds to one or more **failure modes** in a component of the system. Initially, the system is in normal mode of operation and fault types have zero values. When a fault occurs in the system, the value of the corresponding fault type becomes nonzero and the system enters the failure mode corresponding to that fault.

Definition 3.1.2. *A fault type f is called* **active** *in a discrete state q if $f(t)$ is a nonzero function for the time that the system is in q.*

For instance, in the heating system example, f is an additive fault type which represents the loss-of-effectiveness fault. Additive fault types are modeled by adding fault signals to the state flow dynamics as inputs. Fault type f is active in the state $ON - F$. When the heating system fails it enters the failure mode F which corresponds to the loss-of-effectiveness fault. The unobservable event $\hat{f}$ is a fault event and represents the occurrence of the fault; the discrete states ON and OFF belong to the normal mode; and $ON - F$ and $OFF - F$ belong to the failure mode F.

As another example, consider the hybrid automaton of Figure 3.4. This hybrid automaton models a solenoid valve. The flow rate passing through the valve shown by the variable x is related to the input voltage u by a linear dynamical system. The valve may fail stuck-closed or stuck-open, or it may have a 10% loss-of-effectiveness fault. Each fault takes the system into a specific failure mode. The failure modes F_1 (respectively, F_2) corresponds to the operational mode of the valve when it fails stuck-closed (respectively, stuck-open). Failure mode F_3 corresponds to the operational mode of the valve when it has a 10% loss-of-effectiveness fault, and N corresponds to the normal mode of operation.

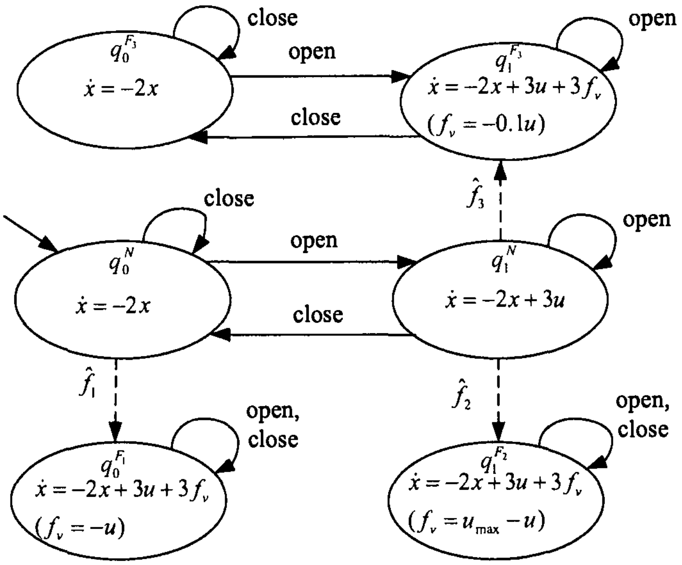


Figure 3.4: A hybrid automaton modeling a solenoid valve with three faults.

The faults in the valve can be represented by an additive fault type f_v. Each fault can be specified by a specific value assigned to f_v. For instance, the stuck-closed fault can be specified by $f_v = -u$, the stuck-open fault can be specified by $f_v = u_{max} - u$, where u_{max} is the maximum of the input, and a 10% loss-of-effectiveness fault can be represented by $f_v = -0.1u$. In Figure 3.4, the fault events $\hat{f}_1$, $\hat{f}_2$ and $\hat{f}_3$ represent the occurrence of faults stuck-closed, stuck-open and a 10% loss-of-effectiveness, respectively.

Each fault type represents a set of failure modes in a component with each mode corresponding to a set of (nonzero) fault signals. The system is taken to a failure mode of operation when a fault occurs. Assume that there are $\hat{m}$ failure modes in the system ($\hat{m} \geq m$), and let FM be the set of failure modes in the system. The mapping

$$\xi : FT \longrightarrow 2^{FM} - \{\emptyset\}$$

yields the failure modes associated with a fault type.

We assume that faults are **permanent**. This means that if the system enters a failure mode it stays in that failure mode forever. It should be noted that fault types are not necessarily active in all the discrete states of their corresponding failure mode. This typically happens when a faulty component is not used or is off in a given mode of operation. For example, in the system in Figure 3.3, f is not active in $OFF - F$.

Let F^j be a failure mode associated with the fault type f^i for $i \in \{1, \cdots, m\}$. We say F^j occurs at time t_0 if $f^i(t_0) = 0$ for $t < t_0$ and for $t \geq t_0$, $f^i(t)$ takes a value corresponding to F^j. The occurrence of a failure mode F^j is modeled by an unobservable **fault event** denoted by $\hat{f}^j$ at the DES level. Let

$$\mathcal{K} = \{N, F^1, \cdots, F^{\hat{m}}, F^{1,2}, \cdots, F^{\hat{m}-1,\hat{m}}, \cdots, F^{1,\cdots,\hat{m}}\}$$

denote the **condition set**. The system can be in normal condition (N) or a condition corresponding to a combination of failure modes. Thus, for example, for a system with two failure modes F^1 and F^2, the condition set will be $\mathcal{K} := \{N, F^1, F^2, F^{1,2}\}$, where $F^{1,2}$ corresponds to the case where both failure modes F^1 and F^2 have occurred. We assume that the discrete state set can be partitioned according to the condition of the system:

$$Q = Q_N \dot{\bigcup} (Q_{F^1} \dot{\bigcup} \cdots \dot{\bigcup} Q_{F^{\hat{m}}}) \dot{\bigcup} (Q_{F^{1,2}} \dot{\bigcup} \cdots \dot{\bigcup} Q_{F^{\hat{m}-1,\hat{m}}}) \dot{\bigcup} \cdots \dot{\bigcup} Q_{F^{1,\cdots,\hat{m}}}$$

For instance, in Figure 3.3, we have $\mathcal{K} = \{N, F\}$, $Q_N = \{OFF - N, OFF - N\}$ and $Q_F = \{OFF - F, ON - F\}$.

Let $\Sigma_{\hat{f}} = \{\hat{f}_1, \cdots, \hat{f}_{\hat{m}}\} \subseteq \Sigma_{uo}$ denote the set of fault events. The fault event $\hat{f}^i$ labels the occurrence of the failure mode F^i for $i \in \{1, \cdots, \hat{m}\}$. Function $\kappa : Q \rightarrow \mathcal{K}$ denotes the **condition map** of the system, and is defined such that for every $q \in Q$, $\kappa(q)$ is the condition of the system at the discrete state q. We say q is a **normal discrete state** if $\kappa(q) = N$, otherwise it is called a **faulty discrete state**. The definition of κ can be extended to the subsets of Q:

$$\kappa(z) = \{\kappa(q) \mid q \in z\}, \text{ for any } z \subseteq Q$$

Let $\mathcal{F} = \mathcal{K} - \{N\}$ denote the set of **faulty conditions**. Also let $\mathcal{F}^i$ be the set of faulty conditions in which the failure mode F^i is present, and let $\overline{\mathcal{F}^i} = \mathcal{F} - \mathcal{F}^i$. For example, consider a system with three failure modes F^1, F^2 and F^3 and assume that

$$\mathcal{K} = \{N, F^1, F^2, F^3, F^{1,2}, F^{1,3}, F^{2,3}, F^{1,2,3}\}$$

We have $\mathcal{F} = \{F^1, F^2, F^3, F^{1,2}, F^{1,3}, F^{2,3}, F^{1,2,3}\}$ and $\mathcal{F}^1 = \{F^1, F^{1,2}, F^{1,3}, F^{1,2,3}\}$.

Fault diagnosis is to find the condition of the system by detecting and isolating failure modes in a bounded time after they occur in the system. In the next section, we describe our diagnosis method for hybrid automata.

3.2 Hybrid Diagnosis of Hybrid Automata

Fault diagnosis is to detect the occurrence of the failure modes and isolate the failure mode that the system is in. In our framework, the DES abstraction of the system includes all the failure modes present in the system. Therefore, if the discrete state in which the system is evolving can be identified at each time, the fault diagnosis can be accomplished. However, the occurrence of failure modes is modeled by unobservable events and therefore, using only the information at the DES level (discrete outputs) may not be sufficient for diagnosis and diagnosability analysis.

As described in the tank and valve example, using only the DES abstract model of a system, some failure modes may remain undiagnosable. On the other hand, it is possible that no isolator can be designed for isolating some fault types. By integrating the information available from discrete and continuous sensors and the DES level and continuous level models, we may be able to provide a more precise diagnosis and diagnose failure modes that are undiagnosable based on only DES or only continuous models.

We assume that there is a bank of residual generators (isolators) designed based on the continuous dynamics to isolate fault types at the continuous level. Each residual generator takes the continuous input and output of the system and produces a residual. The solvability conditions for the existence of residual generators for isolating faults in linear dynamical systems and nonlinear dynamical systems have been studied in [80] and [33], respectively. In the remainder of the thesis, we refer to the residual generators designed for detection and isolation of faults as isolators.

We develop a method for constructing a hybrid diagnoser that integrates the information generated by the isolators with the information available at the DES level (outputs of discrete sensors) to diagnose failures. Figure 3.5 shows the schematic of our diagnoser design methodology. First, a DES abstraction of the information generated by each isolator is constructed which represents the isolator. The DES model of the isolators is then integrated with the DES abstraction of the system to construct an **Extended Discrete-Event System Abstraction (EDESA)** of the system and isolators. The hybrid diagnoser is a diagnoser which is designed based on the EDESA of the system and isolators. Although the diagnoser is constructed based on a DES model, we denote it a hybrid diagnoser because it uses the information from both the discrete and the continuous dynamics.

In general, we follow the following four steps to construct the EDESA:

1. Each isolator in the system is modeled by a DES;

2. The DES abstraction of the system (H_{abs}) is modified (by adding appropriate self-loop transitions) to make the transitions in the DES models of isolators (step 1) consistent with the system transition system;

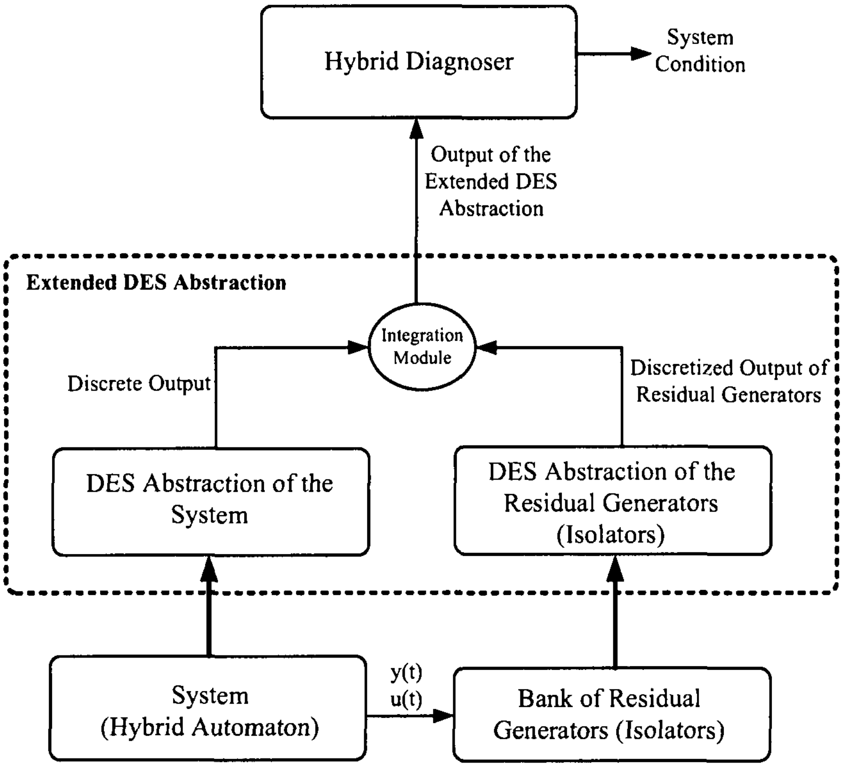

Hybrid Diagnoser
System Condition
Output of the Extended DES Abstraction
Extended DES Abstraction
Integration Module
Discrete Output
Discretized Output of Residual Generators
DES Abstraction of the System
DES Abstraction of the Residual Generators (Isolators)
System (Hybrid Automaton)
y(t)
u(t)
Bank of Residual Generators (Isolators)

Figure 3.5: The schematic of the hybrid diagnosis framework.

3. We assume that the isolators generate an event between every two consecutive transition of the system. This assumption is enforced by an appropriate DES model. This will be discussed in more detail subsequently.

4. The EDESA will be constructed by combining the DES models of the isolators, modified DES abstraction of the system and the DES enforcing the assumptions (step 3) using the synchronous product operation.

In the following, we first describe the modeling of isolators that are designed at the continuous level in our work. We then describe a systematic approach to perform each of the aforementioned steps for building the EDESA of a hybrid automaton and isolators.

3.2.1 Modeling of Isolators

Isolators are designed based on the continuous dynamics of each discrete state. The solvability conditions for the existence of isolators for isolating faults in linear systems has been studied in [80]. In linear systems faults in the system can be modeled as additive faults added to the state-flow of the system similar to inputs. In [80], it is shown that additive fault signals can be used to model the faults in the actuators, sensors and the structure of the system. In [33], isolator design for isolating faults in nonlinear dynamical systems has been studied and conditions for the existence of isolators has been provided. In these nonlinear systems, faults are also modeled by additive fault signals.

In this work, we assume that faults can be modeled by additive fault type signals. Therefore, the dynamics of the system at each discrete state can be represented by

$$S_q := \begin{cases} \dot{x} = E_q(x,u) + G_q(f) \\ y = M_q(x,u) \end{cases} \tag{3.3}$$

where f is a subset of the fault types present in the system.

Two discrete states q_1 and q_2 are called **EM-similar** if $E_{q_1}(x,u) = E_{q_2}(x,u)$ and $M_{q_1}(x,u) = M_{q_2}(x,u)$ (for every x, u). For instance, in Fig. 3.3, ON-N and ON-F are EM-similar.

Assume that there are d distinct pairs (E,M) for the continuous dynamics of the system. The discrete state set of the system can be partitioned based on the (E,M) pairs:

$$Q = Q^{EM_1} \dot{\bigcup} Q^{EM_2} \cdots \dot{\bigcup} Q^{EM_d}$$

where the continuous dynamics of all the discrete states in every Q^{EM_i} for $i \in \{1, \cdots, d\}$ have the same (E,M). We group EM-similar discrete states and design a set of isolators based on the continuous dynamics of each group.

Let Q^{EM} be a set of EM-similar states and let $FT^{Q^{EM}} = \{f_1^{Q^{EM}}, \cdots, f_l^{Q^{EM}}\}$ be the set of fault types active in the discrete states of Q^{EM}. Each isolator designed for Q^{EM} takes the continuous input and output of the system and produces a residual vector to isolate some fault types from the others while the system is evolving in one of the discrete states of Q^{EM}. For simplicity, we assume that the isolators are initialized to zero. *We also assume that the system stays in each discrete state for at least τ^{min}, and the isolators are designed so that their responses reach steady state in less than τ^{min}.* More precisely, the transient response due to the mismatch in the initial conditions of isolators and the system dies out in less than τ^{min}. Let $\Phi \subseteq FT^{Q^{EM}}$ be a nonempty subset of fault types. The isolator $Is^{Q^{EM}}(\Phi)$ is designed to be sensitive to the fault types of Φ and be insensitive to the fault types of $FT^{Q^{EM}} - \Phi$.

Assumption 1. *An isolator $Is^{Q^{EM}}(\Phi)$ designed for isolating fault types in Q^{EM} generates a nonzero residual for all inputs $u \in \mathcal{U}$ after the transient response due to the mismatch in the initial conditions of $Is^{Q^{EM}}(\Phi)$ and the system dies out if the system is not evolving in one of the discrete states of Q^{EM}.* ■

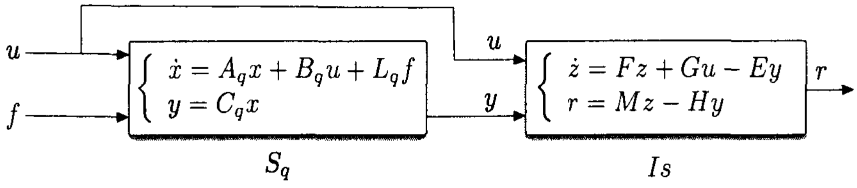


Figure 3.6: Block diagram of the system and the isolator Is when the system is in q.

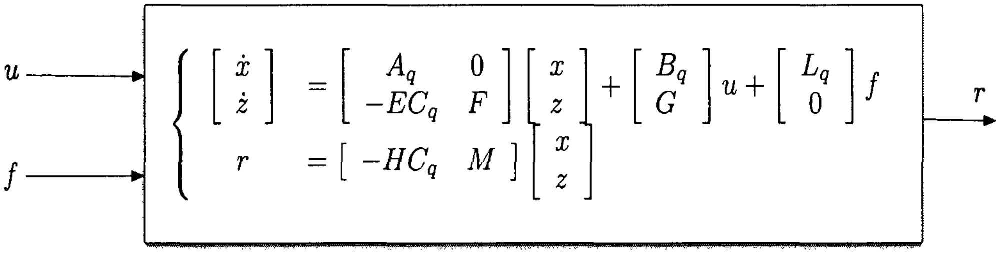

u
f
$\begin{bmatrix} \dot{x} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} A_q & 0 \\ -EC_q & F \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} + \begin{bmatrix} B_q \\ G \end{bmatrix} u + \begin{bmatrix} L_q \\ 0 \end{bmatrix} f$
$r = \begin{bmatrix} -HC_q & M \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix}$
r

Figure 3.7: The augmented dynamics of the system and the isolator Is when the system is in q.

Assumption 1 is not a very limiting assumption because the continuous dynamics of the system in the discrete states of Q^{EM} is different from the continuous dynamics of the system in any other discrete state, and therefore, it is unlikely that the isolator designed based on the dynamics of Q^{EM} generates a zero residual while the system is not evolving in one of the discrete states of Q^{EM}. In this work, we show how Assumption 1 can be verified for a hybrid automaton with linear dynamics.

Let $q \in Q - Q^{EM}$ and Is be a linear isolator with dynamics as in (2.4) designed based on the dynamics of Q^{EM}. Figure 3.6 shows the block diagram of the system and the isolator Is when the system is in q. The isolator Is takes the continuous input and output of the system and produces a residual vector r. Figure 3.7 shows the augmented dynamics of the system and the isolator Is when the system is in the state q. In order to have a nonzero residual r, we must verify that the transfer matrix from f and u to r is not zero for all $q \in Q - Q^{EM}$. A more restrictive condition is that for all $q \in Q - Q^{EM}$, the series system shown in Figures 3.7 is input observable:

$$Im\ [B_a\ L_a] \bigcap < Ker\ C_a | A_a > \ = \emptyset$$

where $A_a = \begin{bmatrix} A_q & 0 \\ -EC_q & F \end{bmatrix}$, $B_a = \begin{bmatrix} B_q \\ G \end{bmatrix}$, $L_a = \begin{bmatrix} L_q \\ 0 \end{bmatrix}$ and $C_a = \begin{bmatrix} -HC_q & M \end{bmatrix}$.

An isolator $Is^{Q^{EM}}(\Phi)$ is a detection and isolation filter designed based on the continuous dynamics of Q^{EM}. It initializes with zero, takes the continuous input and output of the system and produces a residual vector $r_{\Phi}^{Q^{EM}}(t)$ with the following properties.

- $||r_{\Phi}^{Q^{EM}}(t)|| \geq \epsilon_{\Phi}^{Q^{EM}}$ after the transient response due to the mismatch in the initial conditions of isolators and the system dies out if the system is evolving in one of the discrete states of Q^{EM} and a fault type of Φ is active.

- $||r_{\Phi}^{Q^{EM}}(t)|| < \epsilon_{\Phi}^{Q^{EM}}$ after the transient response due to the mismatch in the initial conditions of isolators and the system dies out if the system is evolving in one of the discrete states of Q^{EM} and no fault type is active or one of the fault types of $FT^{Q^{EM}} - \Phi$ is active.

- $||r_{\Phi}^{Q^{EM}}(t)|| \geq \epsilon_{\Phi}^{Q^{EM}}$ after the transient response due to the mismatch in the initial conditions of isolators and the system dies out if the system is not evolving in one of the discrete states of Q^{EM}.

where $\epsilon_{\Phi}^{Q^{EM}} \geq 0$ is a threshold chosen to evaluate the residual. The choice of this threshold has been explained in Section 2.4.

In the case that $\Phi = FT^{Q^{EM}}$, $Is^{Q^{EM}}(\Phi)$ will be the isolator sensitive to all the fault types present in Q^{EM}, or simply a fault detector. An **observer** designed based on the continuous dynamics of Q^{EM} without considering faults can be a fault detector. If $\Phi = \emptyset$, $Is^{Q^{EM}}(\Phi)$ will be an isolator which generates a zero residual if the system is in any state of Q^{EM} and produces a nonzero residual vector if the system is in any state of $Q - Q^{EM}$. In this case, $Is^{Q^{EM}}(\Phi)$ is called an **EM-distinguisher**. In the case that no fault type is active in Q^{EM}, i.e., $FT^{Q^{EM}} = \emptyset$, the continuous dynamics of all $q \in Q^{EM}$ will be similar, and an observer designed based on the dynamics of any $q \in Q^{EM}$ will be an EM-distinguisher.

Let $Q^{EM}_{inact} \subseteq Q^{EM}$ be the set of discrete states of Q^{EM} in which no fault type is active:

$$Q^{EM}_{inact} = \{q | \ q \in Q^{EM} \text{ and } f(\cdot) \equiv 0 \text{ in } q \text{ for all } f \in f^{Q^{EM}}\}$$

The function *active* : $Q \longrightarrow 2^{FT}$ yields the fault types active in a discrete state.

$$active(q) = \{f \mid f \in FT \text{ and } f \text{ is active in } q\}$$

The mapping *active* can also be extended to state sets as follows. Let $\overline{Q} \subseteq Q$, then

$$active(\overline{Q}) = \bigcup_{q \in \overline{Q}} active(q)$$

The inverse function $active^{-1} : 2^{FT} \longrightarrow 2^{Q}$ maps a set of fault types of FT to the set of discrete states of Q in which the fault types are active. Let $\Phi \subseteq FT$, then

$$active^{-1}(\Phi) = \{q \mid active(q) \bigcap \Phi \neq \emptyset\}$$

For a single fault type f, we have:

$$active^{-1}(\{f\}) = \{q \mid q \in Q \text{ and } f \text{ is active in } q\}$$

Also, we have:

$$active^{-1}(\Phi) = \bigcup_{f \in \Phi} active^{-1}(\{f\})$$

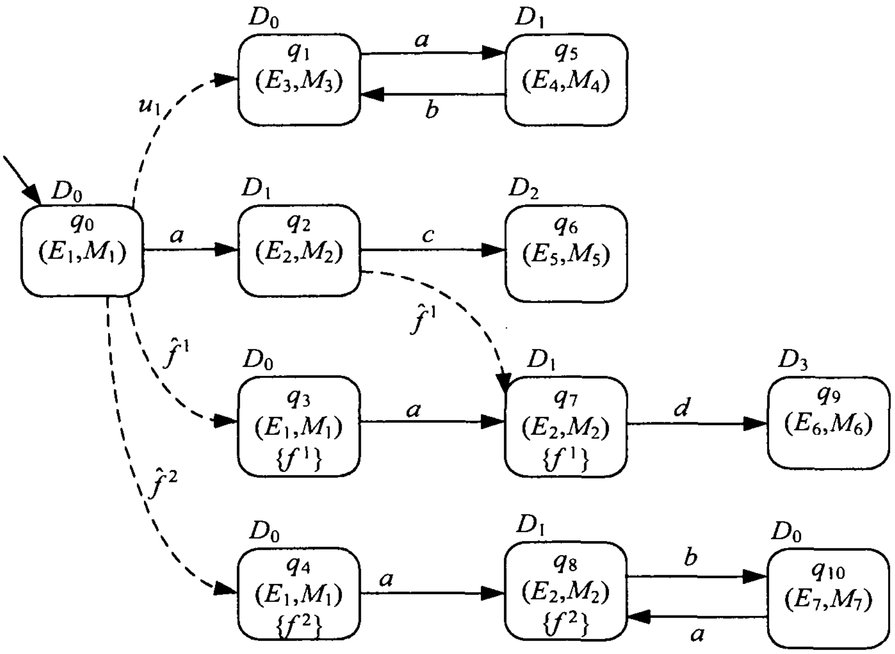


Figure 3.8: Example 3.2.1: DES abstraction of a hybrid automaton with two failure modes.

We have $Q_{inact}^{EM} = active^{-1}(\emptyset) \bigcap Q^{EM}$.

Example 3.2.1. *The graph of the DES abstraction of a hybrid automaton is shown in Figure 3.8. In each discrete state in Figure 3.8, the EM group the state belongs to, the fault type active in the state and the discrete output generated by the system at that state are also displayed. Two fault types f^1 and f^2 with the corresponding failure modes F^1 and F^2 are assumed. Fault type f^1 may occur when the system is in the discrete state q_0 or q_2. Fault type f^2 may occur when the system is in q_0. The occurrence of the failure modes F^1 and F^2 are modeled by transitions labeled with the events $\hat{f}^1$ and $\hat{f}^2$, respectively. Symbol 'u' is an unobservable event changing the discrete state of the system in the normal mode. The discrete stste set is $Q = \{q_0, \cdots, q_{10}\}$. The condition set of the system is $\mathcal{K} = \{N, F^1, F^2\}$. The discrete states of the system can be partitioned according to the condition of the system. Here, $Q_N = \{q_0, q_1, q_2, q_5, q_6\}$, $Q_{F^1} = \{q_3, q_7, q_9\}$ and $Q_{F^2} = \{q_4, q_8, q_{10}\}$. In this example, we assume that the discrete states q_0, q_3 and q_4 are EM-similar discrete* *states with functions (E^1, M^1): $Q^{E^1M^1} = \{q_0, q_3, q_4\}$. Also the discrete states q_2, q_7 and q_8 are EM-similar discrete states with functions (E^2, M^2): $Q^{E^2M^2} = \{q_2, q_7, q_8\}$. The rest of the discrete states have different (E, M) functions.*

The fault type f^1 is active in the discrete states q_3 and q_7, i.e., $active^{-1}(f^1) = \{q_3, q_7\}$. We also have $active^{-1}(f^2) = \{q_4, q_8\}$, $active(q_1) = \emptyset$, $Q^{E_1M_1}_{inact} = active^{-1}(\emptyset) \bigcap Q^{E_1M_1} = \{q_0\}$ and $Q^{E_2M2}_{inact} = \{q_2\}$. ■

Let $W \subseteq Q^{EM}$ denote the set of discrete states that the system can be in when a fault type of Φ becomes active: $W = active^{-1}(\Phi) \bigcap Q^{EM}$. An isolator $Is^{Q^{EM}}(\Phi)$ distinguishes the set of discrete states W from $Q^{EM} - W$. The properties of the residual vector $r^{Q^{EM}}_{\Phi}(t)$ produced by $Is^{Q^{EM}}(\Phi)$ can be expressed as follows.

- $||r^{Q^{EM}}_{\Phi}(t)|| < \epsilon^{Q^{EM}}_{\Phi}$ in steady state if the system is evolving in one of the discrete states of $Q^{EM} - W$;
- $||r^{Q^{EM}}_{\Phi}(t)|| \geq \epsilon^{Q^{EM}}_{\Phi}$ in steady state if the system is evolving in one of the discrete states of $(Q - Q^{EM}) \bigcup W$;

We assume that there is a signal processing unit that takes residual vector $r^{Q^{EM}}_{\Phi}(t)$ and generates an output signal $\tilde{r}^{Q^{EM}}_{\Phi}(t)$ as follows:

$$\tilde{r}^{Q^{EM}}_{\Phi}(t) = \begin{cases} 0 & \text{if } ||r^{Q^{EM}}_{\Phi}(t)|| < \epsilon^{Q^{EM}}_{\Phi} \text{ in steady state} \\ 1 & \text{if } ||r^{Q^{EM}}_{\Phi}(t)|| \geq \epsilon^{Q^{EM}}_{\Phi} \text{ in steady state} \end{cases}$$

The binary output of the signal processing unit allows us to model an isolator with a DES. In the following, we denote the binary signal $\tilde{r}^{Q^{EM}}_{\Phi}(t)$ when we refer to the output of the isolator $Is^{Q^{EM}}(\Phi)$.

3.2.2 Modeling the Isolators with DES

Every isolator in our work can be modeled as a finite-state Moore automaton with two states ZERO and ONE. Let $\mathbf{IS}_{tot}$ be the set of all isolators designed based on the continuous dynamics of the system. For any isolator $Is \in \mathbf{IS}_{tot}$,

$$\overline{Is} = (Q^{Is}, \Sigma^{Is}, T^{Is}, D^{Is}, \lambda^{Is}, q_0^{Is})$$

will be the FSA model of Is. We have

$$Q^{Is} = \{ZERO, ONE\}$$

$D^{Is} = \{0, 1\}$ is the set of discrete output symbols with $\lambda^{Is}(ZERO) = 0$ and $\lambda^{Is}(ONE) = 1$. The FSA $\overline{Is}$ stays in the state ZERO as long as the output of Is is zero. There will be a transition from ZERO to ONE when the output of Is becomes one. Figure 3.9 shows the FSA model of the isolator Is. In Figure 3.9, the events '$Is : 0 \to 1$' and '$Is : 1 \to 0$' label the transitions from the state ZERO to ONE and from the state ONE to ZERO, respectively. The unobservable events '$Is : 0$' and '$Is : 1$' are fictitious events added for design consistency (to be discussed subsequently). These self-loop transitions (transitions from one discrete state to itself) are unobservable and do not change the output. We have

$$\Sigma^{Is} = \{Is : 0 \to 1, Is : 1 \to 0, Is : 0, Is : 1\}$$

Let $\Sigma^{\mathbf{IS}_{tot}}$ denote the set of all events of the isolators, that is

$$\Sigma^{\mathbf{IS}_{tot}} = \bigcup_{Is \in \mathbf{IS}_{tot}} \Sigma^{Is}$$

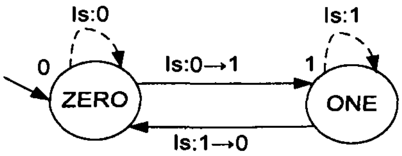


Figure 3.9: The FSA modeling the isolator Is.

3.2.3 Consistency Between the System and the Isolator DES Models

Let $Q^{EM} \subseteq Q$ be a set of EM-similar discrete states. We are developing DES models for the system and isolators. These models must capture the interactions between the system and each isolator. These interactions, as explained in the following, correspond to changes in the output of isolators in response to changes in the system such as mode changes and occurrences of faults.

While the system is evolving in any $q \in Q^{EM}$, the isolators designed for the discrete states $q \in Q - Q^{EM}$ can only have transitions from ZERO to ONE or stay at ONE (if already at ONE). Assume that the system enters the discrete state $q \in Q^{EM}$. While the system is in q, an isolator $Is^{Q^{EM}}(\Phi)$ has a transition from ONE to ZERO or stays at ZERO if $q \notin active^{-1}(\Phi) \bigcap Q^{EM}$. Otherwise, it has a transition from ZERO to ONE or stays at ONE.

We modify the DES model of the system H_{abs} to enforce the above-mentioned consistency requirements. Let $\hat{H}_{abs} = (Q, \hat{\Sigma}, \hat{T}, D, \lambda, q_0)$ be the new FSA. The FSA $\hat{H}_{abs}$ is constructed by modifying the DES abstraction of H, H_{abs}, by adding appropriate self-loop transitions to H_{abs}. In $\hat{H}_{abs}$,we have: $\hat{\Sigma} = \Sigma \bigcup \Sigma^{\mathbf{IS}_{tot}}$;

$$\hat{T} = T \bigcup T_1 \bigcup T_2 \bigcup T_3 \bigcup T_4$$

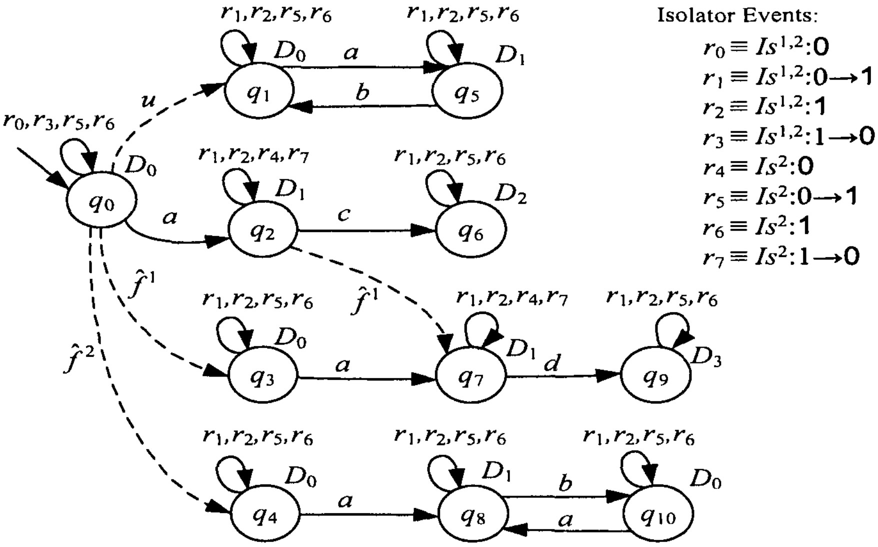


Figure 3.10: Example 3.2.1: The modified DES abstraction of system ($\hat{H}_{abs}$).

where

$$T_1 = \{(q, Is^{Q^{EM}}(\Phi) : 0 \to 1, q), \mid q \in (Q - Q^{EM}) \bigcup (active^{-1}(\Phi) \bigcap Q^{EM})\}$$

$$T_2 = \{(q, Is^{Q^{EM}}(\Phi) : 1, q), \mid q \in (Q - Q^{EM}) \bigcup (active^{-1}(\Phi) \bigcap Q^{EM})\}$$

$$T_3 = \{(q, Is^{Q^{EM}}(\Phi) : 1 \to 0, q), \mid q \in (Q^{EM} - (active^{-1}(\Phi) \bigcap Q^{EM}))\}$$

$$T_4 = \{(q, Is^{Q^{EM}}(\Phi) : 0, q), \mid q \in (Q^{EM} - (active^{-1}(\Phi) \bigcap Q^{EM}))\}$$

Example 3.2.2. *(Example 3.2.1 Continued): Assume that we are only able to design the following two isolators for the hybrid system shown in Figure 3.8: the isolator $Is(\{f^1, f^2\})$ (denoted as $Is^{1,2}$) for $Q^{E^1M^1}$ and the isolator $Is(\{f^2\})$ (denoted as Is^2) for $Q^{E^2M^2}$. The FSA $\hat{H}_{abs}$ satisfying the consistency specification is shown in Figure 3.10. The FSA $\hat{H}_{abs}$ is constructed from H_{abs} by adding certain selfloop transitions at each discrete state. For instance, at the discrete state q_0, the events $Is^{1,2} : 0$, $Is^{1,2} : 1 \to 0$, $Is^2 : 1$ and $Is^2 : 0 \to 1$ label the selfloop transitions. This implies that the isolator $Is^{1,2}$ cannot have a transition labeled with $Is^{1,2} : 1$ or $Is^{1,2} : 0 \to 1$ while the system is in q_0.* ■

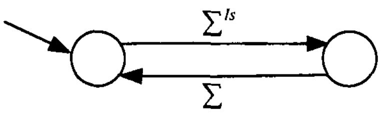


Figure 3.11: The FSA ASM_{Is}.

3.2.4 Enforcing the Assumptions by DES Models

As mentioned earlier in this section, we assume that the system stays in each discrete state long enough (τ^{min}) so that the transient response due to the mismatch in the initial conditions of isolators and the system dies out. Hence, the time between the occurrence of two consecutive events is sufficient to evaluate and use the output of the isolators for diagnosis. This assumption is captured at the DES level as follows: one event of every isolator must occur between the occurrence of any two consecutive events in the system. We also assume that before the first event in the system occurs, the isolators reach their steady state. The FSA ASM_{Is} shown in Figure 3.11 enforces these assumptions for an isolator Is. The transition labeled with Σ implies that any event in the event set of the system can label that transition.

3.2.5 Constructing the EDESA of the Hybrid Automata and Isolators

Assume that there are b ($b \geq 0$) isolators designed for the system, and let $\overline{\mathbf{IS}}_{tot} = \{\overline{Is}_1, \cdots, \overline{Is}_b\}$ be the set of FSA modeling the isolators. The EDESA of the hybrid system and isolators, denoted as $\tilde{H}$, is an FSA defined as

$$\tilde{H} = (\tilde{Q}, \tilde{\Sigma}, \tilde{T}, \tilde{D}, \tilde{\lambda}, \tilde{q}_0)$$

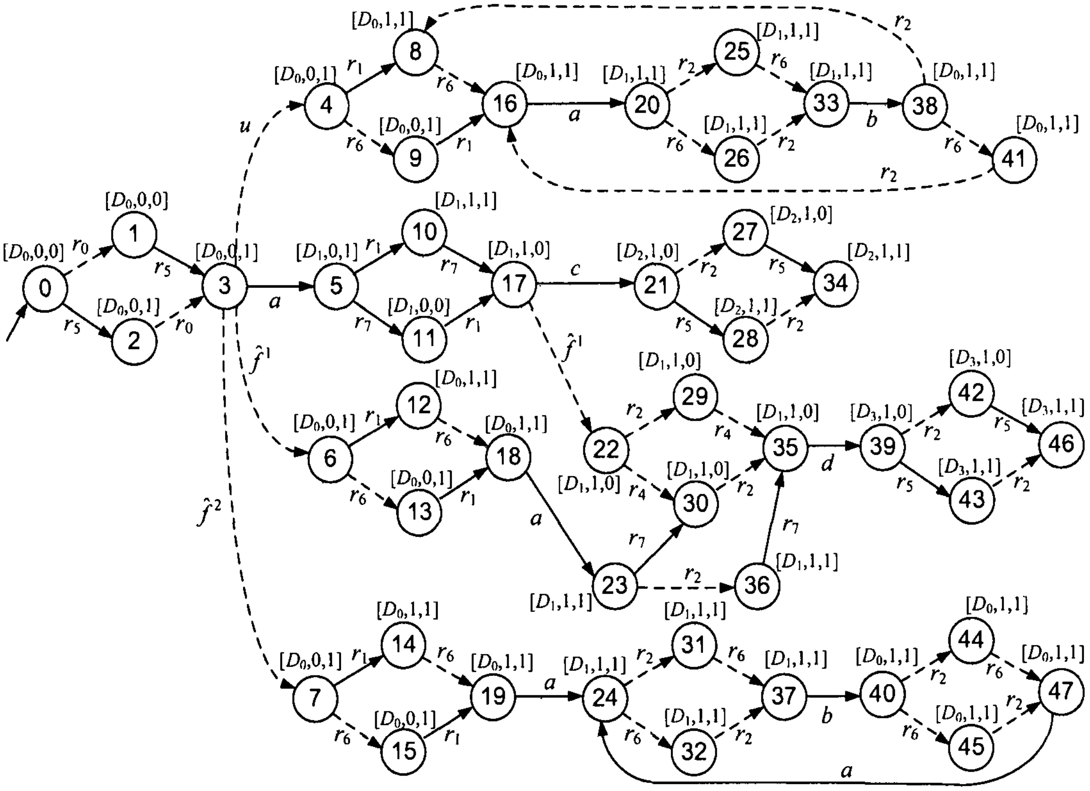


Figure 3.12: The EDESA of the system and isolators.

and is constructed by combining $\hat{H}_{abs}$, the FSA modeling the isolators and the FSA enforcing the assumptions $ASM_{Is_1}, \cdots, ASM_{Is_b}$:

$$\tilde{H} = \textbf{sync}(\hat{H}_{abs}, \overline{Is}, ASM)$$

where

$$\overline{Is} = \textbf{sync}(\overline{Is}_1, \cdots, \overline{Is}_b)$$

and

$$ASM = \textbf{sync}(ASM_{Is_1}, \cdots, ASM_{Is_b})$$

Example 3.2.3. *(Example 3.2.1 Continued): Figure 3.12 shows the EDESA of the hybrid automaton as shown in Figure 3.8 and the isolators* $\overline{Is}(\{f^1, f^2\})$ *and*

$\overline{Is}(\{f^2\})$. The discrete output of each discrete state of the EDESA is an array with three elements. The first element is coming from the discrete output of the system; the second element is the discrete output of $\overline{Is}(\{f^1, f^2\})$ and the third is the discrete output of $\overline{Is}(\{f^2\})$. In the following, we describe the operation of the diagnoser designed for the EDESA shown in Figure 3.12 for the event sequence 'ad' generated in the system.

Assume that the initial state estimate of the diagnoser designed for the EDESA is $z_0 = \tilde{Q}$. Therefore, the initial condition estimate provided by the diagnoser is $\kappa(z_0) = \{N, F^1, F^2\}$. Suppose when the diagnoser starts, the discrete output '$[D_0 \ 1 \ 1]$' is generated by the EDESA. The state estimate provided by the diagnoser is updated to $z_1 = \{8, 12, 14, 16, 18, 19\}$. The condition estimate is $\kappa(z_1) = \{N, F^1, F^2\}$. Now assume that the event 'a' is generated in the system and the discrete output '$[D_1 \ 1 \ 1]$' is observed in the EDESA. The state estimate provided by the diagnoser is updated to $z_2 = \{20, 23, 24\}$, and the condition estimate is $\kappa(z_2) = \{N, F^1, F^2\}$. Suppose after the isolators reach their steady state, we observe '$[D_1 \ 1 \ 0]$' in the EDESA. The state estimate provided by the diagnoser is updated to $z_3 = \{30, 35\}$ and the condition estimate is updated to $\kappa(z_3) = \{F^1\}$. This implies that the failure mode F^1 has occurred in the system. Now assume that the event 'd' is generated in the system and the discrete output '$[D_3 \ 1 \ 0]$' is observed in the EDESA. The state estimate provided by the diagnoser is updated to $z_4 = \{39\}$ and the condition estimate is $\kappa(z_4) = \{F^1\}$.

Figure 3.13 shows a part of the diagnoser designed for the EDESA corresponding to the above-mentioned output sequence. The failure mode F^1 can be detected by using only the discrete outputs generated in the system when the F^1-indicative discrete output 'D_3' is observed. However, as described above, by using the information provided by the isolators, F^1 is detected before discrete output D_3 is generated. ■

Remark 3.2.1. *In the previous discussion, it was assumed that the isolators provide*

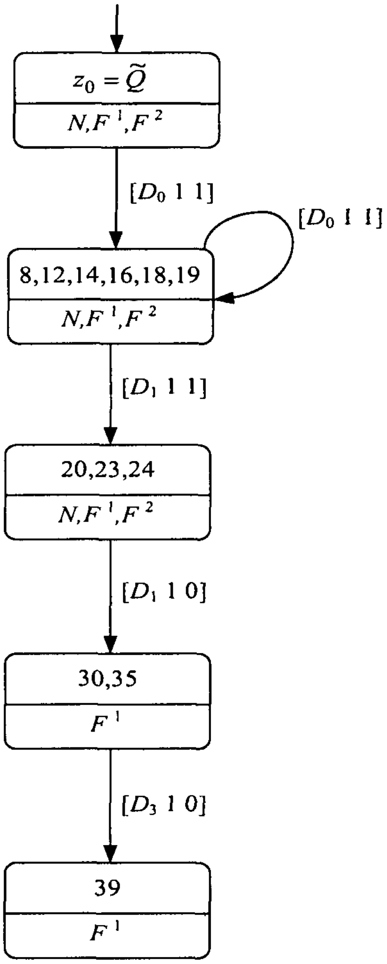


Figure 3.13: A part of the diagnoser constructed for the EDESA shown in Figure 3.12.

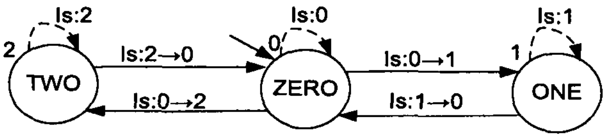


Figure 3.14: The FSA modeling an isolator with a three-level residual.

information to help with the determination of fault types. Next, this information is integrated with the information coming from the discrete sensors by the hybrid diagnoser to determine the failure mode. In general, the residual signals can be further processed to assist in the determination of failure modes. In this case, the output of the signal processing unit of each isolator would be a multi-level signal, say, zero to n, with each level corresponding to one failure mode. The construction of DES models for isolators and an EDESA for the integrated model of system and isolators is similar to what has been previously discussed in this chapter, and is omitted for brevity. Here, as an example, we have provided an FSA model for an isolator with three levels of output (Figure 3.14).

In the next chapter, we discuss diagnosability of failure modes in our framework.

3.3 Summary

In this chapter, we developed an approach for fault diagnosis in systems that are modeled by hybrid automata. Generally, the information available for diagnosis comes from the discrete outputs and continuous output and control input signals. In this approach, we developed a systematic method to integrate the information at both DES and continuous levels and use it for diagnostic purposes. In our framework, all faults are modeled at the continuous level as unknown input signals. Every fault type signal corresponds to a set of failure modes in a component. We used a bank of isolators (residual generators) for detection and isolation of fault types at the continuous dynamics. We developed a systematic approach for modeling the isolators by DES models and integrating these DES models with the DES abstraction of the system to construct an extended DES model. We use the extended DES model for diagnosis and diagnosability analysis.

Chapter 4

DIAGNOSABILITY OF FAILURE MODES IN HYBRID AUTOMATA

In this chapter, we investigate diagnosability of failure modes in hybrid automata. We introduce a notion of diagnosability in hybrid automata and provide sufficient conditions under which a failure become diagnosable in hybrid automata. Furthermore, we study diagnosability of failure modes in the case that a number of isolators produce unreliable outputs (false alarms and fault silences).

4.1 Diagnosability of Failure Modes

Consider a hybrid automaton

$$H = (Q, \mathcal{X}, \mathcal{U}, FT, \mathcal{Y}, Init, S, \Sigma, T, G, \rho, D, \lambda, q_0)$$

with DES abstraction

$$H_{abs} = (Q, \Sigma, T, D, \lambda, q_0)$$

Let $\mathcal{F}$ be the set of failure modes. Also let $\mathbf{IS}_{tot}$ be the set of isolators and

$$\tilde{H} = (\tilde{Q}, \tilde{\Sigma}, \tilde{T}, \tilde{D}, \tilde{\lambda}, \tilde{q}_0)$$

be the EDESA of the system and isolators. We intend to investigate the diagnosability of the failure modes in H. We will show that a failure mode is diagnosable in H, if it is diagnosable in the EDESA of H and the isolators designed for H. First, we discuss diagnosability of a failure mode in the EDESA.

In Chapter 2, we discussed the diagnosability of failure modes in DES. We also reviewed the necessary and sufficient conditions for the diagnosability of a failure mode in Theorem 2.3.1. Diagnosability results discussed in Chapter 2 for DES do not consider time. In other words, it is assumed that at any state, one of transitions defined at that state will definitely occur. In a hybrid automaton, the system may become stuck in some discrete states forever even if a transition defined at that discrete state exists. Therefore, the assumption that one of the transitions defined at a discrete state will eventually occur may not hold. Therefore, before studying the diagnosability of EDESA and relating it to the diagnosability of H, the above issue needs to be addressed.

We assume that the discrete states can be partitioned into two sets:

$$Q = Q^{inf} \dot{\bigcup} Q^{fin}$$

where Q^{fin} is the set of discrete states in which the system never stays longer than a finite time, say τ^{max}, and Q^{inf} is the set of discrete states in which the system can remain indefinitely. Similarly, partition $\tilde{Q}$ as:

$$\tilde{Q} = \tilde{Q}^{inf} \dot{\bigcup} \tilde{Q}^{fin}$$

The value of τ^{max} can be calculated based on the type of events labeling the outgoing transitions, and the dynamics of the system at the discrete states and the guard conditions. We assume that τ^{max} is given. In the following, we briefly address how it is determined whether a state $\tilde{q} \in \tilde{Q}^{inf}$ or $\tilde{q} \in \tilde{Q}^{fin}$, and discuss the ways τ^{max} may be calculated.

As explained in Chapter 3, there are two types of transitions in a system: autonomous transitions whose occurrence depends on the values of the continuous state and the control input of the system, and transitions that have a pure DES nature such as supervisor commands. All transitions in our framework are labeled with events. We assume that the events labeling the transitions can be partitioned into *prospective* events and *remote* events[1]. Prospective events are those events that will occur before an upper time bound unless they are preempted by another event. Remote events are those that can take an arbitrarily long time to occur or may never occur even if eligible (enabled) to occur. For example, supervisor commands are prospective events and fault events are remote events.

The events labeling the autonomous transitions can be prospective events or remote events. For example, in the gas turbine engine described in Chapter 6, the event "*max*2*ab*" labeling the autonomous transition that takes the engine to the operating regime "Afterburner on" is a remote event because the engine may never go to this operating regime, but the event "*ab*2*max*" labeling the autonomous transition that takes the engine from the operating regime "Afterburner on" to the operating regime "Max power" is a prospective event because the engine cannot stay in the operating regime "Afterburner on" for ever due to the fuel limitation. However, the event "*ab*2*max*" can be preempted by the event "*ab*2*maxab*" that takes the engine to the operating regime "Max afterburner thrust".

If there is no outgoing transition at q labeled with a supervisory command and

[1] Here, we have adopted the terminology used in [21] for timed DES.

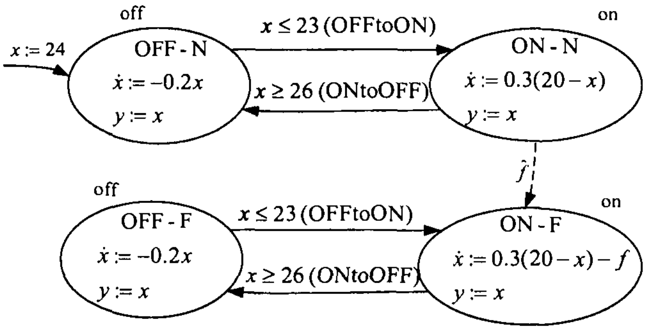


Figure 4.1: A hybrid automaton modeling a heating system with a fault.

some of the outgoing transitions at q are autonomous, by using analytical methods or by simulation, it can be inferred whether $q \in Q^{inf}$ or $q \in Q^{fin}$. In order to use analytical methods or simulation, we assume (as often is the case) that when a guard condition of a transition becomes true, the system cannot stay any longer in the source state of that transition and the transition will occur unless it is preempted by another transition. For example, in the heating system example in Chapter 3 (shown in Figure 4.1), when the temperature of the room reaches 23^o, the transition from $OFF - N$ to $ON - N$ takes place, and when the temperature reaches 26^o the transition from $ON - N$ to $OFF - N$ occurs. Therefore, verifying if a discrete state q belongs to Q^{fin} can be investigated analytically by reachability analysis in hybrid automata, even though it may not be simple. For example, if the region of states specified by the domain of the guard conditions of q can be reached from the initial state of the system, then $q \in Q^{fin}$.

Let τ_q^{max} be the maximum time that the system may stay in q. Thus, τ^{max} will be the maximum of τ_q^{max} for all $q \in Q^{fin}$. If $q \in Q^{fin}$, and there is no outgoing transition at q labeled with a supervisory command, and some of the outgoing transitions at q are autonomous, then τ_q^{max} depends on the continuous dynamics at q and may be computed analytically by solving the differential equations of the system. For instance, in the heating system example shown in Figure 3.3, assume that the initial temperature is between 23^0 and 30^0, i.e., $23^0 \leq x(0) \leq 30^0$. When the system starts, the discrete state is $OFF - N$. There is only one outgoing autonomous transition labeled with "$OFFtoON$" defined at $OFF - N$. This event occurs if $x(t) = 23^0$. The event "$OFFtoON$" is a prospective event because there exists a time t_0 such that $x(t_0) = 23^0$. Therefore, $OFF - N \in Q^{fin}$. The differential equation in the discrete state $OFF - N$ is

$$\dot{x}(t) = -0.2x(t)$$

and the solution of the differential equation in $OFF - N$ is

$$x(t) = C_1 e^{-\frac{1}{5}t}$$

where C_1 is a constant and can be calculated based on the initial condition. The event "$OFFtoON$" will occur at $OFF - N$ if $x(t) = 23^o$. Assume that the event "$OFFtoON$" occurs at $t = t_0$. If $x(0) = 23^o$, then $t_0 = 0$ *sec.*, and if $x(0) = 30^o$, then $t_0 = 1.33$ *sec.* If $23^0 \leq x(0) \leq 30^0$, then $0 \leq t_0 \leq 1.33$ *sec.* Therefore, τ^{max}_{OFF-N} can be taken as 1.33 *sec.* or any finite value greater than 1.33 *sec.* The system transitions to the discrete state $ON - N$ at $t = t_0$. There are two outgoing transitions defined at $ON - N$: autonomous transition labeled with "$ONtoOFF$" and the remote transition "$\hat{f}$". The event "$ONtoOFF$" occurs if $x(t) = 26^0$. The event "$ONtoOFF$" is a prospective event because if "$ONtoOFF$" is not preempted by $\hat{f}$, there exists a time t_1 such that $x(t_1) = 26^o$. Therefore, $ON - N \in Q^{fin}$. Let t_1 be the time that "$ONtoOFF$" occurs. Initially, assume that the fault does not occur when the system is in $ON - N$. The differential equation in $ON - N$ is

$$\dot{x}(t - t_0) = -0.3(20 - x(t - t_0))$$

the initial state in $ON - N$ is $x(t_0) = 23^o$ and the solution of the differential equation in $ON - N$ is

$$x(t - t_0) = 20 + 3e^{\frac{3}{10}(t-t_0)}$$

The event "$ONtoOFF$" occurs at $t_1 = t_0 + 2.31$ *sec.* Therefore, τ^{max}_{ON-N} can be taken 2.31 *sec.* or any finite value greater than 2.31 *sec.* Now assume that the fault event $\hat{f}$ occurs at $t = t'_0 \geq t_0$ and the system is taken to the discrete state $ON - F$. Also assume that $0.25 \leq f(t) \leq 0.5$ for $t'_0 \leq t \leq t_1$. It can be verified that there exists a time t_2 such that $x(t_2) = 26^o$. Hence, "$ONtoOFF$" is a prospective event at $ON - F$, and $ON - F \in Q^{fin}$. Moreover, it can be verified that $t_2 <= t'_0 + 3.93$ *sec.* Therefore, τ^{max}_{ON-F} can be taken 3.93 *sec.* or any finite value greater than 3.93 *sec.* It can be also verified that $OFF - F \in Q^{fin}$, and τ^{max}_{OFF-F} can be taken 1.33 *sec.* or any finite value greater than 1.33 *sec.* Thus, τ^{max} for the heating system can be taken 3.93 *sec.* or any finite value greater than 3.93 *sec.*

The reachability analysis in hybrid automata is an open area of research (e.g., see [76]). Let Rg_q denote the region of continuous states specified by the guard conditions of q. Verifying the reachability of Rg_q may not always be possible. There are also approaches that investigate reachability in hybrid automata by approximation, e.g., see [8, 12, 67]. If the reachability of Rg_q is not easy to guarantee, the reachability of a region inside Rg_q may be investigated and a more conservative value may be found for τ^{max}_q.

If all the outgoing transitions at q are labeled with remote events, the system can stay in q indefinitely and $q \in Q^{inf}$.

Next, we discuss how to determine whether $\tilde{q} \in \tilde{Q}^{inf}$ or not. The events of the FSA modeling the isolators are all prospective. As explained in Chapter 3, each state of the EDESA of the system with b isolators is a $(b + 1)$-tuple in which the first component is a state of the DES abstraction of the hybrid system and the rest of the components comes from the FSA modeling the isolators. Let $\tilde{q} \in \tilde{Q}$ and q be the component of $\tilde{q}$ corresponding to the DES abstraction of the hybrid system. We have $\tilde{q} \in \tilde{Q}^{fin}$ if and only if $q \in Q^{fin}$ and there is no outgoing transition defined at $\tilde{q}$ labeled with the events of the isolators.

Remark 4.1.1. *Computing τ^{max} may be useful in online diagnosis for reducing the size of the discrete state estimate provided by the diagnoser. For example, assume that at time $t = t_0$, the discrete state estimate provided by the diagnoser is $z_n = \{q_1, q_2\}$ with $q_1 \in Q^{inf}$ and $q_2 \in Q^{fin}$. If there is no transition by $t = t_0 + \tau^{max}$, it can be concluded that the system is in a discrete state of Q^{inf}. Therefore, z_n can be updated to $z_n = \{q_2\}$.* ■

Remark 4.1.2. *As explained earlier, calculating an exact value for τ^{max} using analytical methods depends on the dynamics of the hybrid system and may not be easy. Therefore, calculating τ^{max} based on simulation and approximation methods is more common, however, the obtained result may not be correct. As a result, the system may stay in a discrete state $q \in Q^{fin}$ for a time greater than τ^{max}. For example, suppose at time $t = t_0$, the discrete state estimate provided by the diagnoser is $z_n = \{q_1, q_2, q_3\}$. Assume that $q_1, q_2, q_3 \in Q^{fin}$, but no transition occurs by $t = t_0 + \tau^{max}$. There may be different strategies to deal with this situation. One approach is to give different weights to different failure modes, and then, make a decision based on the criticality of the condition estimate. For example, if $\kappa(z)$ contains a condition which is very critical and may result in disaster in the system, one may take the system to a safe mode. In this case, some failure modes may remain undiagnosed.* ■

In order to be able to use the results of Theorem 2.3.1 in our work to study the diagnosability of EDESA, we add an unobservable self-loop transition labeled with a fictitious event to every discrete state of Q^{inf} which has an outgoing transition to another state. By doing this, we allow transitions to occur at these discrete states without changing the discrete state of the system and thus model EDESA getting stuck in a state. Therefore, for the diagnosability of a failure mode in $\tilde{H}$, we can use the results of Theorem 2.3.1. For instance, in Example 3.2.1, F^1 is diagnosable but F^2 is not diagnosable because it violates condition (3) of Theorem 2.3.1.

For $F \in \mathcal{F}$, define $\tilde{H}_F$ as the sub-generator of $\tilde{H}$ consisting of the states of $\tilde{Q}_F$ only. Similarly, define $\tilde{H}_N$, $\tilde{H}_{\mathcal{F}}$, $\tilde{H}_{\mathcal{F}^i}$ and $\tilde{H}_{N,\overline{\mathcal{F}_i}}$ as the sub-generators of $\tilde{H}$ corresponding to the states of $\tilde{Q}_N$, $\tilde{Q}_{\mathcal{F}}$, $\tilde{Q}_{\mathcal{F}^i}$ and $\tilde{Q}_N \bigcup \tilde{Q}_{\overline{\mathcal{F}^i}}$.

Now we extend the definition of output language described in Chapter 2. The output language $L_o(\tilde{H}, \tilde{q}, \tilde{q}')$ generated by $\tilde{H}$ from the state $\tilde{q} \in \tilde{Q}$ to state $\tilde{q}' \in \tilde{Q}$ is defined as follows: If $\tilde{q} \neq \tilde{q}'$, $L_o(\tilde{H}, \tilde{q}, \tilde{q}') := \{\tilde{d}_1\tilde{d}_2 \cdots \tilde{d}_m$ such that $\tilde{d}_i \in \tilde{D}$ for $(1 \leq i \leq m), \tilde{d}_1 = \tilde{\lambda}(\tilde{q}), \tilde{d}_m = \tilde{\lambda}(\tilde{q}')$ and [there exists $\tilde{q}_i \in \tilde{Q}$ $(1 \leq i \leq m)$: $\tilde{q}_1 = \tilde{q}, \tilde{q}_{i-1} \Rightarrow \tilde{q}_i, \tilde{q}_m = \tilde{q}', \tilde{d}_i = \tilde{\lambda}(\tilde{q}_i), (2 \leq i \leq m)]\}$. If $\tilde{q} = \tilde{q}'$, $L_o(\tilde{H}, \tilde{q}, \tilde{q}') := \{\tilde{\lambda}(\tilde{q})\} \bigcup \{\tilde{d}_1\tilde{d}_2 \cdots \tilde{d}_m$

such that $\tilde{d}_i \in \tilde{D}$ for $(1 \leq i \leq m), \tilde{d}_1 = \tilde{\lambda}(\tilde{q}), \tilde{d}_m = \tilde{\lambda}(\tilde{q}')$ and [there exists $\tilde{q}_i \in \tilde{Q}$ $(1 \leq i \leq m)$: $\tilde{q}_1 = \tilde{q}, \tilde{q}_{i-1} \Rightarrow \tilde{q}_i, \tilde{q}_m = \tilde{q}', \tilde{d}_i = \tilde{\lambda}(\tilde{q}_i), (2 \leq i \leq m)]\}$.

$L_o(\tilde{H}_{\mathcal{F}^i}, \tilde{q}, \tilde{q}')$, $L_o(\tilde{H}_{N,\overline{\mathcal{F}_i}}, \tilde{q}, \tilde{q}')$, $L_o(\tilde{H}_N, \tilde{q}, \tilde{q}')$ and $L_o(\tilde{H}_{\mathcal{F}}, \tilde{q}, \tilde{q}')$ are defined similarly.

Proposition 4.1.1. *Let b be the number of isolators designed for the system. Also let $|Q|$ be the cardinality of Q. The complexity of verifying the diagnosability of the failure mode F^i in $\tilde{H}$ is $\mathcal{O}(2^{4b}|Q|^4)$.*

Proof - As explained in Chapter 2, the complexity of verifying the diagnosability of the failure mode F^i in an FSA with the state set Q is $\mathcal{O}(|Q|^4)$ [90]. Since each FSA modeling an isolator has two discrete states, and the synchronous product of H_{abs} and isolators are used for constructing $\tilde{H}$, the size of the state set $\tilde{Q}$ has the order of $\mathcal{O}(|2^bQ|)$. Therefore, the complexity of verifying the diagnosability of the failure mode F^i in a $\tilde{H}$ is $\mathcal{O}(|2^bQ|^4) = \mathcal{O}(2^{4b}|Q|^4)$. ■

The complexity of verifying the diagnosability of the failure mode F^i in $\tilde{H}$ is exponential in the number of isolators, but polynomial in the number of discrete states of H.

In [114], it is shown that the occurrence of a diagnosable failure mode F^i can be detected within $|\tilde{Q}|^2$ discrete transitions after the fault occurs. Now, we investigate the diagnosability of a failure mode in the EDESA of the system and isolators with a constraint Π on the number of events.

Definition 4.1.1. *We call a failure mode F^i **Π-diagnosable** if it is always possible to detect after at most a bounded number of events Π generated in the system (following the occurrence of the failure mode and initialization of the diagnoser) whether the system has entered and stayed in the set $Q_{\mathcal{F}^i}$.* ■

Theorem 4.1.2. *Let $b = |\mathbf{IS}_{tot}|$ be the number of isolators in the system, and assume that $z_0 = \tilde{Q}$. A permanent failure mode F^i is Π-diagnosable if and only if:*

1. *For any $\tilde{q} \in \tilde{Q}_{\mathcal{F}^i}$, if $\tilde{q} \in \tilde{Q}^{inf}$ then $\tilde{\lambda}^{-1}(\tilde{\lambda}(\tilde{q})) \bigcap (\tilde{Q} - \tilde{Q}_{\mathcal{F}^i}) = \emptyset$;*

2. *For any $\tilde{q} \in \tilde{Q}_{\mathcal{F}^i}$ and $\tilde{q}' \in (\tilde{Q}_N \bigcup \tilde{Q}_{\overline{\mathcal{F}_i}})$ satisfying $\tilde{\lambda}(\tilde{q}) = \tilde{\lambda}(\tilde{q}')$, we have:*

$$\{s \mid s \in L(\tilde{H}_{\overline{\mathcal{F}_i}}, \tilde{q}),\ \tilde{\lambda}(s) \in L_o(\tilde{H}_{N,\overline{\mathcal{F}_i}}, \tilde{q}') \bigcap L_o(\tilde{H}_{\mathcal{F}^i}, \tilde{q}), |s| \geq ((b+1)\Pi + b)\} = \emptyset$$

Proof - Condition (1) states that there should be no deadlock states in $\tilde{Q}_{\mathcal{F}^i}$ with no transition out of the state unless the output in that state can be generated only when F^i has occurred. In [48], such an output is called **F^i-indicative**. Moreover, the system should not stay in any discrete state for ever unless the output of that state is F^i-indicative. We assumed that all isolators generate an event between two consecutive events generated in the system. Therefore, before $\Pi + 1$ events are generated in the system, there will be $b\Pi + b$ events generated by the isolators. Thus, before the system generates $(\Pi + 1)$th event after the occurrence of F^i, the number of events generated in $\tilde{H}$ will be $(b+1)\Pi + b$. Condition (2) states that there should be no sequence of events with a length greater than $(b+1)\Pi + b$ in $\tilde{H}$ that generates common output sequence in $\tilde{Q}_{\mathcal{F}^i}$ and $\tilde{Q} - \tilde{Q}_{\mathcal{F}^i} = \tilde{Q}_N \bigcup \tilde{Q}_{\overline{\mathcal{F}^i}}$ (otherwise, F^i cannot be distinguished before the $(\Pi + 1)$th event generated in the system after the occurrence of F^i and hence F^i will not be Π-diagnosable).

(If part) Assume that condition (1) and condition (2) hold. Following the occurrence of the failure mode and the initiation of diagnosis, if the system generates upto Π events and then stops generating new outputs, then by condition (1), F^i will be diagnosed. If the system generates $\Pi + 1$ events or more, before the system generates the $(\Pi + 1)$th event, the discrete output sequence generated by $\tilde{H}$ in $\tilde{Q}_{\mathcal{F}^i}$ will be different from the discrete output sequence generated by $\tilde{H}$ in $\tilde{Q} - \tilde{Q}_{\mathcal{F}^i}$. Thus, F^i will be Π-diagnosable.

(only if part) Condition (1) is necessary because if condition (1) does not hold, there will be $\tilde{q} \in \tilde{Q}_{\mathcal{F}^i} \bigcap \tilde{Q}^{inf}$ and $\tilde{q}' \in (\tilde{Q} - \tilde{Q}_{\mathcal{F}^i})$ such that $\tilde{\lambda}(\tilde{q}) = \tilde{\lambda}(\tilde{q}')$. If the system is in $\tilde{q}$, it may not generate any new event. If the diagnosis is started after system enters $\tilde{q}$, F^i will not be diagnosed. Condition (2) is necessary because if condition (2) does not hold, there will be $\tilde{q} \in \tilde{Q}_{\mathcal{F}^i}$ and $\tilde{q}' \in (\tilde{Q}_N \bigcup \tilde{Q}_{\overline{\mathcal{F}_i}})$ and a sequence of events s generated in $\tilde{H}$ such that $|s| \geq ((b + 1)\Pi + b)$ and the sequence of outputs generated by $\tilde{H}$ for s belongs to $L_o(\tilde{H}_{N,\overline{\mathcal{F}_i}}, \tilde{q}') \bigcap L_o(\tilde{H}_{\mathcal{F}^i}, \tilde{q})$. Therefore, F^i will not be distinguished in $\tilde{H}$ before the system generates $(\Pi + 1)$th event after the occurrence of F^i. Thus, F^i will not be Π-diagnosable. ■

Example 4.1.1. *Figure 4.2 shows the DES abstraction of a hybrid automaton H with two fault types f^1 and f^2 and corresponding failure modes F^1 and F^2, respectively. The occurrence of the failure modes F^1 and F^2 are modeled by transitions labeled with the events $\hat{f}^1$ and $\hat{f}^2$, respectively. The unobservable event 'u_1' changes the discrete state of the system in normal mode. The labels 'u_2' and 'u_3' are unobservable event changing the discrete state of the system in the failure mode F^2 and form a cycle of states with the same discrete output. The set of discrete states is $Q = \{q_0, \cdots, q_{13}\}$ and $Q^{inf} = \{q_8, q_9, q_{13}, q_{14}\}$. Since there is an outgoing transition*

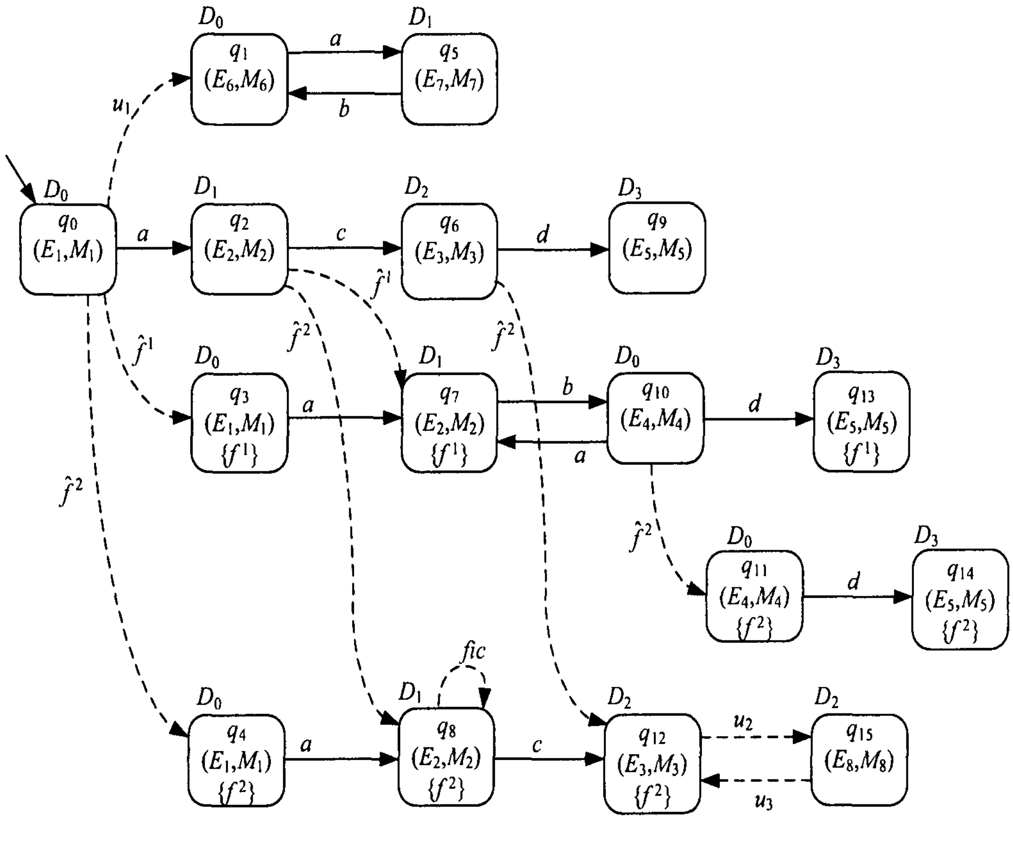


Figure 4.2: DES abstraction of a hybrid automaton with two failure modes.

at q_8, a fictitious unobservable selfoop event is added at q_8. The condition set of the system is $\mathcal{K} = \{N, F^1, F^2, F^{1,2}\}$. The discrete states of the system can be partitioned according to the condition of the system. Here, $Q_N = \{q_0, q_1, q_2, q_5, q_6, q_9\}$, $Q_{F^1} = \{q_3, q_7, q_{10}, q_{13}\}$, $Q_{F^2} = \{q_4, q_8, q_{12}, q_{15}\}$ and $Q_{F^{1,2}} = \{q_{11}, q_{14}\}$. We also have $Q_{\mathcal{F}^1} = \{q_3, q_7, q_{10}, q_{11}, q_{13}, q_{14}\}$ and $Q_{\mathcal{F}^2} = \{q_4, q_8, q_{11}, q_{12}, q_{14}, q_{15}\}$. Moreover, we have $Q^{E_1M_1} = \{q_0, q_3, q_4\}$, $Q^{E_2M_2} = \{q_2, q_7, q_8\}$, $Q^{E_3M_3} = \{q_6, q_{12}\}$, $Q^{E_4M_4} = \{q_{10}, q_{11}\}$ and $Q^{E_5M_5} = \{q_9, q_{13}, q_{14}\}$. The rest of the discrete states have different (E, M) functions. Besides, $active^{-1}(\{f^1\}) = \{q_3, q_7, q_{13}\}$ and $active^{-1}(\{f^2\}) = \{q_4, q_8, q_{11}, q_{12}, q_{14}\}$. The set of discrete outputs of the system is: $D = \{D_0, D_1, D_2, D_3\}$.

Let z_0 be the initial state of the diagnoser designed for H_{abs} and assume that $z_0 = Q$. None of the failure modes are diagnosable using the diagnoser designed for H_{abs}. Failure mode F^1 is not diagnosable because the cycle of discrete states q_7 and q_{10} generate the similar output sequence as the cycle of discrete states q_1 and q_5. Therefore, condition (3) of Theorem 2.3.1 in Chapter 2 is violated. Furthermore, q_{13} and q_{14} belong to Q^{inf} but have the same discrete output as q_9. Thus, condition (1) of the theorem is violated. Failure mode F^2 violates condition (1) of that theorem because $q_{14} \in Q^{inf}$ but it cannot be distinguished from q_9 and q_{13} by using the discrete outputs generated in the system. Failure mode F^2 violates condition (2) of that theorem too because cycle q_8 and cycle q_{12} and q_{15} do not have F^2-indicative outputs.

Let the set of isolators that can be designed for the system be: $\mathbf{IS}_{tot} = \{Is^{Q^{E_2M_2}}(\{f^1\}), Is^{Q^{E_2M_2}}(\{f^1, f^2\}), Is^{Q^{E_2M_2}}(\{f^2\}), Is^{Q^{E_3M_3}}(\{f^2\}), Is^{Q^{E_5M_5}}(\{f^1\}), Is^{Q^{E_5M_5}}(\{f^2\}), Is^{Q^{E_8M_8}}(\emptyset)\}$. Here, $Is^{Q^{E_8M_8}}(\emptyset)$ is an E_8M_8-distinguisher. We can verify that both failure modes are diagnosable in $\tilde{H}$. Failure mode F^1 becomes diagnosable in $\tilde{H}$ because the outputs produced by the isolators $Is^{Q^{E_2M_2}}(\{f^1\})$, $Is^{Q^{E_2M_2}}(\{f^1, f^2\})$ and $Is^{Q^{E_2M_2}}(\{f^2\})$ when the system is in cycle q_7 and q_{10} are different from the outputs produced when the system is in cycle q_1 and q_5. In fact, q_7 can *be distinguished from q_5. Therefore, condition (3) of Theorem 2.3.1 is satisfied for the failure mode F^1. Also the isolators $Is^{Q^{E_5M_5}}(\{f^1\})$ and $Is^{Q^{E_5M_5}}(\{f^2\})$ generate a set of F^1-indicative outputs to distinguish q_{13} and q_{14} from q_9. Hence, condition (1) of the theorem is satisfied too for the failure mode F^1. Failure mode F^1 becomes diagnosable in $\tilde{H}$ because the outputs produced by the isolators $Is^{Q^{E_2M_2}}(\{f^1\})$, $Is^{Q^{E_2M_2}}(\{f^1, f^2\})$ and $Is^{Q^{E_2M_2}}(\{f^2\})$ can distinguish q_8 from q_2, q_5 and q_7, and the outputs produced by the isolators $Is^{Q^{E_3M_3}}(\{f^2\})$ and $Is^{Q^{E_8M_8}}(\emptyset)$ can distinguish the cycle of q_{12} and q_{15} from q_6. Therefore, condition (2) of Theorem 2.3.1 is satisfied for the failure mode F^2.*

We can verify that failure modes F^1 and F^2 are Π-diagnosable for any $\Pi \geq 1$. This implies that when F^1 or F^2 occur, they can be detected and isolated after a maximum of one event generated in the system. ■

So far, we have studied diagnosability in the EDESA of a hybrid automaton with isolators. The diagnosability results developed for the EDESA are based on the discrete outputs of the system and outputs generated by the isolators designed for the system. Generally, the constraints for diagnosability analysis in DES is expressed in terms of the number of events generated in the system. However, in hybrid automata, the notion of time is included in the model. Therefore, it is desirable to develop diagnosability results with respect to time requirements. In order to discuss diagnosability of failure modes with time constraints, we describe the notion of *execution* in hybrid automata. First, we define a hybrid time set.

Definition 4.1.2. *A **hybrid time set** [60] is either a finite sequence of intervals $\tau = \{I_i\}_{i=0}^{\tilde{N}}$ such that*

- $I_i = [\tau_i, \tau_i']$ *for all* $0 \leq i < \tilde{N}$ *with* $\tau_0 \leq \tau_0' = \tau_1 \leq \tau_1' \leq \cdots \leq \tau_{\tilde{N}-1} \leq \tau_{\tilde{N}-1}'$;
- $I_{\tilde{N}} = [\tau_{\tilde{N}}, \tau_{\tilde{N}}')$;

or an infinite sequence of intervals $\tau = \{I_i\}_{i=0}^{\infty}$ such that

- *$I_i = [\tau_i, \tau_i']$ for all $i \geq 0$ with $\tau_0 \leq \tau_0' = \tau_1 \leq \tau_1' \leq \cdots$.* ■

The τ_i's are the times at which discrete transitions take place. We assume that all the continuous models S_q are time-invariant. Therefore, without loss of generality we assume that $\tau_0 = 0$. Let $|\tau| = \tilde{N} + 1$ be the cardinality of τ that gives the number of intervals. For a hybrid time trajectory $\tau = \{I_i\}_{i=0}^{\tilde{N}}$, we define $< \tau >$ as the set $\{0, 1, \cdots, \tilde{N}\}$ if $\tilde{N}$ is finite and $\{0, 1, \cdots\}$ if $\tilde{N} = \infty$. In the following, we define an execution of hybrid automata in our work.

Definition 4.1.3. *An **execution** of a hybrid automaton [60] is a tuple $e = (\tau^e, q^e, x^e)$, where $\tau^e = \{I_i^e\}_{i=0}^{\tilde{N}^e}$ is a hybrid time set with time intervals $I_i^e = [\tau_i^e, \tau_i'^e]$ for all $0 \leq i \leq \tilde{N}^e$; $q^e :< \tau^e > \longrightarrow Q$ is a map such that $q^e(0) = q_0$; $x^e = \{x_i^e : i \in < \tau^e >\}$ is a set of differentiable maps $x_i^e : I_i^e \longrightarrow \mathcal{X}$, such that*

- *$x_0^e(0) \in Init$;*

- *for all $t \in [\tau_i, \tau_i')$, $\dot{x^e}_i(t) = E_{q^e(i)}(x_i^e(t), u_i^e(t)) + G_{q^e(i)}(f_{1,i}^e(t), \cdots, f_{m,i}^e(t))$, where $u_i^e : I_i^e \longrightarrow \mathcal{U}$ and $f_{j,i}^e : I_i \longrightarrow \mathbb{R}$ (for $1 \leq j \leq m$) are the maps which give the input signal and the signal of active fault types in interval I_i^e, respectively;*

- *if $\tilde{N}^e$ is finite, for all $i \in < \tau > \backslash \{\tilde{N}^e\}$, there exists $\sigma \in \Sigma :$ $d = (q^e(i), \sigma, q^e(i+1)) \in T$, $G(d, x_i^e(\tau_i')) = \{True\}$ and $x_{i+1}^e(\tau_{i+1}) = \rho(d, x_i^e(\tau_i'))$.*

- *if $\tilde{N} = \infty$, for all $i \in < \tau >$, there exists $\sigma \in \Sigma :$ $d = (q^e(i), \sigma, q^e(i+1)) \in T$, $G(d, x_i^e(\tau_i')) = \{True\}$ and $x_{i+1}^e(\tau_{i+1}) = \rho(d, x_i^e(\tau_i'))$.* ■

The maps q^e and x^e describe the evolution of the discrete states q and the continuous state x, respectively. They satisfy the discrete and the continuous dynamics and their interactions (initial state, guard conditions and reset maps) in the system. The map q^e is the **discrete state trajectory**, and x^e is the **continuous state trajectory** of the system for execution e.

Let $\mathcal{E}$ be the **set of all executions** of H. For any execution $e \in \mathcal{E}$, we have $|e| = |\tau^e|$. In this paper, we assume that all hybrid executions are defined for all $t \geq 0$. In other words, we study *non-blocking* hybrid automata. We do not have Zeno executions [60] (executions with infinite discrete transitions in a finite time) in our framework, therefore, the executions are infinite, i.e., $\Sigma_{k=0}^{|e|}(\tau_k'^e - \tau_k^e) = \infty$. However the hybrid time sets may be finite or infinite.

Let $e \in \mathcal{E}$ be an execution. **The continuous output signal** y^e for execution e is the set of maps $y^e = \{y_i^e : i \in < \tau^e >\}$ with $y_i^e : I_i^e \longrightarrow \mathcal{Y}$, such that for all $t \in [\tau_i^e, \tau_i'^e)$,

$$y_i^e(t) = M_{q^e(i)}(x_i^e(t), u_i^e(t))$$

Let $t_i \in I_i^e$ and $t_j \in I_j^e$ with $i < j$. The map $x^e|_{t_i}^{t_j}$ denotes the continuous state trajectory from time t_i to t_j for execution e:

$$x^e|_{t_i}^{t_j} = \{x_i^e(t) \mid t_i \leq t \leq \tau_i'^e, x_{i+1}^e, \cdots, x_{j-1}^e, x_j^e(t) \mid \tau_j^e \leq t \leq t_j\}$$

Similarly, $u^e|_{t_i}^{t_j}$, $f^e|_{t_i}^{t_j}$ and $y^e|_{t_i}^{t_j}$ denote the control input, fault type and output signals from time t_i to t_j for execution e.

The map $\mu : \mathcal{E} \longrightarrow D^*$ associates a sequence of discrete output to each execution e as

$$\mu(e) = \lambda(q^e(0))\lambda(q^e(1)) \cdots \lambda(q^e(|e|))$$

We denote by $\mu(e)|_{t_i}^{t_j}$ the sequence of the discrete outputs associated with e from time t_i to time t_j:

$$\mu(e)|_{t_i}^{t_j} = \lambda(q^e(i))\lambda(q^e(i+1)) \cdots \lambda(q^e(j)) \text{ such that } t_i \in I_i^e \text{ and } t_j \in I_j^e$$

We omit all discrete outputs $\lambda(q^e(j))$ such that $\lambda(q^e(j)) = \lambda(q^e(j+1))$ from $\mu(e)$ and denote the new sequence by $\hat{\mu}(e)$. For instance, if $\mu(e) = aab$, then $\hat{\mu}(e) = ab$.

The map $\hat{\mu}(e)$ shows only the output changes that are used for diagnosis. The map $\hat{\mu}(e)|_{t_i}^{t_j}$ will be similarly constructed from $\mu(e)|_{t_i}^{t_j}$.

An execution $e \in \mathcal{E}$ is called an **$\mathbf{F^i}$-faulty execution** if there exists $0 \leq k_i \leq |e|$ such that for all $k < k_i$, $q^e(k) \notin Q_{\mathcal{F}^i}$ and $q^e(k_i) \in Q_{\mathcal{F}^i}$. Let $t^e_{F^i}$ be the time that F^i-faulty execution e enters $Q_{\mathcal{F}^i}$. Since the occurrence of each failure mode is modeled by an unobservable transition, execution e enters $Q_{\mathcal{F}^i}$ at time $t^e_{F^i} = \tau^e_{k_i}$. Let $\mathcal{E}^{F^i}$ be the set of all F^i-faulty executions. For diagnosability analysis, we assume that the failure modes are permanent.

Definition 4.1.4. *Failure mode F^i is* ***permanent*** *if for any $e \in \mathcal{E}^{F^i}$ such that $q^e(k_i) \in Q_{\mathcal{F}^i}$, then $q^e(k) \in Q_{\mathcal{F}^i}$ for all $k_i \leq k \leq |e|$.*

We intend to express diagnosability in hybrid automata in terms of measured variables. Measured variables in our hybrid automata model are discrete outputs at discrete states and the continuous input and output signals at the continuous dynamics. In the following, we present a definition for diagnosability in hybrid automata in terms of discrete outputs and continuous input and output signals.

Let $\Delta \geq 0$ be a positive real number. We call a failure mode F^i Δ-diagnosable if it is always possible to detect, with a delay no longer than Δ following the occurrence of F^i, whether the system has visited the set $Q_{\mathcal{F}^i}$ (by using the sequence of discrete outputs and continuous output signal of the system.

Definition 4.1.5. *Assume that a permanent failure mode F^i occurs at $t = t_0 \geq 0$ and the diagnosis starts at $t_d \geq 0$. Also let $t_m = max(t_0, t_d)$. Failure mode F^i is* ***Δ-diagnosable*** *in H if for all $e \in \mathcal{E}^{F^i}$ and $e' \in \mathcal{E} - \mathcal{E}^{F^i}$, $\hat{\mu}(e)|_{t_m}^{(t_m+\Delta)} \neq \hat{\mu}(e')|_{t_m}^{(t_m+\Delta)}$ or* $\begin{bmatrix} u^e \\ y^e \end{bmatrix} \Bigg|_{t_m}^{t_m+\Delta} \neq \begin{bmatrix} u^{e'} \\ y^{e'} \end{bmatrix} \Bigg|_{t_m}^{t_m+\Delta}$[2]. ■

[2] We say $u^e|_{t_1}^{t_2} \neq u^{e'}|_{t_1}^{t_2}$ if the set $\{t \mid t_1 \leq t \leq t_2 \text{ and } t \in I_i^e \text{ and } t \in t_j^{e'} : u_i^e(t) = u_j^{e'}(t)\}$ is of measure zero. Similarly, we say $y^e|_{t_1}^{t_2} \neq y^{e'}|_{t_1}^{t_2}$ if the set $\{t \mid t_1 \leq t \leq t_2 \text{ and } t \in I_i^e \text{ and } t \in t_j^{e'} : y_i^e(t) = y_j^{e'}(t)\}$ is of measure zero.

Proposition 4.1.3. *If a failure mode F^i is not Δ-diagnosable in H, F^i is not Δ'-diagnosable in H for all $\Delta' \leq \Delta$.*

Proof - Consider the contrary and assume that F^i is Δ'-diagnosable in H. Therefore by definition, for all $e \in \mathcal{E}^{F^i}$ and $e' \in \mathcal{E} - \mathcal{E}^{F^i}$, $\hat{\mu}(e)|_{t_m}^{(t_m+\Delta')} \neq \hat{\mu}(e')|_{t_m}^{(t_m+\Delta')}$ or $u^e|_{t_m}^{t_m+\Delta'} \neq u^{e'}|_{t_m}^{t_m+\Delta'}$ or $y^e|_{t_m}^{t_m+\Delta'} \neq y^{e'}|_{t_m}^{t_m+\Delta'}$. As a result, for any $\Delta \geq \Delta'$, we also have for all $e \in \mathcal{E}^{F^i}$ and $e' \in \mathcal{E} - \mathcal{E}^{F^i}$, $\hat{\mu}(e)|_{t_m}^{(t_m+\Delta)} \neq \hat{\mu}(e')|_{t_m}^{(t_m+\Delta)}$ or $u^e|_{t_m}^{t_m+\Delta} \neq u^{e'}|_{t_m}^{t_m+\Delta}$ or $y^e|_{t_m}^{t_m+\Delta} \neq y^{e'}|_{t_m}^{t_m+\Delta}$. Thus, F^i is Δ-diagnosable in H. ■

In addition to discrete outputs generated by the system, the EDESA includes the information provided by the isolators between every two consecutive discrete transitions. In the following, we show that a failure mode is diagnosable in H, if it is diagnosable in the EDESA.

Theorem 4.1.4. *Assume that the diagnoser of $\tilde{H}$ is initialized with $z_0 = \tilde{Q}$. For any $\Delta > 0$, a permanent failure mode F^i is Δ-diagnosable in H if F^i is Π-diagnosable in $\tilde{H}$ for some $\Pi \leq [\frac{\Delta}{\tau_{max}}]$.* ■

Proof - By contradiction, we prove that if F^i is not Δ-diagnosable (for a $\Delta \geq 0$) in H, there exists no $\Pi \leq [\frac{\Delta}{\tau_{max}}]$ such that F^i is Π-diagnosable in $\tilde{H}$.

Assume that the failure mode F^i corresponding to the fault type f^i is not Δ-diagnosable in H. Thus, there exist $e \in \mathcal{E}^{F^i}$ and $e' \in \mathcal{E} - \mathcal{E}^{F^i}$ such that both executions e and e' will generate the same sequence of discrete outputs and identical continuous input and output signals from t_m to $t_m+\Delta$, i.e., $\hat{\mu}(e)|_{t_m}^{(t_m+\Delta)} = \hat{\mu}(e')|_{t_m}^{(t_m+\Delta)}$ and $u^e|_{t_m}^{t_m+\Delta} = u^{e'}|_{t_m}^{t_m+\Delta}$ and $y^e|_{t_m}^{t_m+\Delta} = y^{e'}|_{t_m}^{t_m+\Delta}$. Therefore, any isolator $Is \in \mathbf{IS}_{tot}$ that takes the continuous input and output of the system generates identical outputs for e and e' from t_m to $t_m + \Delta$. Thus, the FSA modeling Is will generate identical discrete outputs and events for e and e' from t_m to $t_m + \Delta$. Let $0 \leq i \leq j \leq |e|$ such that $t_m \in I_i^e$ and $(t_m + \Delta) \in I_j^e$. Also let $0 \leq i' \leq j' \leq |e'|$ such that $t_m \in I_{i'}^{e'}$ and $(t_m + \Delta) \in I_{j'}^{e'}$. The output sequence generated by $\tilde{H}$ consists of the system

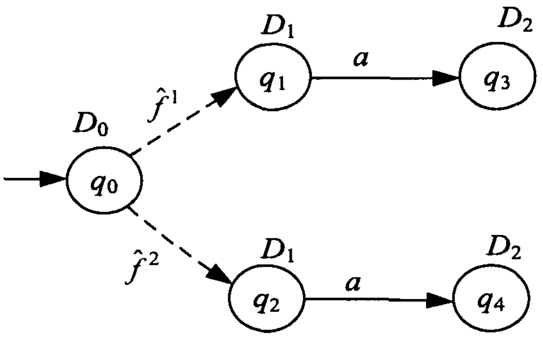


Figure 4.3: A hybrid automaton with two failure modes.

discrete outputs and outputs generated by all isolators between each system output. Hence, the output sequence generated by $\tilde{H}$ from t_m to $t_m + \Delta$ will be identical for both executions e and e'. Let $q^e(i)$ and $q^{e'}(i')$ denote the initial discrete states of $\tilde{H}$ on e and e', respectively. Also let $q^e(j)$ and $q^{e'}(j')$ denote the final discrete states of $\tilde{H}$ on e and e', respectively. If the diagnoser for $\tilde{H}$ is started at $t_d = t_m$, then $q^e(i), q^{e'}(i') \in z_0$, and since the output sequences of $\tilde{H}$ on e and e' are the same, then $q^e(j), q^{e'}(j') \in z(t_m + \Delta)$, where $z(t_m + \Delta)$ is the state estimate provided by the diagnoser at $t_m + \Delta$. Thus, $z(t_m + \Delta)$ is F_i-uncertain. If all the discrete states from $q^e(i)$ to $q^e(j)$ are not in Q^{inf}, then during e, at least $\frac{\Delta}{\tau^{max}}$ events must have been generated by $\tilde{H}$. If one or more of discrete states from $q^e(i)$ to $q^e(j)$ is in Q^{inf}, considering the fictitious self-loops for the states of Q^{inf}, then the sequence from $q^e(i)$ to $q^e(j)$ can be considered to have an arbitrary large number of events. Since the final diagnoser state is F^i-uncertain, then F^i is not Π-diagnosable for any $\Pi \leq [\frac{\Delta}{\tau^{max}}]$. ■

Example 4.1.2. *(Example 4.1.1 Continued): Assume that* $\tau^{max} = 100$ *sec. Therefore,* F^1 *and* F^2 *are* $\Delta-$*diagnosable in the hybrid system shown in Figure 4.2 for any* $\Delta \geq 100$ *sec.* ■

Example 4.1.3. *Consider the hybrid automaton in Figure 4.3 with fault types* f^1 *and* f^2 *corresponding to failure modes* F^1 *and* F^2*, respectively. Assume the following* *dynamics for the system.*

$$S_0 := \begin{cases} \dot{x} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} x + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u \\ y = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} x \end{cases}$$

$$S_1 := \begin{cases} \dot{x} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} x + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u + \begin{bmatrix} 1 \\ -1 \end{bmatrix} f^1 \\ y = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} x \end{cases}$$

$$S_2 := \begin{cases} \dot{x} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} x + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u + \begin{bmatrix} -2 \\ 2 \end{bmatrix} f^2 \\ y = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} x \end{cases}$$

$$S_3 := \begin{cases} \dot{x} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \end{bmatrix} u + \begin{bmatrix} 0.1 \\ 2 \end{bmatrix} f^1 \\ y = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} x \end{cases}$$

$$S_4 := \begin{cases} \dot{x} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \end{bmatrix} u + \begin{bmatrix} -0.3 \\ 1.2 \end{bmatrix} f^2 \\ y = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} x \end{cases}$$

Let $F^1(s)$ and $F^2(s)$ be the Laplace transform of f^1 and f^2, respectively. The *continuous output of the hybrid system in the discrete state q_3 is*

$$Y(s) = \begin{bmatrix} 0.1 \\ 2 \end{bmatrix} \frac{F^1(s)}{s+1}$$

The continuous output of the hybrid system in the discrete state q_4 is

$$Y(s) = \begin{bmatrix} 1.2 \\ -0.3 \end{bmatrix} \frac{F^2(s)}{s+1}$$

It can be verified that the continuous output of the hybrid system in q_3 is different from that in q_4. Therefore, according to Definition 4.1.5, F^1 and F^2 are diagnosable. Let $Is^{q_3}(f^1)$ be the isolator designed based on the dynamics of q_3. According to the properties of the isolators in our framework, $Is^{q_3}(f^1)$ must generate zero if the system is in an EM-similar discrete state (here, a discrete state with the same A, B and C matrices) where f^1 is not active. Otherwise, $Is^{q_3}(f^1)$ must generate one. It can be verified that $Is^{q_3}(f^1)$ can be designed. Also let $Is^{q_4}(f^2)$ be the isolator designed based on the dynamics of q_4. Similarly, it can be verified that $Is^{q_4}(f^2)$ exists. The isolator $Is^{q_3}(f^1)$ generates one in q_3 and q_4 (see Assumption 1 in Section 3.2) at all times. Similarly, the isolator $Is^{q_4}(f^2)$ generates one in q_3 and q_4 at all times. Hence, the transitions and discrete outputs that $Is^{q_3}(f^1)$ and $Is^{q_4}(f^2)$ insert in the EDESA of the system and isolators are identical and failure modes F^1 or F^2 remain undiagnosable in the EDESA. This simple example demonstrates that the diagnosability of a failure mode in the EDESA is not necessary for the diagnosability of that failure mode in the hybrid automaton. ■

4.2 Fault Diagnosis and Diagnosability of Failure Modes in the Presence of Unreliable Isolators

So far, we have assumed that isolators function without error. However, in practice, isolators may generate incorrect output (error). The generation of incorrect output by isolators may have several reasons. One common reason that isolators may generate unreliable output is the sensitivity of isolators to thresholds due to modeling uncertainty and noise. In some cases, when the residual should be zero, due to noise and modeling uncertainty, a nonzero residual close to the threshold may be generated which triggers a false alarm. Some times even in the presence of failures, the residual does not reach the threshold because of the interference by noise or modeling uncertainty. In this work, we assume that the generation of incorrect output by the isolators is intermittent implying that an isolator may generate incorrect output every now and then. An isolator that may generate incorrect output is not reliable for fault diagnosis.

4.2.1 Problem Formulation

The problem that we solve in this section is the diagnosability of failure modes in the presence of unreliable isolators. If the isolator generating incorrect output can be identified, it can be removed from the set of isolators. In this case, the diagnosability of failure modes can be investigated using the new set of isolators. However, in practice, identifying the incorrect output of an isolator is not an easy task. One solution is removing all the isolators that may generate incorrect output from the set of isolators. This solution is not reasonable because the isolators do not generate unreliable output at all time. Moreover, it is very unlikely that all isolators generate incorrect output simultaneously. We will assume that the number of isolators generating incorrect output at a given time is bounded and the upper bound is known. We formulate the problem as follows.

Given a set of b unreliable isolators $\mathbf{IS}_{tot} = \{Is_1, \cdots, Is_b\}$ designed for the system, and assuming that at any given time up to $k_{sim} \leq b$ isolators may generate incorrect output simultaneously, we want to investigate if there is sufficient redundant information from isolators and discrete sensors to make a failure mode F diagnosable in the EDESA of the system and isolators.

In the following, we develop a systematic approach for verifying diagnosability of failure modes in the presence of unreliable isolators. In this approach, we first modify the model of an isolator to take into account potential isolator errors. Then, we construct the EDESA for the system and isolators. Diagnosability of the failure modes will be verified in the EDESA of the hybrid system and isolators similar to that explained in Section 4.1.

4.2.2 Modeling Unreliable Isolators

In general, an isolator may generate incorrect output in two cases.

1. The isolator should produce zero (no fault) but it produces one (fault present). In this case, the output of the isolator is called a **false alarm**.

2. The isolator should produce one (fault present) but it produces zero (no fault). In this case, the output of the isolator is called a **false silence**.

We modify the model of a residual generator to include these two types of errors. Let Is be an unreliable isolator. Figure 4.4 shows the FSA modeling the isolator Is with unreliable behavior. The isolator has three modes of operation: normal, false alarm and false silence. The generation of incorrect output by the isolator is modeled by the occurrence of unobservable events. The unobservable event '$Is : fa$' labels the transition from normal to false alarm mode, and the unobservable event '$Is : fs$' labels the transition from normal to false silence mode. Isolator errors

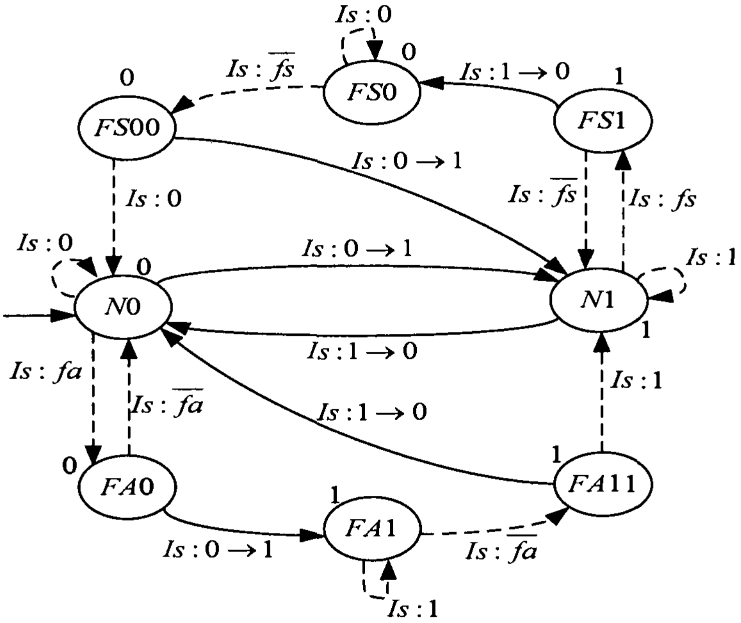


Figure 4.4: A finite-state automaton modeling an unreliable isolator.

are assumed intermittent. Therefore, the isolator may have a transition from false alarm mode or false silence mode to normal. Both of these transitions are assumed unobservable. The transition of the isolator from false alarm mode to normal mode is labeled by the unobservable event 'Is : $\overline{fa}$', and the transition of the isolator from false alarm mode to normal mode is labeled by the unobservable event 'Is : $\overline{fs}$'.

For any isolator $Is \in \mathbf{IS}_{tot}$,

$$\overline{Is} = (Q^{Is}, \Sigma^{Is}, T^{Is}, D^{Is}, \lambda^{Is}, q_0^{Is})$$

will be the FSA model of Is. We have

$$Q^{Is} = \{N0, N1, FA0, FA1, FA211, FS0, FS00, FS1\}$$

$D^{Is} = \{0, 1\}$ is the set of discrete output symbols with $\lambda^{Is}(N0) = 0$, $\lambda^{Is}(N1) = 1$, $\lambda^{Is}(FA0) = 0$, $\lambda^{Is}(FA1) = 1$, $\lambda^{Is}(FA11) = 1$, $\lambda^{Is}(FS0) = 0$, $\lambda^{Is}(FS00) = 0$,

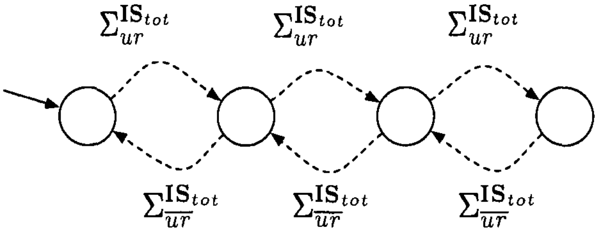

$\Sigma_{ur}^{\mathbf{IS}_{tot}}$
$\Sigma_{ur}^{\mathbf{IS}_{tot}}$
$\Sigma_{ur}^{\mathbf{IS}_{tot}}$
$\Sigma_{\overline{ur}}^{\mathbf{IS}_{tot}}$
$\Sigma_{\overline{ur}}^{\mathbf{IS}_{tot}}$
$\Sigma_{\overline{ur}}^{\mathbf{IS}_{tot}}$

Figure 4.5: Finite-state automaton $ASM_{k_{sim}}$ for the case that $k_{sim} = 3$.

$\lambda^{Is}(FS1) = 1$. The isolator event set is

$$\Sigma^{Is} = \{Is : 0 \rightarrow 1, Is : 1 \rightarrow 0, Is : 0, Is : 1, Is : fa, Is : \overline{fa}, Is : fs, Is : \overline{fs}\}$$

Let $\Sigma_{ur}^{Is} = \{Is : fa, Is : fs\}$ and $\Sigma_{\overline{ur}}^{Is} = \{Is : \overline{fa}, Is : \overline{fs}\}$ be two event sets. We define $\Sigma_{ur}^{\mathbf{IS}_{tot}}$ and $\Sigma_{\overline{ur}}^{\mathbf{IS}_{tot}}$ as follows:

$$\Sigma_{ur}^{\mathbf{IS}_{tot}} = \bigcup_{Is \in \mathbf{IS}_{tot}} \Sigma_{ur}^{Is}$$

$$\Sigma_{\overline{ur}}^{\mathbf{IS}_{tot}} = \bigcup_{Is \in \mathbf{IS}_{tot}} \Sigma_{\overline{ur}}^{Is}$$

4.2.3 Constructing EDESA of the System and Isolators in the Presence of Unreliable Isolators

The assumption that at any given time up to $k_{sim} \leq b$ isolators may generate incorrect output simultaneously can be enforced by an FSA denoted as $ASM_{k_{sim}}$. Figure 4.5 shows the FSA enforcing this assumption for the case that $k_{sim} = 3$.

Similar to the diagnoser design procedure described in Chapter 3, we make the following assumption: one event of every isolator (not considering events introduced for modeling isolator errors, i.e, $Is : fa$, $Is : \overline{fa}$, $Is : fs$ and $Is : \overline{fs}$) must occur between the occurrence of any two consecutive events in the system. We also assume that the isolators reach their steady state before the first event in the system occurs.

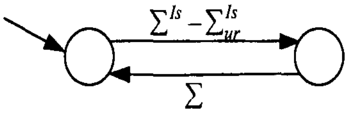


Figure 4.6: Finite-state automaton ASM_{Is}.

Moreover, the isolators may have a state transition only when an event occurs in the system. Let ASM_{Is} be the FSA enforcing these assumptions for Is. The FSA ASM_{Is} is shown in Figure 4.6. The transition labeled with Σ implies that any event in the event set of the system can label that transition.

Let $\overline{\mathbf{IS}}_{tot} = \{\overline{Is}_1, \cdots, \overline{Is}_b\}$ be the set of FSA modeling the isolators. The EDESA of the hybrid system and isolators, denoted as $\tilde{H}$, is an FSA

$$\tilde{H} = (\tilde{Q}, \tilde{\Sigma}, \tilde{T}, \tilde{D}, \tilde{\lambda}, \tilde{q}_0)$$

and is constructed by combining H_{abs}, the FSA modeling the isolators, the FSA $ASM_{k_{sim}}$ and the FSA enforcing the assumptions $ASM_{Is_1}, \cdots, ASM_{Is_b}$:

$$\tilde{H} = \mathbf{sync}(H_{abs}, \overline{IS}, ASM_{k_{sim}}, ASM)$$

where

$$\overline{IS} = \mathbf{sync}(\overline{Is}_1, \cdots, \overline{Is}_b)$$

and

$$ASM = \mathbf{sync}(ASM_{Is_1}, \cdots, ASM_{Is_b})$$

A DES diagnoser is designed for the EDESA of the hybrid system and isolators similar to that explained in Section 2.2. Diagnosability of the failure modes will be verified in the EDESA similar to that explained in Section 4.1.

Remark 4.2.1. *Here we assumed that all the isolators designed for the system may generate incorrect output. In other words, all the isolators have been assumed to be* *unreliable. However, in practice, only a subset of isolators (depending on the method used for designing the isolators) may be assumed unreliable. In this case, the set of isolators* $\mathbf{IS}_{tot}$ *can be partitioned as follows.*

$$\mathbf{IS}_{tot} = \mathbf{IS}_{rel} \dot{\bigcup} \mathbf{IS}_{unrel}$$

where $\mathbf{IS}_{rel}$ *is the set of isolators that always generate reliable output, and* $\mathbf{IS}_{unrel}$ *is the set of unreliable isolators. For the set of* $\mathbf{IS}_{rel}$*, we construct the FSA* $\hat{H}_{abs}$ *as explained in Section 3.2 of Chapter 3 to guarantee the consistency between the transitions of isolators of* $\mathbf{IS}_{rel}$ *and the hybrid system. Isolators of* $\mathbf{IS}_{unrel}$ *may have arbitrary transitions. Therefore, we do not consider isolators of* $\mathbf{IS}_{unrel}$ *in the FSA* $\hat{H}_{abs}$*. The EDESA* $\tilde{H}$ *can be computed as*

$$\tilde{H} = \mathbf{sync}(\hat{H}_{abs}, Is, ASM_{k_{sim}}, ASM) \qquad \blacksquare$$

Example 4.2.1. *(Example 4.1.1 Continued): Again we assume that the set of isolators that can be designed for the system is* $\mathbf{IS}_{tot} = \{\ Is^{Q^{E_2M_2}}(\{f^1\}), Is^{Q^{E_2M_2}}(\{f^1, f^2\}), Is^{Q^{E_2M_2}}(\{f^2\}), Is^{Q^{E_3M_3}}(\{f^2\}), Is^{Q^{E_5M_5}}(\{f^1\}), Is^{Q^{E_5M_5}}(\{f^2\}), Is^{Q^{E_8M_8}}(\emptyset)\ \}$. *As explained earlier, both failure modes* F^1 *and* F^2 *are diagnosable in* $\tilde{H}$ *when* $k_{sim} = 0$. *We can verify that if* $k_{sim} = 1$*, the failure mode* F^2 *remains diagnosable because for distinguishing* q_8 *from* q_2*,* q_5 *and* q_7 *two of the isolators* $Is^{Q^{E_2M_2}}(\{f^1\})$*,* $Is^{Q^{E_2M_2}}(\{f^1, f^2\})$ *and* $Is^{Q^{E_2M_2}}(\{f^2\})$ *are sufficient. Moreover, for distinguishing the cycle of* q_{12} *and* q_{15} *from* q_6*, only one of the isolators* $Is^{Q^{E_3M_3}}(\{f^2\})$ *and* $Is^{Q^{E_8M_8}}(\emptyset)$ *is required. Also for distinguishing* q_{14} *from* q_9 *and* q_{13} *only one of the isolators* $Is^{Q^{E_5M_5}}(\{f^1\})$ *and* $Is^{Q^{E_5M_5}}(\{f^2\})$ *is necessary. We can also verify that if* $k_{sim} = 1$*, the failure mode* F^1 *becomes undiagnosable because for distinguishing* q_{13} *and* q_{14} *from* q_9 *both of the isolators* $Is^{Q^{E_5M_5}}(\{f^1\})$ *and* $Is^{Q^{E_5M_5}}(\{f^2\})$ *are required.* $\blacksquare$

4.3 Summary

In this chapter, we investigated diagnosability of failure modes. We introduced the notion of diagnosability in hybrid automata and showed that a failure mode in a hybrid automaton is diagnosable if the failure mode is diagnosable in the EDESA of the hybrid automaton and isolators. We also investigated diagnoser design and diagnosability of failure modes in the case that some isolators generate unreliable outputs.

In this chapter, we assumed that all the isolators that can be designed are used for diagnosability analysis. In the next chapter, we investigate the problem of isolator selection and develop approaches for computing a minimal set of isolators ensuring failure diagnosability.

Chapter 5

ISOLATOR SELECTION IN HYBRID AUTOMATA

In Chapter 3, we developed a framework for diagnosis in hybrid systems. Combining the DES abstraction of the system and the isolators designed for isolating the fault types, we developed an Extended Discrete-Event System Abstraction (EDESA) for the system and the isolators. We introduced a notion of diagnosability in hybrid automata based on the sequence of discrete outputs and the continuous output signal. We showed that if a failure mode is diagnosable in the EDESA of a hybrid automaton and isolators, it will be diagnosable in the hybrid automaton. In Chapter 3, we assumed that all the isolators that satisfy the existence conditions are designed and used for diagnosis. However, some of the isolators may be redundant. For example, let $Q^{EM} \subseteq Q$ be a set of EM-similar discrete state and $FT^{Q^{EM}} = \{f^1, f^2\}$. Assume that the following isolators have been designed:

$$Is^{Q^{EM}}(\{f^1\}), Is^{Q^{EM}}(\{f^2\}), Is^{Q^{EM}}(\{f^1, f^2\})$$

Suppose the system is in a discrete state of Q^{EM}. Since we can individually isolate f^1 and f^2 by $Is^{Q^{EM}}(\{f^1\})$ and $Is^{Q^{EM}}(\{f^2\})$, respectively, we do not obtain

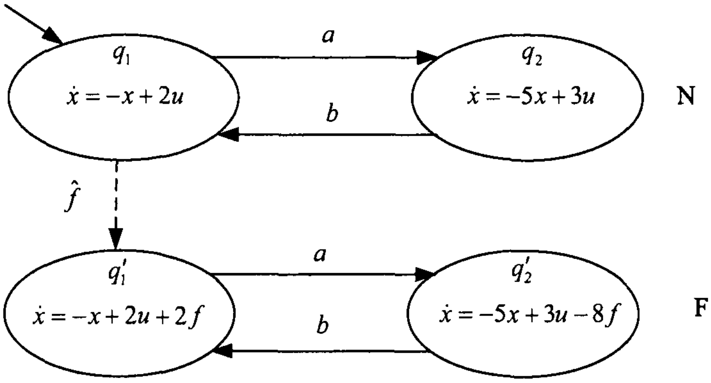


Figure 5.1: A hybrid automaton with one fault type.

more information from the isolator $Is^{Q^{EM}}(\{f^1, f^2\})$. Therefore, $Is^{Q^{EM}}(\{f^1, f^2\})$ is a redundant isolator and can be discarded from the set of isolators. As another example, consider the hybrid automaton in Figure 5.1. Suppose $Q^{EM} = \{q_1, q_1'\}$ and $Q^{E'M'} = \{q_2, q_2'\}$ are two sets of EM-similar discrete states. The fault type present in the system is f with the corresponding failure mode F, and $f \in FT^{Q^{EM}}$ and $f \in FT^{Q^{E'M'}}$. Assume that we can design the isolators $Is^{Q^{EM}}(\{f\})$ and $Is^{Q^{E'M'}}(\{f\})$ and the diagnosis system starts simultaneously with the system. As can be seen in Figure 5.1, F can only occur when the system is in Q^{EM}. Therefore, when f becomes active, the isolator $Is^{Q^{EM}}(\{f\})$ can isolate F, and there is no need to use $Is^{Q^{E'M'}}(\{f\})$.

In this chapter, we study the problem of isolator selection, that is, the problem of selecting sufficient isolators for ensuring the diagnosability of a failure mode in a hybrid automaton. We investigate the problem of "minimal isolator set" which is to find a minimal isolator set for attaining the diagnosability of a failure mode in the EDESA of the hybrid system and isolators.

The remainder of this chapter is organized as follows. In Section 5.1, we formulate the problem of isolator selection. Isolator selection for distinguishing discrete states from each other is studied in Section 5.2. A bottom-up algorithm for isolator selection in a hybrid automaton is developed in Section 5.3.

5.1 Problem Formulation

In our framework, we use the observations of discrete sensors and residuals generated by the isolators for fault diagnosis. Therefore, failure diagnosability depends in part on the set of discrete sensors and the set of isolators used. Discrete sensor selection for DES has been studied for example in [90, 32, 5]. In this work, we study the problem of isolator selection in hybrid automata. Assuming failure diagnosability, some of the isolators may provide redundant information and therefore, they may be not necessary for fault diagnosis. Thus, we want to investigate the problem of selecting a minimal set of isolators to ensure the diagnosability of failure modes in the system.

In this chapter, we study the problem of minimal isolator selection. In this problem, we are given a hybrid automaton model of the system with a set of failure modes and a set of isolators that can be designed for the system. The objective is to find a minimal isolator set such that a specific failure mode remains diagnosable in the EDESA of the system and isolators. Assume that the system to be diagnosed is a hybrid automaton

$$H = (Q, \mathcal{X}, \mathcal{U}, \mathcal{Y}, FT, Init, S, \Sigma, T, G, \rho, D, \lambda, q_0)$$

and let $\mathbf{IS}_{tot}$ be the set of all isolators designed for the system. For a set of isolators $\mathbf{IS} \subseteq \mathbf{IS}_{tot}$, let $\tilde{H}(\mathbf{IS})$ denote the EDESA constructed based on the system H and isolators of $\mathbf{IS}$. Also let $\mathbf{SIS}^{F^i}$ be the set of isolator sets for which a failure mode F^i is diagnosable in the EDESA of the system and isolators $\tilde{H}$, namely:

$$\mathbf{SIS}^{F^i} = \{\mathbf{IS} \mid \mathbf{IS} \subseteq \mathbf{IS}_{tot} \text{ and } F \text{ is diagnosable in } \tilde{H}(\mathbf{IS})\}$$

In this chapter, we investigate the problem of finding minimal elements of

$\mathbf{SIS}^{F^i}$. An isolator set $\mathbf{IS}$ is a minimal element of $\mathbf{SIS}^{F^i}$ if $\mathbf{IS} \in \mathbf{SIS}^{F^i}$ and for any $\mathbf{IS}' \subset \mathbf{IS}$ ($\mathbf{IS}' \neq \mathbf{IS}$), we have $\mathbf{IS}' \notin \mathbf{SIS}^{F^i}$.

If by using $\mathbf{IS}_{tot}$, F^i is not diagnosable in the EDESA of the system and isolators, no subset of the $\mathbf{IS}_{tot}$ can make F^i diagnosable, and we assume that $\mathbf{SIS}^{F^i}$ does not exists.

A straightforward *top-down* solution for finding a minimal element of $\mathbf{SIS}^{F^i}$ is given in Procedure 5.1. In Procedure 5.1, $\mathbf{IS}^{F^i}_{Min}$ is a minimal element of $\mathbf{SIS}^{F^i}$.

Procedure 5.1: Given an isolator set $\mathbf{IS}_{tot}$.

```
1. Initialization: $\mathbf{IS}^{F^i}_{Min} = \mathbf{IS}_{tot}$
2. For all $Is \in \mathbf{IS}_{tot}$
     $\mathbf{IS}^{F^i}_{Min} = \mathbf{IS}^{F^i}_{Min} - \{Is\}$
     Check the conditions of Theorem 2.3.1 for $\tilde{H}(\mathbf{IS}^{F^i}_{Min})$
     If $\mathbf{IS}^{F^i}_{Min} \notin \mathbf{SIS}^{F^i}$ (conditions of Theorem 2.3.1 fail)
         $\mathbf{IS}^{F^i}_{Min} = \mathbf{IS}^{F^i}_{Min} \bigcup \{Is\}$
     End (If)
   End (For)
```

In general, $\mathbf{IS}_{tot}$ may consist of a large number of isolators. Verifying the solvability conditions for the existence of all the isolators of $\mathbf{IS}_{tot}$ may be computationally intensive. Moreover, in Procedure 5.1, $\tilde{H}$ has to be computed and the conditions of Theorem 2.3.1 have to be verified at each iteration. Therefore, computing a smaller set $\mathbf{IS} \subseteq \mathbf{IS}_{tot}$ such that $\mathbf{IS} \in \mathbf{SIS}^{F^i}$ and applying Procedure 5.1 to $\mathbf{IS}$ (instead of $\mathbf{IS}_{tot}$) reduces the computational complexity. This would be a *bottom-up* approach.

In this chapter, we investigate the problem of isolator selection by developing a *bottom-up* algorithm for finding a set $\mathbf{IS} \subseteq \mathbf{IS}_{tot}$ such that $\mathbf{IS} \in \mathbf{SIS}^{F^i}$. A minimal set of isolators is calculated from $\mathbf{IS}$ by using the top-down Procedure 5.1. In this algorithm, diagnosability of a failure mode is initially investigated based on the DES abstract model. The isolator selection procedure is based on the diagnosability at the DES level. If the information gathered from the discrete outputs of the system is not sufficient for diagnosis of a failure mode, appropriate isolators are selected to make that failure mode diagnosable in the hybrid system.

Fault diagnosis in our work is to determine the system's condition (normal/ faulty), and if the condition is faulty, specifying the failure modes present in the system's condition. The system's condition and the failure modes present in the system's condition can be determined if the discrete state of a failure mode can be distinguished from the rest of states. Therefore, a failure mode F^i can be diagnosed if the discrete state of the system after the occurrence of F^i can be distinguished from any state $q \in Q - Q_{\mathcal{F}^i}$ based on the output sequence (obtained from discrete sensors and isolators). For example, in Figure 5.1, the failure mode F can be diagnosed if the discrete state q_1 can be distinguished from the discrete state q_1', or the discrete state q_2 can be distinguished from the discrete state q_2'. In the next section, we provide a systematic method for computing a minimal set of isolators that can distinguish a set of discrete states from another set.

5.2 Minimal Discrete State Distinguishers

In our framework, a bank of isolators is constructed for each set of EM-similar discrete states. As described in Chapter 3, the function *active*(q) can be used for determining the fault types active in a discrete state q. We use this mapping to develop a systematic method for computing a minimal set of isolators that can distinguish a set of discrete states from another set of discrete states.

Let Q^{EM} be a set of EM-similar discrete states, $FT^{Q^{EM}}$ be the set of fault types present in Q^{EM}, and $\mathbf{IS}^{Q^{EM}}$ be the set of all isolators that can be designed for Q^{EM}. As described in Chapter 3, an isolator $Is^{Q^{EM}}(\Phi) \in \mathbf{IS}^{Q^{EM}}$ isolates the set of fault types Φ from the rest of fault types in $FT^{Q^{EM}}$. The isolator $Is^{Q^{EM}}(\Phi)$

is sensitive to fault types of Φ and insensitive to the fault types of $(FT^{Q^{EM}} - \Phi)$. Recall that the isolator $Is^{Q^{EM}}(\Phi)$ generates zero output only if the system is in a discrete state of Q^{EM} where no fault type of Φ is active. Therefore, the output generated by an isolator can be considered as a function of the discrete state of the system. We associate a binary function with any isolator $Is^{Q^{EM}}$ designed for Q^{EM} as follows.

Definition 5.2.1. *The function $\Gamma^{Is^{Q^{EM}}(\Phi)} : Q \longrightarrow \{0, 1\}$ is defined as*

$$\Gamma^{Is^{Q^{EM}}(\Phi)}(q) = \begin{cases} 1 \text{ if } q \notin Q^{EM} \\ 1 \text{ if } q \in Q^{EM} \text{ and } active(q) \bigcap \Phi \neq 0 \\ 0 \text{ if } q \in Q^{EM} \text{ and } active(q) \bigcap \Phi = 0 \end{cases}$$

■

Let $\mathbf{IS} = \{Is^1, \cdots, Is^l\} \subseteq \mathbf{IS}_{tot}$ be a set of isolators that can be designed for the system. Function $\Gamma^{\mathbf{IS}}$ associated with a set of isolators **IS** defined as

$$\Gamma^{\mathbf{IS}}(q) = \Gamma^{Is^1}(q) \times \cdots \times \Gamma^{Is^l}(q)$$

gives the output of the isolators at q.

Two discrete states are called distinguishable if by using the outputs of the isolators, one can identify which of the discrete states the system is in. In other words, two discrete states are distinguishable if there exists an isolator that generates different outputs for each of the two discrete states.

Definition 5.2.2. *Two discrete states $q, q' \in Q$ are called **distinguishable** from each other (by using $\mathbf{IS}_{tot}$) if there exists an isolator $Is \in \mathbf{IS}_{tot}$ such that*

$$\Gamma^{Is}(q) \neq \Gamma^{Is}(q')$$

■

Alternatively, q and q' are distinguishable if

$$q \not\equiv q' \; mod \; ker \; \Gamma^{\mathbf{IS}}$$

where $ker \; \Gamma^{\mathbf{IS}}$ is the equivalence kernel of $\Gamma^{\mathbf{IS}}$.

Given $q, q' \in Q^{EM}$, q and q' are distinguishable if there exists an isolator $Is^{Q^{EM}}(\Phi) \in \mathbf{IS}^{Q^{EM}}$ such that $\Phi \bigcap active(q) \neq \emptyset$ and $\Phi \bigcap active(q') = \emptyset$, or $\Phi \bigcap active(q') \neq \emptyset$ and $\Phi \bigcap active(q) = \emptyset$. Given $q \in Q^{EM}$ and $q' \in Q^{E'M'}$ and $Q^{EM} \neq Q^{E'M'}$, q and q' are distinguishable if there exists an isolator $Is^{Q^{EM}}(\Phi) \in \mathbf{IS}^{Q^{EM}}$ such that $\Phi \subseteq FT^{Q^{EM}}$ and $active(q) \bigcap \Phi = \emptyset$, or there exists an isolator $Is^{Q^{E'M'}}(\Phi') \in \mathbf{IS}^{Q^{E'M'}}$ such that $\Phi' \subseteq FT^{Q^{E'M'}}$ and $active(q') \bigcap \Phi = \emptyset$.

Example 5.2.1. *Consider a hybrid automaton with the set of fault types* $FT = \{f^1, f^2, f^3, f^4, f^5, f^6\}$*. Suppose the EM-similar state sets are* $Q^{E^1M^1} = \{q_0, q_1, q_2\}$, $Q^{E^2M^2} = \{q_3, q_4\}$ *and* $Q^{E^3M^3} = \{q_5, q_6\}$ *with* $active(q_0) = \{f^1, f^2, f^4\}$, $active(q_1) = \{f^1, f^2\}$, $active(q_2) = \{f^1, f^3, f^4\}$, $active(q_3) = \{f^1, f^5\}$, $active(q_4) = \{f^2, f^5\}$, $active(q_5) = \{f^1, f^4, f^6\}$ *and* $active(q_6) = \{f^2, f^5, f^6\}$*. Consider a set of isolators* $\mathbf{IS}_{tot} = \{Is^{Q^{E^1M^1}}(\{f^1\}), Is^{Q^{E^1M^1}}(\{f^2\}), Is^{Q^{E^1M^1}}(\{f^3\}), Is^{Q^{E^1M^1}}(\{f^2, f^4\}), Is^{Q^{E^1M^1}}(\{f^3, f^4\}), Is^{Q^{E^2M^2}}(\{f^1, f^2\}), Is^{Q^{E^2M^2}}(\{f^5\}), Is^{Q^{E^3M^3}}(\{f^4\}), Is^{Q^{E^3M^3}}(\{f^1, f^6\}), Is^{Q^{E^3M^3}}(\{f^1, f^2, f^6\})\}$*. It can be verified that* q_0 *and* q_1 *are not distinguishable. The discrete states* q_1 *and* q_2 *are distinguishable because* $\Gamma^{Is^{Q^{E^1M^1}}(\{f^3\})}(q_1) = 0$ *but* $\Gamma^{Is^{Q^{E^1M^1}}(\{f^3\})}(q_2) = 1$*. The discrete states* q_0 *and* q_2 *are also distinguishable because* $\Gamma^{Is^{Q^{E^1M^1}}(\{f^3\})}(q_0) = 0$ *but* $\Gamma^{Is^{Q^{E^1M^1}}(\{f^3\})}(q_2) = 1$*. The discrete states* q_0 *and* q_1 *are also distinguishable from* q_3, q_4, q_5 *and* q_6 *because* $\Gamma^{Is^{Q^{E^1M^1}}(\{f^3\})}(q_3) = \Gamma^{Is^{Q^{E^1M^1}}(\{f^3\})}(q_4) = \Gamma^{Is^{Q^{E^1M^1}}(\{f^3\})}(q_1) = \Gamma^{Is^{Q^{E^1M^1}}(\{f^3\})}(q_1) = 1$*. The discrete state* q_2 *is distinguishable from* q_6 *because* $\Gamma^{Is^{Q^{E^3M^3}}(\{f^4\})}(q_6) = 0$ *but* $\Gamma^{Is^{Q^{E^3M^3}}(\{f^4\})}(q_2) = 1$.

It can be verified that q_2 is distinguishable from q_3, q_4 and q_5. Moreover, it can be verified that q_3 is not distinguishable from q_2, q_4 and q_5, and q_4 is not distinguishable from q_2, q_3 and q_5. ■

The definition of distinguishable discrete states can be extended to discrete state sets as follows.

Let $P, P' \subseteq Q$ be two discrete state sets. We say P and P' are distinguishable from each other if for any $q \in P$ and $q' \in P'$, q and q' are distinguishable from each other. Let $\mathbf{IS} \subseteq \mathbf{IS}_{tot}$ be the set of isolators such that P and P' are distinguishable from each other by using the isolators of $\mathbf{IS}$. The isolator set $\mathbf{IS}$ is called a **P|P′-distinguisher**.

Let $\mathbf{SDS}(P|P')$ denote the set of all $P|P'$-distinguishers.

Definition 5.2.3. *Let the isolator set* $\mathbf{IS}$ *be a $P|P'$-distinguisher, i.e,* $\mathbf{IS} \in \mathbf{SDS}(P|P')$. *An isolator set* $\mathbf{IS}$ *is called a* ***minimal*** **P|P′*-distinguisher*** *if none of the proper subsets of* $\mathbf{IS}$ *is a $P|P'$-distinguisher.* ■

Lemma 5.2.1. *Let $P, P \subseteq Q$ be two sets of discrete states. Also let $V \subseteq P$ and $V' \subseteq P'$. If* $\mathbf{IS} \in \mathbf{SDS}(P|P')$, *then* $\mathbf{IS} \in \mathbf{SDS}(V|V')$.

Proof - Let $\mathbf{IS} \in \mathbf{SDS}(P|P')$. Suppose $\mathbf{IS} \notin \mathbf{SDS}(V|V')$. This implies that there are states $q \in V$ and $q' \in V'$ such that $\Gamma^{\mathbf{IS}}(q) = \Gamma^{\mathbf{IS}}(q')$. Therefore, $\mathbf{IS} \notin \mathbf{SDS}(P|P')$ which contradicts the assumption. Thus $\mathbf{IS} \in \mathbf{SDS}\ (V|V')$. ■

Theorem 5.2.2. *Let $P, P' \subseteq Q$.*

$$\mathbf{SDS}(P|P') = \bigcap_{q \in P} \bigcap_{q' \in P'} \mathbf{SDS}(\{q\}|\{q'\})$$

Proof - Suppose $\mathbf{IS} \in \mathbf{SDS}(P|P')$. According to Lemma 5.2.1, we have $\mathbf{IS} \in \mathbf{SDS}(\{q\}|\{q'\})$ for any $q \in P$ and $q' \in P'$. Therefore, $\mathbf{IS} \in \bigcap_{q \in P} \bigcap_{q' \in P'} \mathbf{SDS}(\{q\}|\{q'\})$. Now, let $\mathbf{IS} \in \bigcap_{q \in P} \bigcap_{q' \in P'} \mathbf{SDS}(\{q\}|\{q'\})$. This implies that for any $q \in P$ and

$q' \in P'$, $\Gamma^{\mathbf{IS}}(q) \neq \Gamma^{\mathbf{IS}}(q')$ and by Definition 5.2.2, $\mathbf{IS} \in \mathbf{SDS}(\{q\}|\{q'\})$. Therefore, $\mathbf{IS} \in \mathbf{SDS}(P|P')$. As a result, $\mathbf{SDS}(P|P') = \bigcap_{q \in P} \bigcap_{q' \in P'} \mathbf{SDS}(\{q\}|\{q'\})$. ■

Obviously, if $P \bigcap P' \neq \emptyset$, then $\mathbf{SDS}(P|P') = \emptyset$.

Example 5.2.2. *(Example 5.2.1 Continued): We want to calculate*

$$\mathbf{SDS}(\{q_0, q_1\}|\{q_2, q_4\})$$

It can be verified that

$\mathbf{SDS}(\{q_0\}|\{q_2\}) = \{Is^{Q^{E^1M^1}}(\{f^2\}), Is^{Q^{E^1M^1}}(\{f^3\})\}$

$\mathbf{SDS}(\{q_0\}|\{q_4\}) = \{Is^{Q^{E^1M^1}}(\{f^3\})\}$

$\mathbf{SDS}(\{q_1\}|\{q_2\}) = \{Is^{Q^{E^1M^1}}(\{f^2\}), Is^{Q^{E^1M^1}}(\{f^3\}), Is^{Q^{E^1M^1}}(\{f^3, f^4\})\}$ *and*

$\mathbf{SDS}(\{q_1\}|\{q_4\}) = \{Is^{Q^{E^1M^1}}(\{f^3\}), Is^{Q^{E^1M^1}}(\{f^3, f^4\})\}$

Therefore, by Theorem 5.2.2, we have

$$\mathbf{SDS}(\{q_0, q_1\}|\{q_2, q_4\})\{Is^{Q^{E^1M^1}}(\{f^3\})\}$$

■

In the following, we develop a "DES-first" approach for isolator selection in hybrid automata.

5.3 DES-first Approach for Isolator Selection in Hybrid Automata

The problem of isolator selection is to choose a minimal set of isolators such that a failure mode remains diagnosable in the EDESA of a hybrid system and isolators. In this section, we develop a "DES-first" approach for isolator selection in our framework. In this approach, the diagnosability of failure modes is initially investigated using discrete outputs in the DES model. Isolators are used if the diagnosability of a failure mode cannot be attained by using the information from the discrete sensors. The isolators are selected so as to resolve the ambiguity in the DES level that has made a failure mode undiagnosable. Theorem 5.3.1 provides guidelines for isolator selection in the DES-first approach.

Let $q \in Q_{\mathcal{F}^i}$. The state set $Amb(q)$ denotes the set of discrete states of $Q - Q_{\mathcal{F}^i}$ from which the system can generate an infinite output sequence identical to one generated from q, that is

$$\begin{aligned} Amb(q) = \{q' | q' \in Q - Q_{\mathcal{F}^i} \text{ and } \\ \{s | s \in L_o(H_{abs,N,\overline{\mathcal{F}^i}_i}, q') \bigcap L_o(H_{abs,\mathcal{F}^i}, q), |s| \geq |Q|^2\} \neq \emptyset\} \end{aligned}$$

Theorem 5.3.1. *Assume that $z_0 = \tilde{Q}$. For an isolator set* **IS**, *a permanent failure mode F^i will be diagnosable in $\tilde{H}(\mathbf{IS})$ if and only if:*

1. *For any $q \in Q_{\mathcal{F}^i}$ such that $q \in Q^{inf}$, if $\lambda^{-1}(\lambda(q)) \bigcap (Q - Q_{\mathcal{F}^i}) \neq \emptyset$, then*

$$\mathbf{IS} \in \mathbf{SDS}(\{q\} | \lambda^{-1}(\lambda(q)) \bigcap (Q - Q_{\mathcal{F}^i}))$$

2. *For any cycle of discrete states $Q^c_{\mathcal{F}^i}$ in $Q_{\mathcal{F}^i}$ consisting of states having the same discrete output in H_{abs}, say d, if $\lambda^{-1}(d) \bigcap (Q - Q_{\mathcal{F}^i}) \neq \emptyset$, then there exists $Q_1 \subseteq Q^c_{\mathcal{F}^i}$ such that for any $q_j \in Q_1$ there exists $Q_j \subseteq \lambda^{-1}(d) \bigcap (Q - Q_{\mathcal{F}^i})$ and $\mathbf{IS} \in \mathbf{SDS}(\{q_j\} | Q_j)$ and $\bigcup_{j=1}^{|Q_1|} Q_j = \lambda^{-1}(d) \bigcap (Q - Q_{\mathcal{F}^i})$;*
3. *For any cycle of discrete states $Q^c_{\mathcal{F}^i}$ in $Q_{\mathcal{F}^i}$, if there exists a cycle of discrete states $Q^c_{N,\overline{\mathcal{F}^i}} \subseteq (Q_N \bigcup Q_{\overline{\mathcal{F}^i}_i})$ generating the same unbounded discrete output sequence in H_{abs}, then there exist $q \in Q^c_{\mathcal{F}^i}$ and $q' \in Q^c_{N,\overline{\mathcal{F}^i}}$ and $q' \in Amb(q)$ such that $\mathbf{IS} \in \mathbf{SDS}(\{q\} | \{q'\})$.*

Proof: (If part)- Conditions (1) guarantees that after F^i occurs, if the system remains indefinitely in a discrete state $q \in Q^{inf}$, the isolator set **IS** generates new output symbols so that q can be distinguished from any $q' \in Q - Q_{\mathcal{F}^i}$ having the same discrete output as q. The output generated by the isolators in the isolator set **IS** (integrated with those of discrete sensors) will be F^i-indicative. Conditions (2) guarantees that after F^i occurs, if the system remains in a cycle of discrete states with the same discrete outputs, the isolator set **IS** can distinguish any $q' \in Q - Q_{\mathcal{F}^i}$ having the same discrete output from some state of that cycle. Therefore, as the system evolves in the cycle, outputs of the isolator set **IS** isolate F^i. Condition (3) guarantees that after F^i occurs, if the system enters a cycle of discrete states $Q^c_{\mathcal{F}^i}$ generating an unbounded sequence of outputs that can be also generated by a cycle of discrete states $Q^c_{N,\overline{\mathcal{F}^i}}$ in $Q - Q_{\mathcal{F}^i}$, then there exists a discrete state $q \in Q^c_{\mathcal{F}^i}$ such that the isolator set **IS** can distinguish q from one $q' \in Amb(q)$. When the system is in $Q^c_{\mathcal{F}^i}$, the sequence of discrete outputs generated by the isolator set will be different from those generated in $Q^c_{N,\overline{\mathcal{F}^i}}$. Hence, faulty behavior $\mathcal{F}^i$ will be isolated.

(Only if part) Suppose either conditions (1) or (2) does not hold. After F^i occurs, the system may stay indefinitely in a discrete state $q \in Q^{inf}$ or a cycle of faulty discrete states whose discrete output is not F^i-indicative. If the diagnoser is initialized after this last output and the isolator set **IS** does not generate F^i-indicative outputs, then F^i will be undiagnosable in $\tilde{H}$. Condition (3) can be also proven to be necessary by using a similar discussion. ■

In the following, by using the results of Theorem 5.3.1, we develop a procedure to compute a set of isolators that makes a failure mode F^i diagnosable. The output of the procedure is a minimal isolator set $\mathbf{IS}^{F^i}_{min}$ that makes F^i diagnosable in $\tilde{H}(\mathbf{IS})$. First we develop a bottom-up algorithm to compute a set $\mathbf{IS}^{F^i}$ that makes F^i diagnosable in $\tilde{H}(\mathbf{IS})$. The set $\mathbf{IS}^{F^i}$ may not be minimal. Thus, by using the top-down Procedure 5.1, we can calculate a minimal isolator set.

Initially, $\mathbf{IS}^{F^i} = \emptyset$. First, we add enough isolators to $\mathbf{IS}^{F^i}$ to satisfy condition

(1) in Theorem 5.3.1. For any states $q \in Q_{\mathcal{F}^i} \bigcap Q^{inf}$ and $q' \in Q - Q_{\mathcal{F}^i}$ such that $\lambda(q) = \lambda(q')$, we compute a $q|q'$-distinguisher. If $q|q'$-distinguisher does not exists, F^i remains undiagnosable in $\tilde{H}(\mathbf{IS})$ for any $\mathbf{IS} \subseteq \mathbf{IS}_{tot}$. If $q|q'$-distinguisher exists, we add it to $\mathbf{IS}^{F^i}$.

Secondly, we add enough isolators to $\mathbf{IS}^{F^i}$ to satisfy condition (2) in Theorem 5.3.1. Let $Q^c_{\mathcal{F}^i} \subseteq Q_{\mathcal{F}^i}$ be a set of discrete states having the same output that make a cycle in $Q_{\mathcal{F}^i}$, and $\lambda(Q^c_{\mathcal{F}^i})$ is not F^i-indicative. Also let $Q^C_{\mathcal{F}^i}$ be the set of all these sets. For any $Q^c_{\mathcal{F}^i} \in Q^C_{\mathcal{F}^i}$, we compute a set of isolators to distinguish one of the states $q \in Q^c_{\mathcal{F}^i}$ from a state $q' \in Q - Q_{\mathcal{F}^i}$ having the same output. If a $q|q'$-distinguisher exists, we add this set to $\mathbf{IS}^{F^i}$. If there exists a cycle $Q^c_{\mathcal{F}^i}$ such that for all $q \in Q^c_{\mathcal{F}^i}$ no $q|q'$-distinguisher exists, F^i remains undiagnosable.

Thirdly, we add enough isolators to $\mathbf{IS}^{F^i}$ to satisfy condition (3) in Theorem 5.3.1. Let $Q^o_{\mathcal{F}^i} \subseteq Q_{\mathcal{F}^i}$ be a set of discrete states that make a cycle in $Q_{\mathcal{F}^i}$ and there exists $q' \in Q - Q_{\mathcal{F}^i}$ such that $q' \in Amb(q)$ for some $q \in Q^o_{\mathcal{F}^i}$. Also let $Q^O_{\mathcal{F}^i}$ be the set of all these state sets. We compute a set of isolators to distinguish one of the states $q \in Q^o_{\mathcal{F}^i}$ from the state $q' \in Amb(q)$. If a $q|q'$-distinguisher exists, we add this set to $\mathbf{IS}^{F^i}$. If there exists a cycle $Q^o_{\mathcal{F}^i}$ such that for all $q \in Q^o_{\mathcal{F}^i}$ no $q|q'$-distinguisher exists, F^i remains undiagnosable. Finally, by using Procedure 5.1, we compute a minimal set $\mathbf{IS}^{F^i}_{min}$.

Let $Q^o_{\overline{\mathcal{F}^i}} \subseteq Q - Q_{\mathcal{F}^i}$ be a set of discrete states that make a cycle in $Q - Q_{\mathcal{F}^i}$ and there exists $q \in Q_{\mathcal{F}^i}$ such that $q' \in Amb(q)$ for some $q' \in Q^o_{\overline{\mathcal{F}^i}}$, and let $Q^O_{\overline{\mathcal{F}^i}}$ be the set of all these state sets. We are now in a position to formally present our procedure.

```
Procedure 5.2: Given sets $Q^C_{\mathcal{F}^i}$, $Q^O_{\mathcal{F}^i}$ and $Q^O_{\overline{\mathcal{F}^i}}$
    Call Initialization
    Call VerifyCon1
    Call VerifyCon2
    Call VerifyCon3
    Call Top_downProc
End (Procedure)
Procedure Initialization
    $\mathbf{IS}^{F^i}_{Min} = \emptyset$, $\mathbf{IS}^{F^i} = \emptyset$
End (Procedure Initialization)
Procedure VerifyCon1
    For all $q' \in Q - Q_{\mathcal{F}^i}$
        For all $q \in Q_{\mathcal{F}^i}$
            If $\lambda(q) = \lambda(q')$ and $q \in Q^{inf}$
                Compute $\mathbf{SDS}(\{q\}|\{q'\})$
                If $\mathbf{SDS}(\{q\}|\{q'\}) = \emptyset$
                    $F^i$ is not diagnosable
                    End of Procedure
                END (If)
                $\mathbf{IS}^{F^i} = \mathbf{IS}^{F^i} \bigcup \mathbf{IS}$ for one $\mathbf{IS} \in \mathbf{SDS}(\{q\}|\{q'\})$
            END (If)
        End (For)
    End (For)
End (Procedure VerifyCon1)
Procedure VerifyCon2
    For all $q' \in Q - Q_{\mathcal{F}^i}$
        $Q(q') = \{Q^c_{\mathcal{F}^i} \mid Q^c_{\mathcal{F}^i} \in Q^C_{\mathcal{F}^i}$ and $\lambda(q') = \lambda(Q^c_{\mathcal{F}^i})\}$
        For all $q \in Q_{\mathcal{F}^i}$
            If $\lambda(q) = \lambda(q')$ and $q \in Q^c_{\mathcal{F}^i}$ for some $Q^c_{\mathcal{F}^i} \in Q^C_{\mathcal{F}^i}$
                Compute $\mathbf{SDS}(\{q\}|\{q'\})$
                $\mathbf{IS}^{F^i} = \mathbf{IS}^{F^i} \bigcup \mathbf{IS}$ for one $\mathbf{IS} \in \mathbf{SDS}(\{q\}|\{q'\})$
                $Q(q') = Q(q') - \{Q^c_{\mathcal{F}^i} \mid Q^c_{\mathcal{F}^i} \in Q^C_{\mathcal{F}^i}$ and $q \in Q^c_{\mathcal{F}^i}\}$
            End (If)
        End (For)
        If $Q(q') \neq \emptyset$
            $F^i$ is not diagnosable
            End of Procedure
        END (If)
        End (For)
    End (Procedure VerifyCon2)
    Procedure VerifyCon3
        For all $q' \in Q - Q_{\mathcal{F}^i}$
            For all $q \in Q_{\mathcal{F}^i}$
                If $\lambda(q) = \lambda(q')$ and $q' \in Amb(q)$ and $\{Q^{o}_{\overline{\mathcal{F}^i}} \mid Q^{o}_{\overline{\mathcal{F}^i}} \in Q^{O}_{\overline{\mathcal{F}^i}}$ and $q' \in Q^{o}_{\overline{\mathcal{F}^i}}\} \neq \emptyset$
                    Compute $\mathbf{SDS}(\{q\}|\{q'\})$
                    $\mathbf{IS}^{F^i} = \mathbf{IS}^{F^i} \bigcup \mathbf{IS}$ for one $\mathbf{IS} \in \mathbf{SDS}(\{q\}|\{q'\})$
                    $Q^{O}_{\overline{\mathcal{F}^i}} = Q^{O}_{\overline{\mathcal{F}^i}} - \{Q^{o}_{\overline{\mathcal{F}^i}} \mid q' \in Q^{o}_{\overline{\mathcal{F}^i}}\}$
                End (If)
            End (For)
        End (For)
        If $Q^{O}_{\overline{\mathcal{F}^i}} \neq \emptyset$
            $F^i$ is not diagnosable
            End of Procedure
        END (If)
    End (Procedure VerifyCon3)
    Procedure Top_downProc
        For all $Is \in \mathbf{IS}^{F^i}$
            $\mathbf{IS}^{F^i} = \mathbf{IS}^{F^i} - \{Is\}$
            Check the conditions of Theorem 2.3.1 for $\tilde{H}(\mathbf{IS}^{F^i})$
            If $\mathbf{IS}^{F^i} \notin \mathbf{SIS}^{F^i}$ (conditions of Theorem 2.3.1 fail)
                $\mathbf{IS}^{F^i} = \mathbf{IS}^{F^i} \bigcup \{Is\}$
            End (If)
        End (For)
        $\mathbf{IS}^{F^i}_{min} = \mathbf{IS}^{F^i}$
    End (Procedure Top_downProc)
```

Remark 5.3.1. *In Theorem 5.3.1 and Procedure 5.2, we need to verify if for two states $q \in Q_{\mathcal{F}^i}$ and $q' \in Q - Q_{\mathcal{F}^i}$, we have $q' \in Amb(q)$. In the following, we briefly explain a method for verifying if $q' \in Amb(q)$. Details of the method can be found in [90].*

Let $\overline{H}_{abs}$ be the Reachability Transition System (RTS) of the DES abstraction of the hybrid system. Also let $\overline{H}_{abs}^{\mathcal{F}^i}(q)$ denote the reachable sub-generator of $\overline{H}_{abs}$ corresponding to the states of $Q_{\mathcal{F}^i}$ with the initial state q, and $\overline{H}_{abs}^{N,\overline{\mathcal{F}^i}}(q')$ denote the reachable sub-generator of $\overline{H}_{abs}$ corresponding to the states of $Q - Q_{\mathcal{F}^i}$ with the initial state q'. We need to find output cycles that are common to $\overline{H}_{abs}^{\mathcal{F}^i}(q)$ and $\overline{H}_{abs}^{N,\overline{\mathcal{F}^i}}(q')$. One way to find common output sequences, and thus common cycles, is to first convert $\overline{H}_{abs}$ to an equivalent mealy generator $M\_\overline{H}_{abs}$ in which output changes in $\overline{H}_{abs}$ are represented as transitions. Let $M\_\overline{H}_{abs}^{\mathcal{F}^i}(q)$ and $M\_\overline{H}_{abs}^{N,\overline{\mathcal{F}^i}}(q')$ be the sub-generators of $M\_\overline{H}_{abs}$ corresponding to $\overline{H}_{abs}^{\mathcal{F}^i}(q)$ and $\overline{H}_{abs}^{N,\overline{\mathcal{F}^i}}(q')$, respectively. It can be verified that $\overline{H}_{abs}^{\mathcal{F}^i}(q)$ and $\overline{H}_{abs}^{N,\overline{\mathcal{F}^i}}(q')$ have common cycles (i.e., $q' \in Amb(q)$) if and only if there is a cycle in $\mathbf{meet}(M\_\overline{H}_{abs}^{\mathcal{F}^i}(q), M\_\overline{H}_{abs}^{N,\overline{\mathcal{F}^i}}(q'))$. ■

Example 5.3.1. *Figure 5.2 shows the DES abstraction of a hybrid automaton H with three fault types f^1, f^2 and f^3 and the corresponding failure modes F^1, F^2 and F^3, respectively. The occurrence of the failure modes F^1, F^2 and F^3 are modeled by transitions labeled with the events $\hat{f}^1$, $\hat{f}^2$ and $\hat{f}^3$, respectively. The unobservable event 'u_1' changes the discrete state of the system in normal mode. The labels 'u_2' and 'u_3' are unobservable events changing the discrete state of the system in the failure mode F^2 and create a cycle of states with the same discrete output. The set of discrete states is $Q = \{q_0, \cdots, q_{17}\}$ and $Q^{inf} = \{q_{14}, q_{17}\}$. The condition set of the system is $\mathcal{K} = \{N, F^1, F^2, F^3, F^{2,3}\}$. The discrete states of the system can be partitioned according to the condition of the system. Here, $Q_N = \{q_0, q_1, q_2, q_6, q_7\}$, $Q_{F^1} = \{q_3, q_8, q_{12}\}$, $Q_{F^2} = \{q_4, q_9, q_{13}, q_{16}\}$, $Q_{F^3} = \{q_5, q_{11}, q_{15}, q_{17}\}$ and $Q_{F^{2,3}} = \{q_{10}, q_{14}\}$. We also have $Q_{\mathcal{F}^1} = Q_{F^1}$, $Q_{\mathcal{F}^2} = \{q_4, q_9, q_{10}, q_{13}, q_{14}, q_{16}\}$ and $Q_{\mathcal{F}^3} = \{q_5, q_{10}, q_{11}, q_{14}, q_{15}, q_{17}\}$. Moreover, $Q^{E_1M_1} = \{q_0, q_3, q_4, q_5\}$, $Q^{E_2M_2} = \{q_2, q_8, q_9, q_{10}, q_{11}\}$, $Q^{E_3M_3} = \{q_7, q_{13}, q_{15}\}$ and $Q^{E_4M_4} = \{q_{14}, q_{17}\}$. The rest of the discrete states have different (E, M) functions. Fault type f^1 is active in the discrete states q_3 and q_8, f^2 is active in q_4, q_9, q_{10}, q_{13} and q_{14}, and f^3 is active*

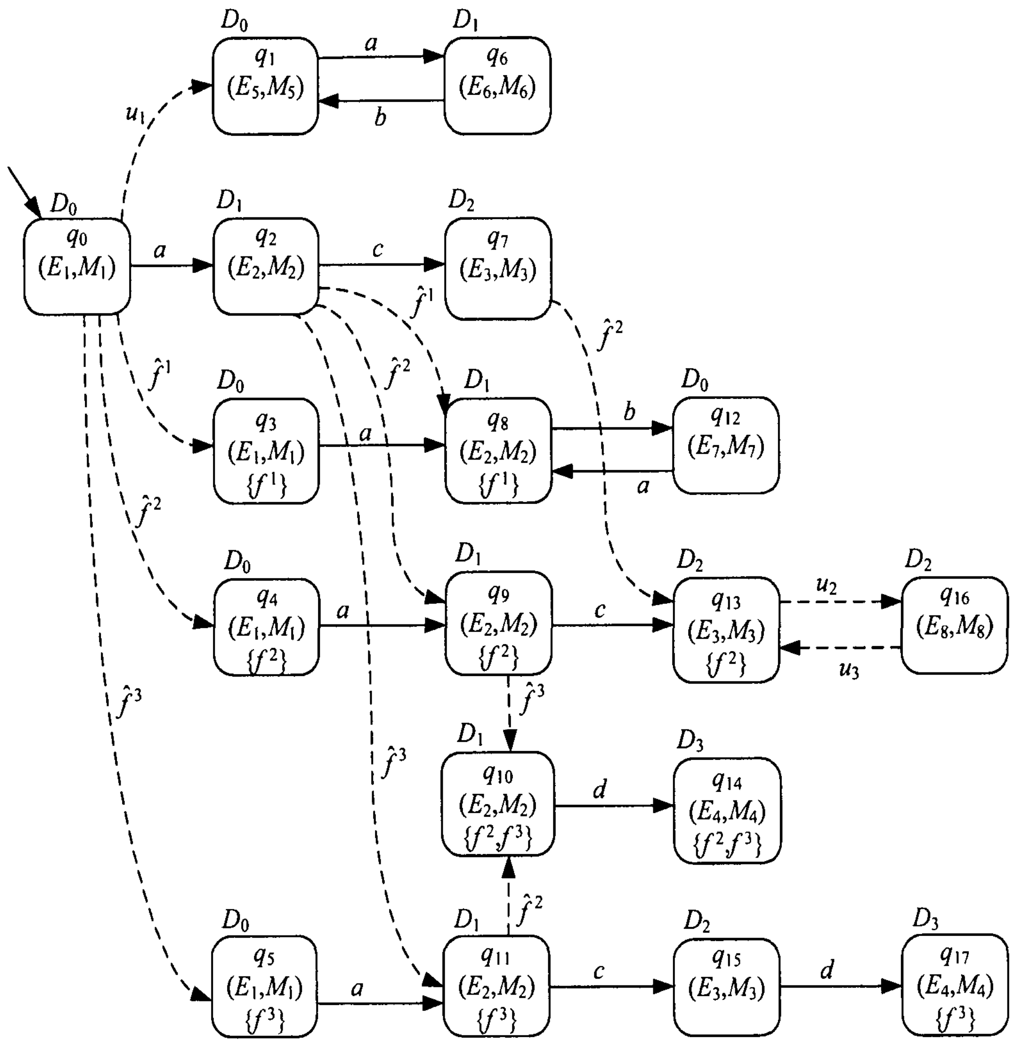


Figure 5.2: DES abstraction of a hybrid automaton with three failure modes.

in $q_5, q_{10}, q_{11}, q_{14}$ and q_{17}. Assume that the following set of isolators can be designed for the system, namely: $\mathbf{IS}_{tot} = \{Is^{Q^{E_1M_1}}(\{f^1, f^3\}),\ Is^{Q^{E_1M_1}}(\{f^1, f^2, f^3\}),\ Is^{Q^{E_2M_2}}(\{f^3\}),\ Is^{Q^{E_2M_2}}(\{f^1, f^2\}),\ Is^{Q^{E_2M_2}}(\{f^2, f^3\}),\ Is^{Q^{E_2M_2}}(\{f^1, f^2, f^3\}),\ Is^{Q^{E_3M_3}}(\{f^2\}),\ Is^{Q^{E_4M_4}}(\{f^2\}),\ Is^{Q^{E_4M_4}}(\{f^3\}),\ Is^{Q^{E_4M_4}}(\{f^2, f^3\})\}$*. The set of discrete outputs of the system is: $D = \{D_0, D_1, D_2, D_3\}$. Let z_0 be the initial state of the diagnoser designed for H_{abs}. Assuming that $z_0 = Q$, none of the failure modes are diagnosable using the diagnoser designed for H_{abs}. Failure mode F^1 is not diagnosable because it violates condition (3) of Theorem 2.3.1, F^2 violates condition (1) and condition (2), and F^3 violates condition (1) of that theorem.*

In order to make F^1 diagnosable, we need to use isolators that make changes in the output sequence generated by the system when the system enters the cycle of states q_8 and q_{12} to distinguish the failure mode F^1 from the normal condition. Thus we need a $q_8|q_6$-distinguisher or a $q_{12}|q_1$-distinguisher to make F^1 diagnosable. There is no isolator designed based on the dynamics of the system at q_1, q_6 and q_{12}. We have $\Gamma^{Is^{Q^{E_2M_2}}(\{f^2,f^3\})}(q_8) = 0$, $\Gamma^{Is^{Q^{E_2M_2}}(\{f^2,f^3\})}(q_6) = 1$, $\Gamma^{Is^{Q^{E_2M_2}}(\{f^3\})}(q_8) = 0$ and $\Gamma^{Is^{Q^{E_2M_2}}(\{f^3\})}(q_6) = 1$. Therefore, $\mathbf{SDS}(\{q_8\}|\{q_6\}) = \{\{Is^{Q^{E_2M_2}}(\{f^2, f^3\})\}, \{Is^{Q^{E_2M_2}}(\{f^3\})\}, \{Is^{Q^{E_2M_2}}(\{f^2, f^3\}), Is^{Q^{E_2M_2}}(\{f^3\})\}\}$*. We also have* $\mathbf{SDS}(\{q_{12}\}|\{q_1\}) = \emptyset$*. Therefore,* $\mathbf{SIS}^{F^1} = \{\{Is^{Q^{E_2M_2}}(\{f^2, f^3\})\}, \{Is^{Q^{E_2M_2}}(\{f^3\})\}, \{Is^{Q^{E_2M_2}}(\{f^2, f^3\}), Is^{Q^{E_2M_2}}(\{f^3\})\}\}$*, and $\{Is^{Q^{E_2M_2}}(\{f^2, f^3\})\}$ and $\{Is^{Q^{E_2M_2}}(\{f^3\})\}$ are the minimal isolator sets for the diagnosability of F^1.*

The failure mode F^2 is not diagnosable in the diagnoser of H_{abs} because the discrete output generated by the system in the cycle made by the states q_{13} and q_{16} is not F^2-indicative. For making F^2 diagnosable, we need a set of isolators to generate appropriate discrete outputs when the system is in the discrete states q_{13} and q_{16} to distinguish F^2 from F^3 and the normal condition. There is no isolator designed based on the dynamics of the system at q_{17}. Therefore, we need a $q_{13}|\{q_7, q_{15}\}$-distinguisher. We have $q_{13}, q_{15}, q_7 \in Q^{E_3M_3}$, and therefore according *to Theorem 5.2.2,* $\mathbf{SDS}(\{q_{13}\}|\{q_7, q_{15}\}) = \mathbf{SDS}(\{q_{13}\}|\{q_7\}) \bigcap \mathbf{SDS}(\{q_{13}\}|\{q_{15}\}) = \{\{Is^{Q^{E_3M_3}}(\{f^2\})\}\}$. *Moreover,* $q_{14} \in Q^{inf}$ *but the discrete output generated by the system in* q_{14} *is not* F^2*-indicative. The system generates the same output when only* F^3 *has occurred and the system is in* q_{17}. *We need to use an isolator that generates appropriate discrete outputs when the system is in* q_{14} *to distinguish* q_{14} *from* q_{17}. *We can verify that* $\mathbf{SDS}(\{q_{14}\}|\{q_{17}\}) = \{\{Is^{Q^{E_4M_4}}(\{f^2\})\}\}$. *Therefore,* $\mathbf{SIS}^{F^2} = \{\{Is^{Q^{E_3M_3}}(\{f^2\}), Is^{Q^{E_4M_4}}(\{f^2\})\}\}$, *and* $\{Is^{Q^{E_3M_3}}(\{f^2\}),$ $Is^{Q^{E_4M_4}}(\{f^2\})\}$ *is the minimal isolator set for the diagnosability of* F^2.

We observe that $q_{17} \in Q^{inf}$ *but the discrete output generated by the system in* q_{17} *is not* F^3*-indicative. We need to use* $q_{14}|q_{17}$*-distinguisher. As explained above, the isolator* $Is^{Q^{E_4M_4}}(\{f^2\})$ *can distinguish* q_{14} *from* q_{17}. *Therefore,* $\mathbf{SIS}^{F^3} = \{\{Is^{Q^{E_4M_4}}(\{f^2\})\}\}$, *and* $\{Is^{Q^{E_4M_4}}(\{f^2\})\}$ *is the minimal isolator set for the diagnosability of* F^3.

We can verify that failure modes F^1, F^2 *and* F^3 *are* Π*-diagnosable for any* $\Pi \geq 1$. *This implies that when* F^1, F^2 *or* F^3 *occur, they can be detected and isolated after maximum one event generated in the system. Assume that* $\tau^{max} = 100$ *sec. Therefore,* F^1, F^2 *and* F^3 *are* $\Delta-$*diagnosable in the hybrid system for any* $\Delta \geq 100$ *sec.* ■

Remark 5.3.2. *In addition to a DES-first approach for isolator selection, a continuous-first approach can be developed for the discrete sensor selection. In this approach, the diagnosability of failure modes is first investigated with the information provided by the isolators. If isolators designed for the system cannot isolate a failure mode on their own, then appropriate discrete sensors are added to the system to make the failure mode diagnosable.* ■

5.4 Summary

In this chapter, we investigated the problem of isolator selection in hybrid automata. We first developed a method for distinguishing discrete states from each other using isolators. Then, we developed a procedure for finding a minimal isolator set to ensure diagnosability of failure modes in hybrid automata.

In the next chapter, we investigate the application of our framework to a gas turbine engine.

Chapter 6

HYBRID FAULT DIAGNOSIS IN GAS TURBINE ENGINE

Advanced aircraft such as military airplanes are able to meet stringent operational requirements such as short take-off, vertical landing, rapid maneuvering and threat avoidance. Because of their wide range of operational and performance requirements, jet engines have become very complex. Many of the jet engines are benefiting from afterburners, multiple stage compressors and complicated actuators and control systems. As a result, reliability and maintainability requirements demand systematic methods for detecting and isolating faults in jet engines.

In this chapter, we illustrate the application of our hybrid diagnosis methodology to a jet turbine engine. Faults in the fuel supply system and actuators have been the source of many failures in jet engines [104, 29]. We employ our hybrid diagnosis approach for investigating faults in the fuel supply system and the nozzle actuator of a single-spool turbojet engine with an afterburner. The dynamics of the components in the fuel supply system and the nozzle actuator such as pumps and solenoid valves can be described in terms of discrete transitions, and can be represented by DES models. On the other hand, thrust generation in the engine is a continuous process, and the operation of the engine components such as compressor and turbine can be described by continuous static and dynamic thermodynamic relations (i.e., algebraic and differential equations). We develop a hybrid automaton model for describing the engine and its actuator systems. An extended DES is constructed for the engine, and based on the extended DES model, a hybrid diagnoser is constructed and developed. We show that there are faults in the fuel supply system and the nozzle actuator that *cannot* be isolated by a purely DES diagnoser *or* by methods based on the residual generators alone. However, the faults *can* be isolated if the hybrid diagnoser is used.

The remainder of this chapter is organized as follows. In Section 6.1, we explain the basic principles of turbine engine design and modeling which are paramount for understanding the work presented here. We explain the static thermodynamical equations for each component of the engine and develop dynamic relations describing the transient behavior of the engine. In Section 6.2, we develop a hybrid model for the engine. We present a typical fuel supply system and a typical nozzle actuator and develop DES models for them. In this section, we also investigate different operating regimes of the engine and develop a set of linear system models describing its behavior in each operating regime. Fault diagnosis based on the hybrid model of the jet engine is explained in Section 6.3. We describe the simulation methodology and present the results in Section 6.4.

6.1 Jet Engines

Turbine engines are used in many land, sea and air vehicles. The primary purpose of a turbine engine is to give a change in momentum to a mass of fluid. This change in momentum is equivalent to an acceleration of a working fluid, producing an external force (thrust) on the system in accordance with Newton's Third Law of Motion. The

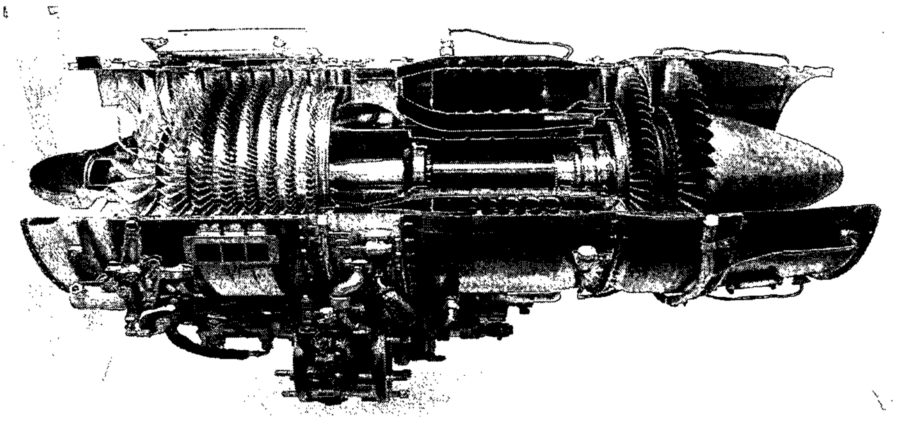


Figure 6.1: A General Electric J85-GE-17A turbojet engine (1970) [3].

jet engine (shown in Figure 6.1) belongs to one type of gas turbine engines and is used to generate a high-speed jet for propulsion. In a jet engine, first air is brought into an intake duct; then the mass of air is compressed by a compressor, and the temperature of the high pressure air is raised by mixing it with fuel and having the mixture burned in a combustion section. The resulting high-pressure, high-temperature fluid is then expanded in a turbine section driving the turbines which in turn power the compressors. The fluid is further expanded through a nozzle section to a high velocity (conversion of pressure and thermal energy into kinetic energy) thus increasing the momentum of the fluid and producing thrust.

Jet engine was near-simultaneously invented by Whittle in England and von Ohain in Germany in early 1940 [104]. Before that, aircraft were powered by the propeller/reciprocating engine propulsion system. Initially, the jet engine was developed solely for military use. Following a great deal of advancement in gas turbine technology, the first civilian airplane appeared in the early 1950s. Until today, jet engines are the main choice for generating propulsion in aircraft.

In general there are four types of jet engines:

1. Turboprop

2. Turbojet

3. Turbofan

4. Turboshaft

In the following, we introduce the thermodynamic terminologies that are used in this chapter.

6.1.1 Thermodynamic Terminologies

Entropy - In thermodynamics, entropy is a measure of the unavailability of a systems energy to do work. In fact, entropy is a measure of the disorder of molecules in a system and is a function of a quantity of heat in a system which is capable of doing a work.

Enthalpy - In thermodynamics and molecular chemistry, the enthalpy (denoted as h) is a quotient or description of thermodynamic potential of a system, which can be used to calculate the heat transfer during a quasistatic process taking place in a closed thermodynamic system under constant pressure.

Adiabatic process - In thermodynamics, an adiabatic process is a thermodynamic process in which no heat is transferred to or from the working fluid.

Reversible process - A reversible process is a process that can be "reversed" by means of extremely small changes in some property of the system without loss or dissipation of energy. A process that is not reversible is called **irreversible**. In an irreversible process, finite changes are made. Thus, the system is not at equilibrium throughout the process.

Isentropic process - An isentropic process is a process during which the entropy of the system remains constant. Any adiabatic and reversible process is an isentropic process.

Specific heat - Specific heat (denoted as C) is the measure of the heat energy required to increase the temperature of a unit quantity of a substance by a certain

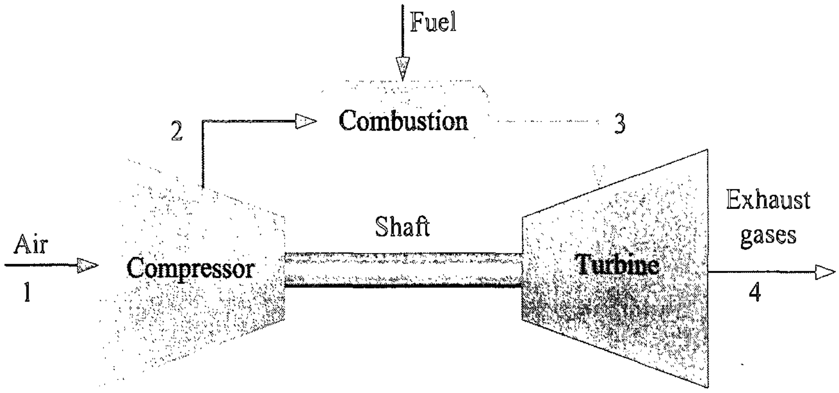


Figure 6.2: Gas generator in a jet engine.

temperature interval (usually 1K).

Specific heat ratio - Specific heat ratio (denoted as γ) is the ratio of the heat capacity at constant pressure c_p to the heat capacity at constant volume c_v.

In this work, we study fault diagnosis in a turbojet with an afterburner. The operation of a gas turbine engine can be described by thermodynamic cycles. In the following, first we explain an ideal thermodynamic cycle called Brayton cycle [102].

6.1.2 Brayton Cycle

Generally, a gas turbine consists of an upstream compressor coupled to a downstream turbine, and a combustion chamber (also called burner) in-between, as shown in Figure 6.2 [102]. In most common air-breathing jet engines, these three components comprise the heart of the engine, called the *gas generator.* The idea behind the gas generator is to take in the air, mix it with fuel, and convert it into a high pressure and high temperature gas. The operation of an ideal gas generator can be described thermodynamically by the Brayton cycle shown in Figure 6.3. In this thermodynamic cycle, the intake air is compressed isentropically (1 to 2), burned at constant pressure inside the combustion chamber (2 to 3), expanded isentropically over the turbine (3 to 4), and finally exhausted back to the starting pressure (4 to 1).

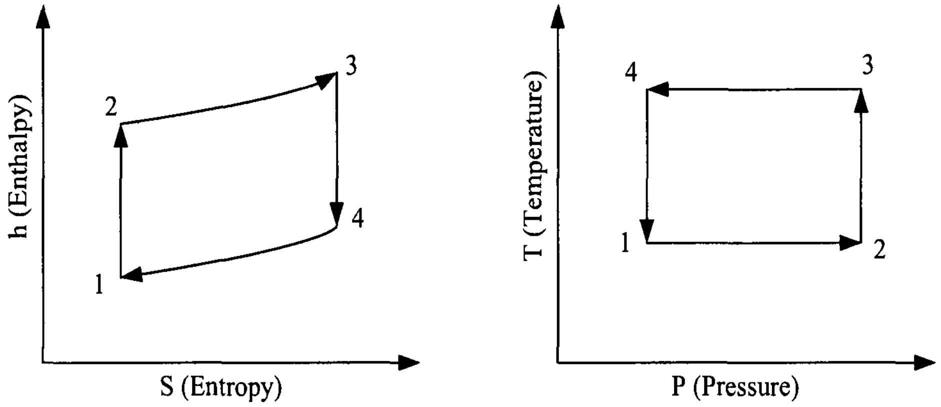


Figure 6.3: Brayton cycle.

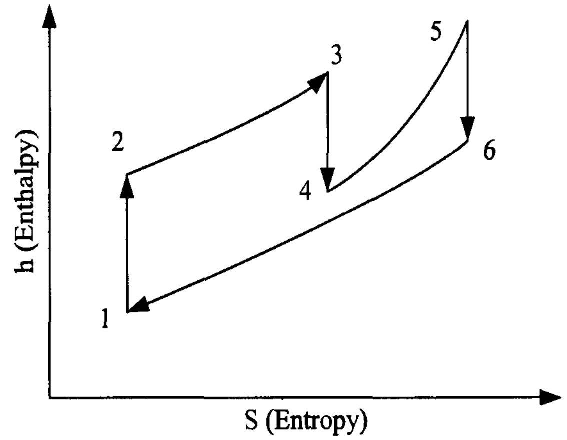


Figure 6.4: Brayton cycle with reheat.

Depending on the applications of the gas turbine, the energy provided is extracted and used for different applications.

The reheat cycle is an effective and widely-used method of increasing thrust quickly. It is mainly employed by military supersonic aircraft. The specific output of a Brayton cycle can be increased through a reheating process, where the expanded gas from the expansion process is reheated before the exhaust through the nozzle. Figure 6.4 illustrates an ideal Brayton cycle with a reheat process.

A turbojet engine, as shown in Figure 6.5, can be constructed by adding an intake duct and an exhaust nozzle to a gas generator. The exhaust nozzle converts

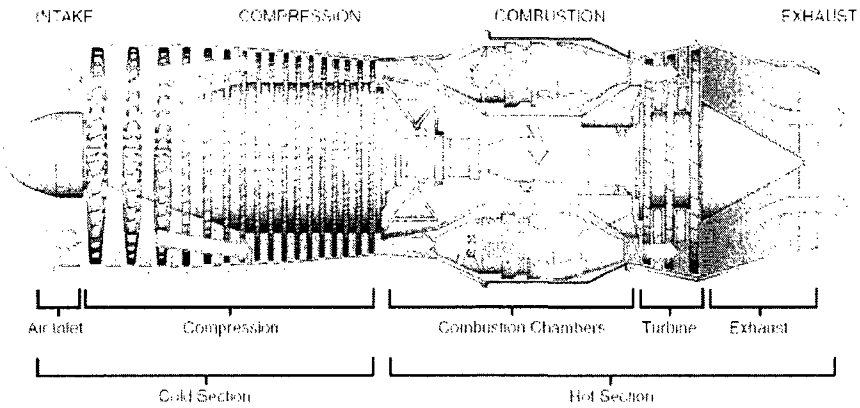


Figure 6.5: A turbojet engine [4].

the internal energy of the hot gas into kinetic energy or thrust. The work extracted by the turbine is to drive the compressor, or to provide auxiliary power. In addition, part of the work extracted by the turbine is also used to drive a fan for a turbofan, or a propeller for a turboprop.

In the following, we explain some concepts important for engine modeling.

6.1.3 Engine Modeling

A mathematical representation of a gas turbine engine is fairly common and has been investigated by several authors in the literature (e.g., [102, 89]). For this work, a simulation of a single-spool turbojet engine was developed by using thermodynamic, aerodynamic and mechanical relationships of each of the major components. This model represents the functional relations that exist among the engine variables, such as pressures, temperatures, and gas flow rates. The details of the thermodynamic relations reviewed in this chapter can be found in [102]. First, we specify the gas model we use.

Gas Model

In this application, it is assumed that the working fluids, i.e., the air and combustion products are modeled as perfect gases in their thermodynamic equilibria. Generally, specific heat at constant pressure (c_p) changes with temperature. Also, c_p and specific heat ratio (γ) for most typical hydrocarbons and air combustion products are functions of temperature and the fuel-air ratios [102]. Therefore, it is necessary to model the variation of c_p and γ across engine components where the changes are significant, for instance, downstream of the combustion chamber. Throughout the analysis, the variation of gas properties with temperature is approximated by assuming constant gas properties, such as c_p, γ, and gas constant R, at two different sections across the engine (refer to Figure 6.6):

- Section 1: components upstream of main combustion chamber (i.e., before station 3)
- Section 2: components downstream of main combustion chamber (i.e., station 3 and after station 3)

Concept of Stagnation

The change in the kinetic energy terms in the steady flow energy equation is accounted for using the concept of stagnation or (total) enthalpy. The stagnation enthalpy h_0 is the enthalpy which a steady fluid of enthalpy h and velocity C would possess when it is brought to rest in the absence of any heat or work interactions. From energy equations, we have:

$$(h_0 - h) + \frac{1}{2}(0 - C^2) = 0 \tag{6.1}$$

and thus h_0 will be obtained from

$$h_0 = h + \frac{1}{2}C^2$$

For calorically perfect gas (i.e., constant c_p), h can be substituted by c_pT and the above equation can be written for stagnation temperature or total temperature as

$$T_0 = T + \frac{C^2}{2c_p} \tag{6.2}$$

Usually, T is referred to as the static temperature and $\frac{C^2}{2c_p}$ is called dynamic temperature. When a gas is slowed down, the temperature rises and there is a rise in the pressure at the same time. Assuming that the steady flowing gas is brought to rest not only adiabatically but also reversibly, i.e., isentropically, a stagnation (or total) pressure p_0 can be defined in a similar way to T_0. Therefore, we have

$$\frac{p_0}{p} = \left(\frac{T_0}{T}\right)^{\gamma/(\gamma-1)} \tag{6.3}$$

where p is the pressure of the fluid before being brought to rest.

Isentropic Efficiencies

Since the objective of the engine is to produce work, the efficiencies of the engine components are normally expressed in terms of the ratio of the actual and ideal power transfers:

$$\eta = \frac{W'}{W} = \frac{\Delta h'}{\Delta h}$$

where W' and $\Delta h'$ are the ideal power transfer and enthalpy change of an engine component when the process is isentropic, and W and Δh are the actual power transfer and enthalpy change in the presence of friction. The ideal process in the engine is isentropic. Therefore, the efficiencies are called isentropic efficiencies. For

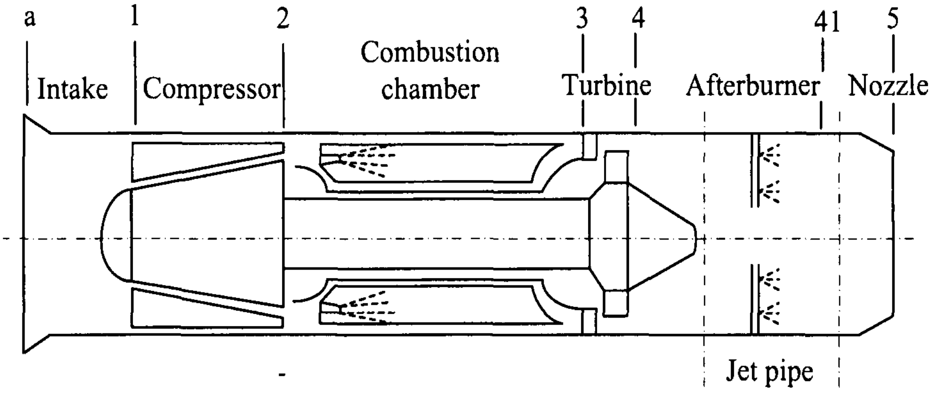


Figure 6.6: Station numbering in a turbojet engine with afterburner.

a perfect gas, we have

$$\Delta h = c_p \Delta T$$

This relation is sufficiently accurate for real gases in gas turbines if a mean c_p over the range of temperature is used. In addition, the ideal and actual temperature changes are not very different, and as a result, the mean c_p can be assumed to be the same for both. Therefore, the isentropic efficiencies in the engine are defined in terms of temperature.

In the following subsection, we explain the components and their thermodynamic relations for a single-spool turbojet with afterburner which will be the focus of our work.

6.1.4 Modeling of Components

Figure 6.6 illustrates a schematic of a turbojet with an afterburner and the station numbering used in our work (adopted from [102]). In the following, we explain each component of the engine in more details.

Intake Duct

Intake duct is placed before the compressor and supplies the engine with the required air flow at the highest possible pressure. The air velocity in the intake duct decreases when air reaches the compressor. At the same time, the temperature and the pressure increase because of stagnation properties. Let C_a, T_a and p_a denote the velocity, temperature and pressure of the fluid gas at the entrance of the intake duct, respectively.

Referring to Equation (6.2), we have

$$T_{01} = T_{0a} = T_a + \frac{C_a^2}{2c_{pa}}$$

and

$$\frac{p_{01}}{p_a} = \left(\frac{T'_{01}}{T_a}\right)^{\gamma/(\gamma-1)}$$

where T_{01} and p_{01} are the temperature and pressure at the compressor inlet, respectively, and T'_{01} is the temperature which would have been reached if the friction was absent. For cycle analysis, calculation of the stagnation pressure at the compressor inlet is necessary. In fact, the pressure rise $p_{01} - p_a$ which is referred to as *ram* pressure is of our interest. Since the process of ram compression is not isentropic, we introduce an isentropic efficiency, η_i for the intake duct defined as

$$\eta_i = \frac{T'_{01} - T_a}{T_{01} - T_a}$$

Therefore, we have

$$T'_{01} - T_a = \eta_i \frac{C_a^2}{2c_{pa}}$$

and consequently, the inlet pressure ratio can be found from

$$\frac{p_{01}}{p_a} = \left[1 + \frac{T'_{01} - T_a}{T_a}\right]^{\gamma/(\gamma-1)} = \left[1 + \eta_i \frac{C_a^2}{2c_{pa}T_a}\right]^{\gamma/(\gamma-1)}$$

We have the following relation for Mach number M:

$$M = C/(\gamma RT)^{1/2}$$

We also have

$$\gamma R = c_{pa}(\gamma - 1)$$

Thus, the inlet pressure ratio equation can be written as

$$\frac{p_{01}}{p_a} = \left[1 + \eta_i \frac{\gamma - 1}{2} M_a^2\right]^{\gamma/(\gamma-1)} \tag{6.4}$$

where M_a is the Mach number in the air temperature and pressure. The inlet temperature ratio can be expressed in terms of M_a as

$$\frac{T_{01}}{T_a} = \left[1 + \frac{\gamma - 1}{2} M_a^2\right] \tag{6.5}$$

Compressor

A compressor in a gas turbine engine is in charge of providing high-pressure air to the combustion chamber. There are two major classes of compressors used in aircraft gas turbines [104, 102]: centrifugal flow compressor and axial flow compressor. In the centrifugal compressor, air is taken into the compressor near the axis and accelerated outward from the axis by a centrifugal force. Subsequently, the spin of the air is removed and the air is ejected at high velocity and high kinetic energy. The pressure rise is produced in part by expansion of the air in a diffuser manifold by conversion of the kinetic energy of the moving air into static pressure energy.

The advantage of centrifugal compressors is that they can be easily built in relatively small size. However, they have smaller efficiency in comparison with the axial flow compressors particularly in high pressure ratios. Centrifugal compressors are usually used in small engines. Axial compressors are used in larger gas turbine engines. The axial flow compressors are composed of a series of rotating airfoils called rotor blades and a stationary set of airfoils called stator vanes. The air is compressed in direction parallel to the axis of the engine. Enthalpy rise occurs in the rotors in which both static pressure and kinetic energy are increased. Some of the swirl velocity produced by the rotor is then removed by the stator vanes, decreasing the kinetic energy and therefore increasing the static pressure. Modern compressors can yield compression ratios over 25 : 1 and efficiencies over 90 percent [104].

For a given pressure ratio $PR_{comp} = p_{02}/p_{01}$, we have the following thermodynamic relation for the compressor:

$$T_{02} - T_{01} = \frac{T_{01}}{\eta_c}\left[\left(\frac{p_{02}}{p_{01}}\right)^{(\gamma-1)/\gamma} - 1\right] \tag{6.6}$$

where η_c is the isentropic efficiency of the compressor. The power consumed by the compressor W_c can be calculated from

$$W_c = c_{pa}\dot{m}_a(T_{02} - T_{01}) \tag{6.7}$$

where $\dot{m}_a$ is the mass flow passing through the compressor. It should be noted that in this work, we do not consider air bleeding.

In single-spool engines, there is only one main shaft (or rotor) in the system. The rotor consists of the rotatory parts of the compressor and the turbine. The speed of the engine is specified by the rotor speed and is presented in Revolution Per Second (RPS) or Revolution Per Minute (RPM). The speed of the engine is a function of the power generated by the turbine for turning the compressor and the total moment of inertia of the rotatory system. The power consumed by the compressor is related to the speed of the shaft as follows:

$$W_c = \frac{J(2\pi N)^2}{2} \tag{6.8}$$

where J is the moment of inertia of the shaft and N is the speed (RPS) of the shaft.

Combustion Chamber

Combustion chamber is the place in the engine in which the fuel is burned in the high pressure air supplied by the compressor to rise the temperature. The rise in the temperature is due to the energy released by the burning fuel. An electrical spark is required only in the beginning of the combustion process. After initial ignition, the flame must be self-sustaining. It is desirable to keep the pressure of the gas unchanged in the combustion chamber. However, there is usually a loss in the pressure represented as a percentage of the total pressure.

The pressure at the turbine inlet p_{03} is calculated as follows:

$$p_{03} = (1 - \Delta p_b)p_{02} \tag{6.9}$$

where Δp_b is the combustion pressure loss. The temperature T_{03} at the steady state is calculated from the energy conversion give by the following relation:

$$c_{pa}\dot{m}_a T_{02} + \eta_b K_f \dot{m}_f = c_{pg}\dot{m}_g T_{03} \tag{6.10}$$

where η_b is the combustion efficiency, K_f is the low calorific value of the fuel, $\dot{m}_f$ is the fuel mass flow rate and $\dot{m}_g$ is the gas mass flow rate at the turbine inlet. In the steady-state operation, we also have

$$\dot{m}_a + \dot{m}_f - \dot{m}_g = 0 \tag{6.11}$$

Turbine

The function of the turbine in a jet engine is to extract a portion of the pressure and kinetic energy from the high-temperature combustion gases for driving the compressor and accessories. In a typical jet engine about 75 percent of the power produced is used internally to drive the compressor. The remaining power is used to generate the required thrust [104]. In contrast to compressor, turbine is a rotatory component in which gas with high-temperature passes. Therefore, the choice of the material used in turbine is very crucial for the engine operation and performance. For any engine, the maximum allowable turbine inlet temperature is specified by the manufacturer. Like compressors, the behavior of a turbine is also usually represented by characteristic maps.

The power developed by the turbine is proportional to the temperature decrease in the turbine and is given by

$$W_t = \dot{m}_g c_{pg} (T_{03} - T_{04}) \tag{6.12}$$

In a jet engine, the power generated by the turbine is proportional to the power consumed by the compressor, namely

$$W_t = \frac{W_c}{\eta_m}$$

where η_m is the mechanical transmission efficiency of the engine. In this work, we study turbojet engines with no air bleedings.

Temperature at the outlet of the turbine can be calculate from the work compatibility relation

$$T_{03} - T_{04} = \frac{c_{pa}(T_{02} - T_{01})}{c_{pg}\eta_m} \tag{6.13}$$

Pressure is calculated from the following thermodynamic relation:

$$p_{04} = p_{03} \left(\frac{T'_{04}}{T_{03}} \right)^{\gamma/(\gamma-1)} \tag{6.14}$$

where T'_{04} is the temperature of the turbine outlet if the expansion process would be isentropic. In case of non-isentropic process, an isentropic efficiency η_t is defined for the turbine and T'_{04} is calculated from the following equation:

$$T'_{04} = T_{03} - \frac{1}{\eta_t}(T_{03} - T_{04}) \tag{6.15}$$

Nozzle

Nozzle is the final component of a jet engine in which the working fluid is expanded to produce a high-velocity jet. The high-pressure exhaust gas is accelerated in a jet pipe situated between the turbine outlet and the nozzle throat to come close to the ambient pressure and consequently, producing thrust. The flow through the nozzle may be subsonic or supersonic. For supersonic engines generating gas jets with very high speeds, the design of the nozzle is very paramount in the engine performance. Initially, assume that the afterburner is not in operation.

The thrust produced by the engine is given by

$$F = \dot{m}_n C_5 + A_n(p_5 - p_a) \tag{6.16}$$

where $\dot{m}_n$ is the mass flow rate of the gas exiting the nozzle, C_5 is the speed of the jet passing through the nozzle, A_n is the area of the nozzle throat and p_5 is the pressure of the gas at the nozzle throat. In this work, we assume that the mass flow rate exiting the nozzle is the same as the mass flow rate passing through the turbine if the afterburner is not in operation, i.e., $\dot{m}_n = \dot{m}_g$.

The nozzle exit temperature T_5 is given by

$$T_{04} - T_5 = \eta_j T_{04} \left[1 - \left(\frac{1}{p_{04}/p_5} \right)^{(\gamma-1)/\gamma} \right] \tag{6.17}$$

where η_j is the isentropic efficiency of the nozzle and is given by

$$\eta_j = \frac{T_{04} - T_5}{T_{04} - T_5'} \tag{6.18}$$

where T_5' is the temperature which would be reached if the expansion process in the nozzle was isentropic.

Let T_{05} denote the temperature of the gas in the jet pipe before the nozzle throat. Assuming there is no drop in the temperature when the gas passes through the jet pipe, i.e., $T_{05} = T_{04}$, $T_{04} - T_5$ is the temperature equivalent of the jet velocity $(C_5^2/2c_{pg})$. The speed of the jet is a function of the pressure ratio p_{04}/p_5. The critical pressure ratio p_{04}/p_c is the pressure ratio p_{04}/p_5 which results in $M_5 = 1$. Here M_5 is the Mach number of the fluid jet. For pressure ratios up to the critical value, p_5 will be equal to the ambient pressure p_a and the pressure thrust $(A_n(p_5 - p_a))$ is zero. For pressure ratios greater than the critical value, the nozzle is chocked, p_5 remains constant at the critical pressure p_c, and C_5 remain constant at the sonic value $(\gamma R T_5)^{1/2}$.

We have the following relations for the speed of the jet

$$\frac{T_{04}}{T_5} = \frac{T_{05}}{T_5} = 1 + \frac{C_5^2}{2c_{pg}T_5} = 1 + \frac{\gamma - 1}{2} M_5^2 \tag{6.19}$$

For sonic speed $M_5 = 1$, we have

$$\frac{T_{04}}{T_c} = \frac{\gamma + 1}{2} \tag{6.20}$$

where T_c is the temperature of the gas corresponding to the critical pressure p_c.

From Equation (6.18), we have

$$T_c' = T_{04} - \frac{1}{\eta_j}(T_{04} - T_c)$$

The critical pressure p_c can be calculated from

$$\frac{p_c}{p_{04}} = \left(\frac{T_c'}{T_{04}}\right)^{\gamma/(\gamma-1)} = \left[1 - \frac{1}{\eta_j}\left(1 - \frac{T_c}{T_{04}}\right)\right]^{\gamma/(\gamma-1)}$$

Replacing for T_c/T_{04} from Equation (6.20), we have the following relation for the critical pressure ratio

$$\frac{p_{04}}{p_c} = \frac{1}{\left[1 - \frac{1}{\eta_j}\left(\frac{\gamma-1}{\gamma+1}\right)\right]^{\gamma/(\gamma-1)}} \tag{6.21}$$

The ratio p_{04}/p_a is called the nozzle pressure ratio. A convenient way to check if the nozzle is chocked is to calculate the nozzle pressure ratio. Nozzle is chocked if $p_{04}/p_a \geq p_{04}/p_c$. We have the following approximate relation for the nozzle area A_n:

$$A_n = \frac{m_n}{\rho_n C_n} \tag{6.22}$$

where m_n is the mass of the gases in the nozzle, ρ_n is the density of the exiting gases and is obtained from $\frac{p_n}{RT_c}$, and C_n is the speed of the exiting gases and is calculated from $[2c_{pg}(T_{04} - T_c)]^{\frac{1}{2}}$ or $(\gamma RT_c)^{\frac{1}{2}}$.

Optimal nozzle performance occurs when the nozzle exit pressure is not far from the ambient pressure. As a result, for nozzles with a large range of operating pressure ratios, mechanisms for geometrical variation must be possible [89]. In this work, we consider nozzle with variable exhaust area.

Afterburner

Afterburner or reheat is another component which is added in the jet pipe of some engines, primarily those on military supersonic aircraft, to boost the thrust temporarily, both for supersonic flight and for takeoff. The jet pipe of an engine with afterburner is longer than that of an engine without afterburner. Like the main fuel system of the combustion chamber, afterburner consists of a fuel system and some fuel injectors situated in the jet pipe. When the afterburner turns on, fuel is injected in the jet pipe. Since the temperature of the incoming gases is very high (in the range of 600 -1400 K for the engine in this work), fuel is ignited easily and as the result of the combustion process, the afterburner exit temperature increases significantly. Consequently, there will be a steep increase in the engine net thrust.

The afterburner exit temperature, T_{041} is calculated from the following relation:

$$c_{pg}\dot{m}_g T_{04} + \eta_{AB} K_f \dot{m}_{fAB} = c_{pg}\dot{m}_n T_{041} \tag{6.23}$$

where η_{AB} is the afterburner efficiency, $\dot{m}_{fAB}$ is the afterburner fuel mass flow rate and $\dot{m}_n$ is the mass flow rate of the gases passing to the nozzle. Here, we also have

$$\dot{m}_g + \dot{m}_{fAB} - \dot{m}_n = 0 \tag{6.24}$$

We assume that the afterburner uses a fuel with the same low calorific value as is used in the combustion chamber. As a result of the afterburner combustion process, nozzle mass flow which is the afterburner entry mass flow plus the afterburner fuel flow increases, but the afterburner exit pressure deceases due to the heating and friction and turbulence losses. We use a variable pressure loss factor for the afterburner. We assume that the pressure loss in the afterburner is proportional to the temperature increase by the afterburner [102]. Here, we assume that there will be a pressure loss of 10 percent when T_{041} is two times of T_{04}. Therefore, an approximate value for the pressure at the end of the jet pipe, p_{041}, can be obtained from

$$p_{041} = p_{04}\left[1 - 0.1\left(\frac{T_{041}}{T_{04}} - 1\right)\right] \tag{6.25}$$

When using afterburner in the engine, the temperature in the jet pipe, T_{041}, is no more equal to the temperature of the turbine outlet, T_{04}. In fact, T_{041} is usually much larger than T_{04}, and as a result, the temperature equivalent of the jet velocity, i.e., $T_{041} - T_5$, yields greater speed for the gas jet. When the afterburner is in operation, the ratio p_{041}/p_a gives the nozzle pressure ratio. Also, T_{04} and p_{04} will be replaced by T_{041} and p_{041} in equations (6.18), (6.20) and (6.21).

The large temperature rise due to the afterburner combustion process changes the density of the flow approaching the nozzle considerably. Specifically, the temperature increase by the afterburner lessens the density of the gases passing through the nozzle and as a result, reduces the mass flow passing through the engine. Consequently, the pressure in the combustion chamber as well as the turbine inlet temperature will increase which may cause a compressor stall or turbine overheating. Therefore, it is essential for engines with an afterburner to use a variable area nozzle to increase the afterburner exit volume flow. Afterburner is normally brought into operation when the engine is at its maximum rotational speed [102]. The rotational speed of the engine should not change when afterburner is in operation, and the nozzle must pass the same mass flow at a much reduced density. This can be achieved only if the nozzle installed allows a significant increase in the nozzle throat area.

The thermodynamic relations presented so far describe the engine performance in the steady state. However, for fault diagnosis purposes, the transient behavior of the engine is also required. In the following, we discuss the dynamic modeling of the engine.

6.1.5 Dynamic Modeling of Engine

Transient behavior of the engine is critical in aircraft applications. It is very important for aircraft to have rapid thrust response to meet their mission requirements while ensuring the safe operation of the engine. Therefore, understanding the dynamic behavior of aircraft engines is an important part of the design and development of control systems. It is also essential for health monitoring and fault diagnosis of engines.

For calculating the thermodynamic properties of the engine, one has to satisfy the requirements for compatibility of flow and work between the components. For example, the mass flow of the gas passing through the turbine is the sum of the mass flow of the air entering the combustion chamber and the mass flow of the fuel. The power generated by the turbine and applied to the rotor is equal to the power consumed by the compressor. During transient operation of the engine, the compatibility of the flow and work may be violated. In the following we explain this issue in more details.

Violation of Work Compatibility [70, 102]

The turbine power applied to the rotor changes when the flow rate of the fuel changes. The change of the turbine power does not transfer to the compressor instantaneously. In other words, the turbine torque can be greater or less than the compressor torque right after the mass flow of the fuel changes. The excess or deficiency of the power applied to the rotor causes the acceleration or deceleration of the rotor. The acceleration of the compressor rotor and the torque discrepancy are related by Newton's Second Law of Motion. We have

$$\Delta G = J\dot{\omega}$$

where J is the moment of inertia of the rotor, $\dot{\omega}$ is the angular acceleration and ΔG is the difference between the turbine and compressor torques and is given by the following equation,

$$\Delta G = \frac{\eta_m W_t - W_c}{2\pi N} = \frac{\eta_m \dot{m}_g c_{pg} \Delta T_{034} - \dot{m}_g c_{pa} \Delta T_{012}}{2\pi N}$$

where N is the rotor speed, T_{034} is the temperature decrease by the turbine, i.e., $T_{034} = T_{03} - T_{04}$ and T_{012} is the temperature increase by the compressor, i.e., $T_{012} = T_{02} - T_{01}$. We also have

$$\omega = 2\pi N$$

and therefore,

$$\dot{\omega} = 2\pi \dot{N}$$

The rate of change of the rotor speed is therefore given by

$$\dot{N} = \frac{\eta_m \dot{m}_g c_{pg} \Delta T_{034} - \dot{m}_a c_{pa} \Delta T_{012}}{(2\pi)^2 J N} \tag{6.26}$$

Violation of Flow Compatibility [70]

When the mass flow of the fuel changes in the combustion chamber, the flow of the gas does not change instantaneously [70]. In fact, we have

$$\dot{m}_a + \dot{m}_f - \dot{m}_g = \Delta \dot{m}_g \tag{6.27}$$

where $\Delta \dot{m}_g$ is an equality violation factor. In addition, pressures and temperatures in the combustion chamber cannot change instantaneously because of the finite volume of the chamber, and therefore, the assumption of flow compatibility at all times is not exactly true [102].

In this work, we assume the violation of the flow compatibility only in the combustion chamber, and do not study the violation of flow compatibility in afterburner and nozzle. Since the speed of the gases in the jet pipe is too high (relative to the speed of the gases in the combustion chamber), we assume that the mass flow of the gases after the afterburner rapidly follows the changes of the afterburner fuel flow. Furthermore, we assume that the pressures and temperatures in the afterburner and in the nozzle change very fast (relative to the pressure and temperature in the combustion chamber) following the change in the afterburner fuel flow.

For a perfect gas, we have the following equation which is known as *the perfect gas equation*

$$\frac{pV}{T} = mR \tag{6.28}$$

In equation (6.28), p, T, m and V are the pressure, the temperature, the mass and the volume of the gas, respectively, and R is the specific gas constant. Since we have assumed that the produced gases in the combustion chamber follow the perfect gas rules, the temperature and pressure of the gases in the combustion chamber satisfy the perfect gas equation, namely

$$\frac{p_{03}V_{comb}}{T_{03}} = m_g^{comb}R \tag{6.29}$$

where, V_{comb} is the volume of the combustion chamber and m_g^{comb} is the mass of the gases produced in the combustion chamber.

Differentiating both sides of the equation (6.29) and assuming a constant volume for the combustion chamber yields the following dynamical equation.

$$\dot{p}_{03}V_{comb} = \dot{T}_{03}m_g^{comb}R + RT_{03}\dot{m}_g^{comb}$$

We have

$$\dot{m}_g^{comb} = \Delta\dot{m}_g$$

Using equations (6.27) and (6.29), we substitute $m_g^{comb}R$ with $p_{03}V_{comb}/T_{03}$ and $\dot{m}_g^{comb}$ with $\dot{m}_a + \dot{m}_f - \dot{m}_g$, and obtain the nonlinear equation

$$\dot{p}_{03} = \frac{RT_{03}}{V_{comb}}(\dot{m}_a + \dot{m}_f - \dot{m}_g) + \frac{p_{03}}{T_{03}}\dot{T}_{03} \tag{6.30}$$

As a result of the violation of the flow compatibility in the combustion chamber, the heat transfer equation is no longer in place and we have

$$c_{pa}\dot{m}_a T_{02} + \eta_b K_f \dot{m}_f - c_{pg}\dot{m}_g T_{03} = \Delta\dot{Q}_g \tag{6.31}$$

where $\Delta\dot{Q}_g$ is an equality violation factor. In general, in steady state, heat and temperature have the following relation in a constant volume:

$$Q = c_v m T$$

where m is the mass of the gases in the volume, Q is the heat energy, c_v is the specific heat when volume is constant and T is the temperature in the volume. In the combustion chamber, we have:

$$Q = c_{vg} m_g^{comb} T_{03} \tag{6.32}$$

The change in the mass flow rate changes the heat energy and temperature. Differentiating both sides of (6.32), we obtain

$$\dot{Q} = c_{vg} m_g^{comb} \dot{T}_{03} + c_{vg} \dot{m}_g^{comb} T_{03} \tag{6.33}$$

We also have

$$\dot{Q} = \Delta\dot{Q}_g$$

and therefore, by substituting for $\Delta\dot{Q}_g$ from Equation (6.31), we obtain the following nonlinear dynamical equation for $\dot{T}_{03}$

$$\dot{T}_{03} = \frac{1}{c_{vg} m_g^{comb}} \left[\left(c_{pa} T_{02} \dot{m}_a + K_f \eta_b \dot{m}_f - c_{pg} T_{03} \dot{m}_g \right) - c_{vg} T_{03} \left(\dot{m}_a + \dot{m}_f - \dot{m}_g \right) \right] \tag{6.34}$$

Control Inputs

In our work, there are three mechanisms for controlling the parameters of the engine: mass flow rate of the main fuel $\dot{m}_f$, mass flow rate of the afterburner fuel $\dot{m}_{fAB}$ and the area of the nozzle throat A_n. All of the control inputs are functions of the Power Level Angle (PLA) set by the pilot. In our work, the PLA may vary from zero degree to 90^o. We assume that when the PLA is at 70^o, the maximum thrust can be obtained with the maximum of mass flow rate of the main fuel. For producing greater thrust the afterburner must be used. Therefore, for the values of the PLA from 70^o to 90^o the mass flow rate of the main fuel remains at its maximum and the mass flow rate of the afterburner fuel increases linearly with the PLA to reach its maximum at 90^o. Furthermore, we assume that the area of the nozzle remains at its initial value if the PLA is less than 70^o. For the PLA values greater than 70^o, the area of the nozzle increases linearly with the PLA to reach its maximum when the PLA reaches 90^o. Let $\dot{m}_f^{max}$, $\dot{m}_{fAB}^{max}$, A_n^{max} and A_n^{init} denote the maximum mass flow rate of the main fuel, the maximum mass flow rate of the afterburner fuel, the maximum nozzle area and the initial nozzle area, respectively.

The relation between the PLA and the control inputs are as follows

$$\dot{m}_f = \begin{cases} \frac{PLA \times \dot{m}_f^{max}}{70} & \text{if } PLA \leq 70^o \\ \dot{m}_f^{max} & \text{if } PLA > 70^o \end{cases} \tag{6.35}$$

$$\dot{m}_{fAB} = \begin{cases} 0 & \text{if } PLA \leq 70^o \\ \frac{(PLA-70)\times \dot{m}_{fAB}^{max}}{20} & \text{if } PLA > 70^o \end{cases} \tag{6.36}$$

$$A_n = \begin{cases} A_n^{init} & \text{if } PLA \leq 70^o \\ A_n^{init} + \frac{(PLA-70)\times(A_n^{max}-A_n^{init})}{20} & \text{if } PLA > 70^o \end{cases} \tag{6.37}$$

Engine States

Dynamic equations (6.26), (6.30) and (6.34) describe the transient behavior of the engine. Given the set of control inputs and atmospheric condition parameters M_a, p_a and T_a, one can calculate all the parameters of the engine such as temperatures, pressures and mass flow rates at any station by using these three dynamical equations and the thermodynamic static equations (6.4) to (6.25) explained in the previous subsections. Hence, the variables T_{03}, p_{03} and N are considered as the state variable of the system in the remaining parts of this chapter.

Engine Measured Variables

Reliability of aircraft engines is a very important factor in the design stage. Therefore, it is desirable to measure and monitor the key parameters of the engine. However, measuring some of the engine parameters may not be easy due to the very high temperature and pressure in the engine components. For example, the temperature in the jet pipe (T_{041}) when the afterburner is in operation can exceed $2000K$. Besides, the speed of the gas jet can be very high particularly when the afterburner is in operation. It is not easy to build temperature sensors which can measure the temperature with high precision under these severe conditions.

We assume that three states of the system, namely T_{03}, p_{03} and N are measured. In real applications, due to practical limitations the turbine outlet temperature (T_{04}) is measured instead of T_{03} [102]. Since we assume that the turbine is reliable and fault free, T_{03} can be calculated by using the turbine characteristic map and the turbine outlet temperature T_{04}. We also assume that the pressure at the turbine inlet p_{03} is proportional to the pressure at the compressor outlet p_{02}. Therefore, by measuring the temperature at the combustor inlet (after the compressor), we are able to calculate p_{03}.

Nonlinear Static and Dynamic Engine Models

Based on the thermodynamic relations discussed in the last subsection for the engine components, we can represent a nonlinear static model of the engine according to the following system

$$\begin{cases} f(x,u,v) = 0 \\ y = g(x,u,v) \end{cases} \tag{6.38}$$

where $x = [T_{03}, p_{03}, N]$ is the state vector, $u = [W_f, W_{fAB}, A_n]$ is the control input vector, $v = [M_a, T_a, p_a]$ is the vector of flight atmospheric conditions, and $y = [T_{03}, p_{03}, N]$ is the vector of measured variables.

Using the dynamical equations (6.26), (6.30) and (6.34), and thermodynamic static equations (6.4) to (6.25), the nonlinear differential equations of the system can be developed as

$$\begin{cases} \dot{x} = f(x,u,v) \\ y = f(x,u,v) \end{cases} \tag{6.39}$$

The vectors x, u, v and y are defined similar to those of the static model of the engine. In steady-state conditions, all derivatives are equal to zero and Equation (6.39) reduces to the system of Equation (6.38).

In the following, we explain the linearization method of the engine model in our work.

6.1.6 Linearization

Techniques based on linear system models are well established when compared to nonlinear modeling techniques. Analyzing and verification of linear systems, in general, are easier than the nonlinear case. Linear system models have been used in jet engines for control, diagnosis and simulation [70]. The hybrid diagnosis method that we develop here is based on the linear system models of the engine. In this subsection, we explain the methods for obtaining the linear system models for the engine assuming a fixed flight atmospheric condition. In fixed flight atmospheric conditions, Equation (6.38) can be represented by

$$\begin{cases} \dot{x} = F(x, u) \\ y = G(x, u) \end{cases} \tag{6.40}$$

Linearization of the nonlinear differential and algebraic equations is a conventional method for deriving a linear system. In this method, a Taylor's series expansion of Equation (6.40) is performed at the operating point $\alpha = (x_0, u_0)$ as follows

$$F(x, u) \simeq F(x_0, u_0) + \frac{\partial F}{\partial x}\bigg|_{\substack{x=x_0 \\ u=u_0}} (x - x_0) + \frac{\partial F}{\partial u}\bigg|_{\substack{x=x_0 \\ u=u_0}} (u - u_0)$$

At the operating point, we have $F(x_0, u_0) = 0$. The series is truncated after the linear terms and by the change of variables $\Delta x = x - x_0$, $\Delta u = u - u_0$, $\Delta y = y - y_0$, we obtain the linear system model

$$\begin{cases} \Delta\dot{x} = A(\alpha)\Delta x + B(\alpha)\Delta u \\ \Delta y = C(\alpha)\Delta x + D(\alpha)\Delta u \end{cases} \tag{6.41}$$

where $A(\alpha) = \frac{\partial F}{\partial x}\Big|_{\substack{x=x_0 \\ u=u_0}}$, $B(\alpha) = \frac{\partial F}{\partial u}\Big|_{\substack{x=x_0 \\ u=u_0}}$, $C(\alpha) = \frac{\partial G}{\partial x}\Big|_{\substack{x=x_0 \\ u=u_0}}$ and $D(\alpha) = \frac{\partial G}{\partial u}\Big|_{\substack{x=x_0 \\ u=u_0}}$ are the Jacobian matrices calculated at the operating point α. It is observed that the eigenvalues of A are all in the left side of the s-plane for all linearized models.

The system matrices A and B can also be estimated by using the observations of inputs and outputs based on the numerical methods developed for system identification. This method is easier for computer implementations and is useful when an engineering system is modeled using detailed computer routines. In this method, the states of the model are successively perturbed about their operating points and the responses are measured. Then, system matrices are calculated using numerical algorithms.

MATLAB software has built-in routines for linearization of nonlinear models using numerical perturbation techniques. We use these routines for linearization of the engine dynamics in our work.

6.1.7 Actuator Systems

As mentioned earlier in this section, we have three ways of controlling the engine parameters, namely: the mass flow rate of the main fuel $\dot{m}_f$, the mass flow rate of the afterburner fuel $\dot{m}_{fAB}$ and the nozzle throat area A_n. In this subsection, we describe general structures of the actuators in our work. The functionality of the fuel supply system (and the nozzle actuator) in many jet engines is more or less based on similar concepts. The detailed description of the fuel supply system and nozzle actuator can be found in [104, 24, 88]. In the following, we refer to the fuel supply system and the nozzle actuator as our actuator systems.

Fuel Supply System

In general, the fuel supply system for an aircraft engine must provide the following characteristics[103]:

- Sufficient pressure to move the fuel from the tank to the engine.

- A metering device to regulate the flow.

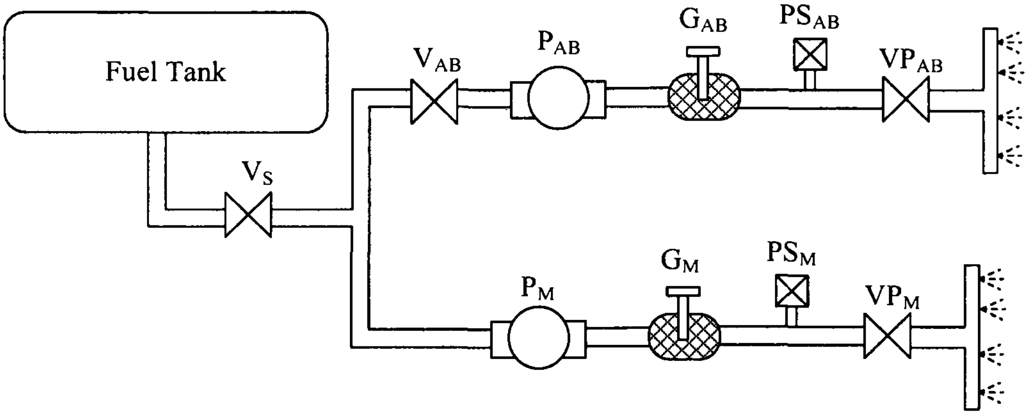


Figure 6.7: Schematic of a simplified typical fuel supply system of a turbojet with afterburner.

- Sufficient pressure at the combustion chamber and afterburner to ensure proper atomization.

- An injection method that will ensure proper fuel distribution.

Figure 6.7 shows a schematic of the fuel supply system that is used in our work. The fuel is reserved in a tank. The fuel supply system has two branches. One branch models the main fuel supply system and is in charge of providing the fuel with sufficient pressure to the combustion chamber and the other branch models the afterburner fuel system and is in charge of supplying afterburner with fuel. In Figure 6.7, the components of the main fuel branch are shown by indices M and the components of the afterburner fuel branch are shown by indices AB. There is a pressurizing pump in each branch for boosting the fuel pressure (P_M and P_{AB}).

When the engine starts, the pumps become operational. They remain in operation until the engine shuts off. Both branches utilize a governor (servo valve/metering system) (shown as G_M and G_{AB}) for regulating the flow of the fuel supplied to the engine. The governors are equipped with a metering device whose position controls the effective area that the fuel passes through. There is also a shut-off solenoid valve (V_S) after the tank to shut-off the fuel flow when the engine is off. Since afterburner

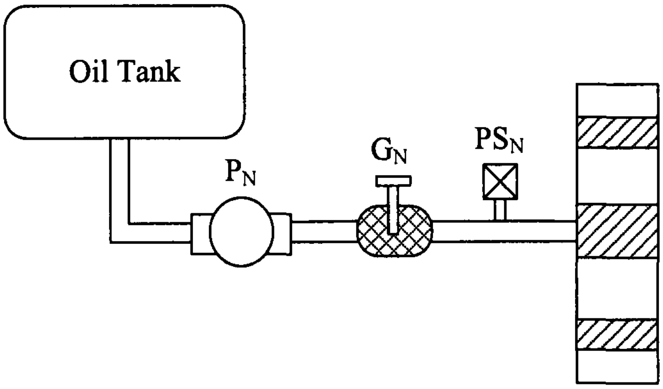


Figure 6.8: Schematic of a simplified typical nozzle actuator of a turbojet.

is not in operation at all times, there is also an on-and-off solenoid valve (V_{AB}) in the path of the fuel to the afterburner. The pressurizing valves (VP_M and VP_{AB}) ensure that the pressure in the pipe is sufficient for proper atomization in the combustion chamber and afterburner. Two (discrete) pressure sensors (PS_M and PS_{AB}) are placed after the governors to measure the pressure in the pipes. The pressure sensors have only two discrete outputs "low" and "high".

Usually, there is a mechanism that maintains a constant pressure at the two ends of the governors as long as the pressure in the pipes is sufficiently high. Therefore, the pressure at the ends of the governors can be assumed independent of the pressure in the pipes if the pressure in the pipes is above a certain threshold. As a result, the position of the metering device in the governor can be assumed to be proportional to the fuel flow rate passing through the governor.

Nozzle Actuator

We assume a hydraulic actuator for the nozzle variable area as shown in Figure 6.8. This actuator system consists of an oil tank, a pump (P_N), a discrete pressure sensor (PS_N) and a governor G_N. The pump provides the required pressure at both ends of the governor. The governor operation is similar to that of the governors of the fuel supply system.

6.1.8 Operating regimes

In general, 12 operating regimes can be defined for a single-spool turbojet, namely [103].

1. Starting
2. Acceleration
3. Maximum Power
4. Afterburner On
5. Maximum Afterburner Thrust
6. Cruise Thrust
7. Idle Thrust
8. Over-speed Limit
9. Maximum Turbine Temperature
10. Maximum Compressor Pressure
11. Deceleration
12. Shutdown

The operating regimes are defined in terms of the logical steps in the engine operation from starting to shutdown. Evaluation of these operating regimes is important for engine control system.

In previous discussions, we have explained the operation of the engine and its components and actuator systems. We have also derived the differential equations governing the operation of the engine. Since the components of the fuel supply system and the nozzle actuator have a discrete-event behavior, the engine actuator systems can be described by DES models. In the next section, we develop DES models for the components of the engine actuator systems and develop a hybrid model for the entire engine by integrating the DES models of the actuator systems and the continuous differential equation models of the engine.

6.2 Hybrid Modeling of a Jet Engine

In this section, we explain the hybrid modeling of the engine. First, we explain the continuous system model of the engine in different operating regimes.

6.2.1 Continuous Models in Different Operating Regimes

As explained in the last section, a turbojet engine equipped with an afterburner has 12 operating regimes. We do not consider regimes 8, 9 and 10 in this work. We also assume similar dynamics for the Acceleration and the Deceleration operating regimes. For simplicity of the modeling and diagnosis discussions, in the rest of this chapter, we assume that the engine works at sea level atmospheric conditions, i.e., $T_a = 288$ K, $p_a = 1$ bar $= 10^5$ Pa. In many real-world situations, it may be required to test the engine in all its operating regimes at the sea level by the manufacturer. It is also important to note that the discussions in this work are for the case of a *manual control* of the engine by the pilot (or manufacturer in the case of factory tests).

Normally, the operating regimes of the engine are related to the PLA input from the pilot [103]. Some typical tables exhibiting the range of the PLA input for different operating regimes of the engine can be found in [103].

We partition the complete range of the PLA into 10 regions each of which corresponding to an operating regime. We call each region an *operating section* identified by a number from 0 to 9. Table 6.1 shows the operating regimes considered

Operating section	Range of the PLA	Operating Regime(s)
0	0	Shutdown
1	$0^o - 10^o$	Starting
2	$10^o - 20^o$	Idle Thrust
3	$20^o - 30^o$	Acceleration1/ Deceleration3
4	$30^o - 40^o$	Acceleration2/Deceleration2
5	$40^o - 50^o$	Acceleration3/ Deceleration1
6	$50^o - 60^o$	Cruise Thrust
7	$60^o - 70^o$	Maximum Power
8	$70^o - 80^o$	Afterburner On
9	$80^o - 90^o$	Maximum Afterburner Thrust

Table 6.1: Operating sections and corresponding operating regimes and range of the PLA.

in our work and their corresponding operating sections and their corresponding range of the PLA.

We develop a linear system model for each operating regime by linearizing the nonlinear system model about an operating point. The operating conditions (values of the continuous states T_{03}, p_{03} and N) for each operating section correspond to the value of the PLA in the middle of the range, and are calculated using the nonlinear dynamics. Therefore, the resulting system will be a piece-wise linear system approximation of the nonlinear model.

Assuming that the system starts when the engine is in the "Shutdown" operating regime, the system transition among different operating sections is shown in Figure 6.9. The FSA shown in Figure 6.9 is the model of the engine at the regime level. The transitions among the states in Figure 6.9 occur autonomously when the engine transitions from one operating section to another. We have labeled these transitions by appropriate events such as '*start2idle*', etc.

Since we did not have access to the manufactures characteristic maps for the turbine and the compressor, we have assumed that the compressor (or turbine) behaves the same when the engine is accelerating and decelerating. In other words, the trajectory of the compressor pressure ratio versus speed (or mass flow rate of

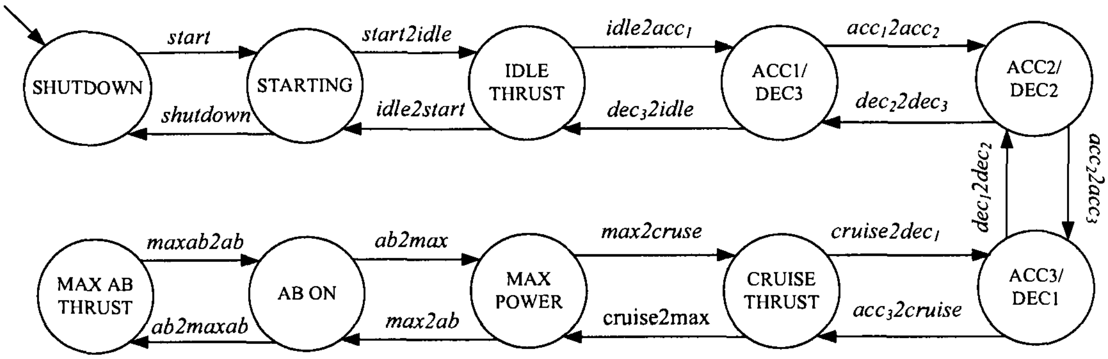


Figure 6.9: The FSA modeling the sequence of the operating regimes of the engine.

the fluid) is the same for both acceleration and deceleration of the engine.

Engine Control Approach

There are many types and modifications of control characteristics for jet engines. However, the control system used on aircraft engines usually falls into one of the following categories [103], namely:

1. Two positions (on/ off)

2. Open-loop scheduling

3. Proportional

4. Integrating

5. Proportional plus integral

6. Pulse integrating

As explained earlier, in our work, the values of the control inputs are functions of the PLA. In other words, we have adopted the "open-loop scheduling" as the control algorithm for the engine. The open-loop scheduling is not the best control algorithm and has many disadvantages. However, it simplifies the modeling and allows us to concentrate more on the diagnosis problem in the engine.

In the following, we specify the faults in the components of the actuator systems.

6.2.2 Faults Considered in the Actuator Systems

Table 6.2 shows the faults of the actuator systems (Figure 6.7 and Figure 6.8) that are studied in our work, the fault events modeling the occurrence of the faults and the failure modes corresponding to the faults. We assume that the faults are permanent, i.e, they remain in the engine indefinitely when they occur.

Fault	Component	Fault Event	Failure Mode
Stuck-closed of the main shut-off valve	V_S	$\hat{f}^1$	F^1
Stuck-closed of the afterburner shut-off valve	V_{AB}	$\hat{f}^2$	F^2
Main fuel pump is not energized	P_M	$\hat{f}^3$	F^3
Afterburner pump is not energized	P_{AB}	$\hat{f}^4$	F^4
Nozzle actuator pump is not energized	P_N	$\hat{f}^5$	F^5
Up to 10 % loss-of-effectiveness of the main fuel governor	G_M	$\hat{f}^6$	F^6
Oversupplying of the main fuel governor	G_M	$\hat{f}^7$	F^7
Up to 10 % loss-of-effectiveness of the afterburner governor	G_{AB}	$\hat{f}^8$	F^8
Oversupplying of the afterburner governor	G_{AB}	$\hat{f}^9$	F^9
Up to 10 % loss-of-effectiveness of the nozzle actuator governor	G_N	$\hat{f}^{10}$	F^{10}
Oversupplying of the nozzle actuator governor	G_N	$\hat{f}^{11}$	F^{11}

Table 6.2: The faults studied in the engine and their corresponding fault events and failure modes.

In this work, we have assumed that a discrete pressure sensors generate a "high" symbol if the pressure in the pipe is above 90% of its nominal value. Therefore, we study loss-of-effectiveness faults of up to 10% in the governors. We have also assumed that the pressurizing valves and the discrete pressure sensors are fault-free.

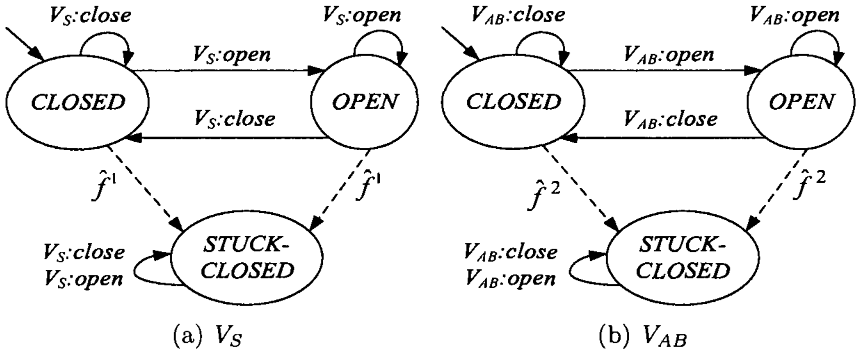


Figure 6.10: The FSA modeling the shutoff valves V_S and V_{AB}.

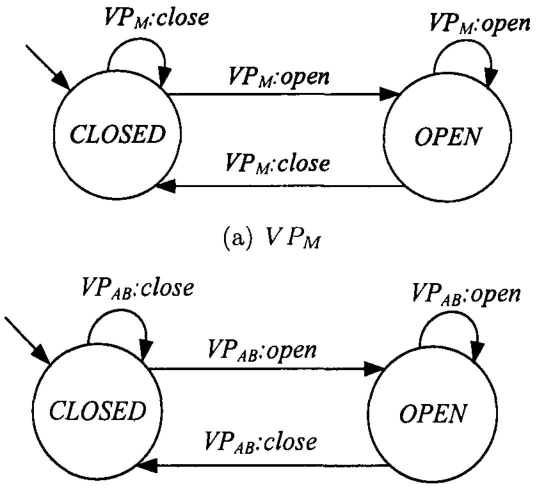
6.2.3 DES Models of the Actuator Systems

Components of the actuator systems exhibit a discrete-event behavior. For example, the pumps are either on or off. The solenoid shut-off valves are either open or closed. The discrete-event nature of these components enables us to model the components by DES models. In the following, we explain the DES modeling of the components of the actuator systems for both normal and faulty mode of operation. In the following, unobservable events are shown by dashed-lines in the automata models.

Valves

Figure 6.10 shows the automata modeling the shutoff valves V_S and V_{AB}. The fault events are unobservable and are shown by dashed lines. We have considered "stuck-closed" fault for these valves. The events 'V_S : *open*', 'V_S : *close*', 'V_{AB} : *open*' and 'V_{AB} : *close*' are commands (controllable events) generated by a supervisor. These commands are generated based on the values of the PLA. We explain the generation of these commands in more detail later in this section.

Figure 6.11 illustrates the automata models of the pressurizing valves VP_M and VP_{AB}. For simplicity, these valves are assumed fault-free. The events 'VP_M : *close*', 'VP_M : *open*', 'VP_{AB} : *close*' and 'VP_{AB} : *open*' are controllable events generated by the supervisor.



(a) VP_M

(b) VP_{AB}

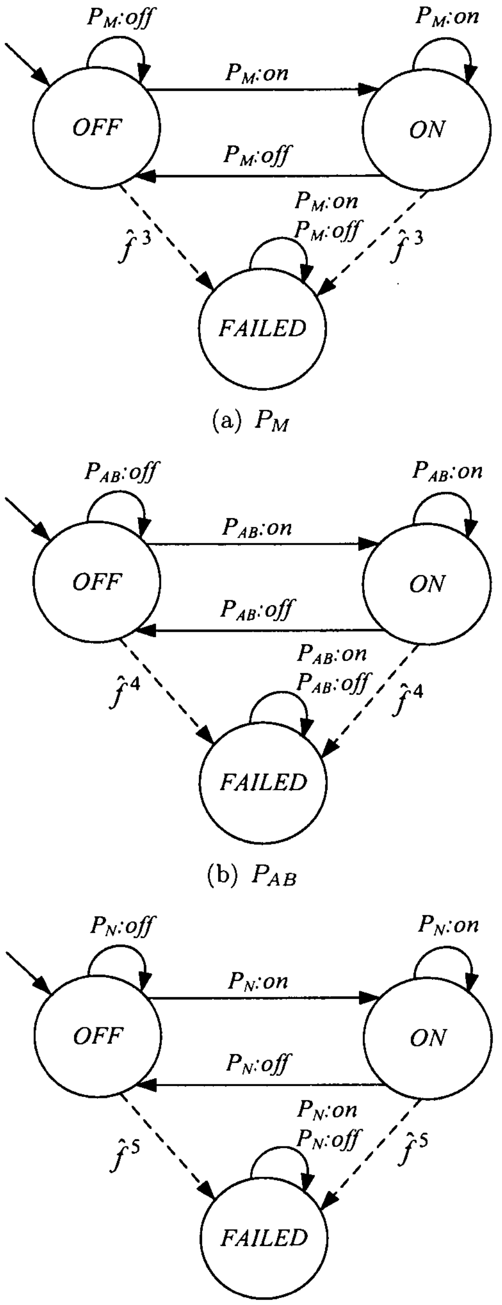
Figure 6.11: The FSA modeling the pressurizing valves VP_M and VP_{AB}.

Pumps

Figure 6.12 shows the automata models of the pumps in the engine. When the pumps become faulty, they are not energized when they are turned on or they are turned off if they are in operation. The events '$P_M : off$', '$P_M : on$', '$P_{AB} : off$', '$P_{AB} : on$', '$P_N : off$' and '$P_N : on$' are controllable events generated by the supervisor.

Pressure sensors

Pressure sensors measure the pressure in the pipes. Usually there are one or more redundant sensors that are used to increase the reliability of the measurements. Therefore, we assume that pressure sensors are fault-free. Pressure sensors generate a discrete output based on the pressure in the pipes. For efficient operation of the governors, the pressure in the pipes has to be above a threshold T_r. Moreover, the pressurizing valves in the main fuel supply and the afterburner fuel supply become open only if the pressure in the corresponding pipe is above T. Initially, the pressure in the pipes are below the threshold T_r and the pressure sensors generate "low" as the output. When the pressure in a pipe passes T, the pressure sensor installed in

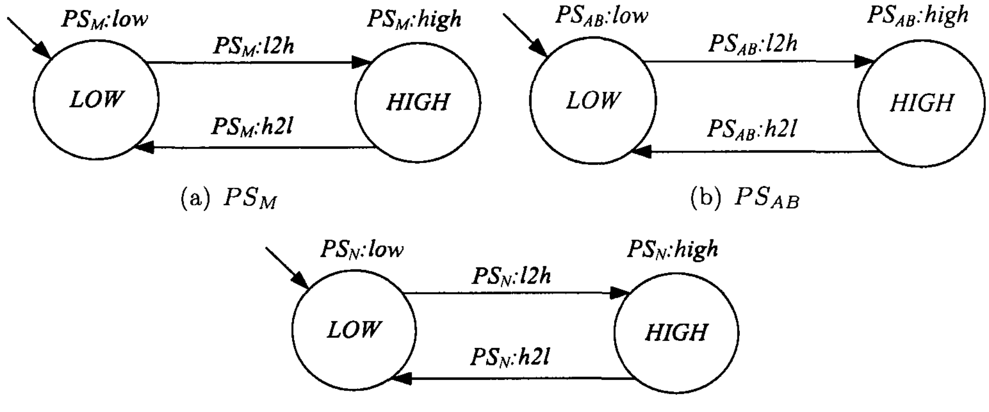


(a) P_M

(b) P_{AB}

(c) P_N

Figure 6.12: The FSA modeling the pressurizing pumps P_M, P_{AB} and P_N.



(a) PS_M

(b) PS_{AB}

(c) PS_N

Figure 6.13: The FSA modeling the discrete pressure sensors PS_M, PS_{AB} and PS_N.

that pipe generates the discrete output "high". Figure 6.13 illustrate the automata models of the pressure sensors. All events in the models of the pressure sensors are uncontrollable.

Governors

The governor is a servo valve. As explained in Subsection 6.1.7, there is a mechanism (not shown in this work) that maintains a constant pressure at both ends of a governor as long as the pressure is sufficiently high (above the threshold T_r) in the pipe. Therefore, the amount of the fluid passing through the governor will be a function of the effective area of the governor. The effective area of a governor is controlled by a metering rod (metering device) which is actuated by a servomechanism. The servomechanism can be either an electronic system or a mechanical system. In both cases, the position of the metering rod in a governor is a function of the PLA. Figures 6.14, 6.15 and 6.16 show the automata modeling the governors.

When more fuel/oil is needed, the position of the metering rod changes continuously to increase the effective area of the governor. This state of the governor is refereed to as acceleration (shown as "ACC" in the models). When the amount of fluid becomes sufficient, the metering rod stops. This static state of the governor

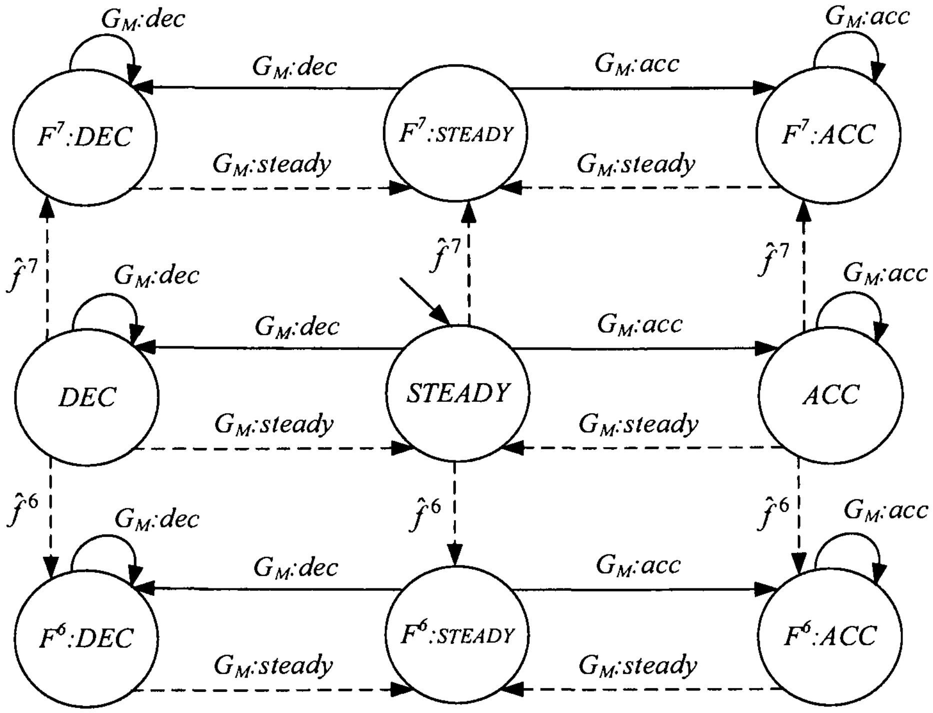


Figure 6.14: The FSA modeling the governors G_M.

is called steady (shown as "STEADY" in the models). When less fuel/oil is needed, the position of the rod changes to decrease the flow rate. This state of the governor in motion is refereed to as deceleration (shown as "DEC" in the models). The events 'G_M : *acc*', 'G_M : *dec*', 'G_{AB} : *acc*', 'G_{AB} : *dec*', 'G_N : *acc*' and 'G_N : *dec*' are the commands generated by the supervisor to change the position of the rod. The events 'G_M : *steady*', 'G_{AB} : *steady*' and 'G_N : *steady*' are uncontrollable and unobservable events modeling the stopping of the metering rod in G_M, G_{AB} and G_N, respectively.

Supervisory controller

Figure 6.17 illustrates the sequence of the events generated by the supervisory controller. The FSA enforcing this sequence is designated by *SequenceController*.

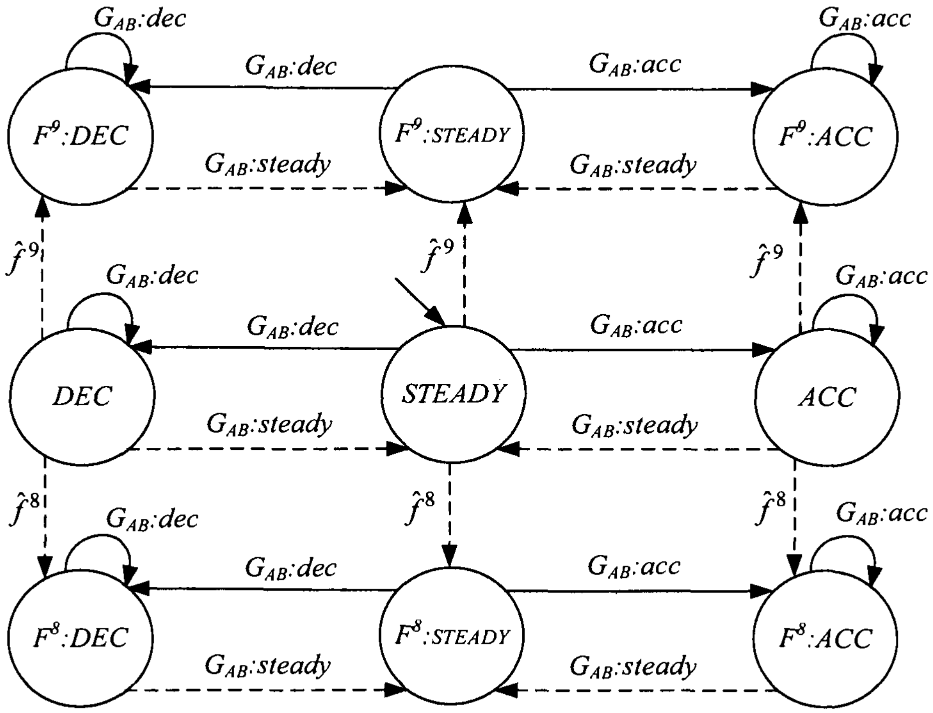


Figure 6.15: The FSA modeling the governors G_{AB}.

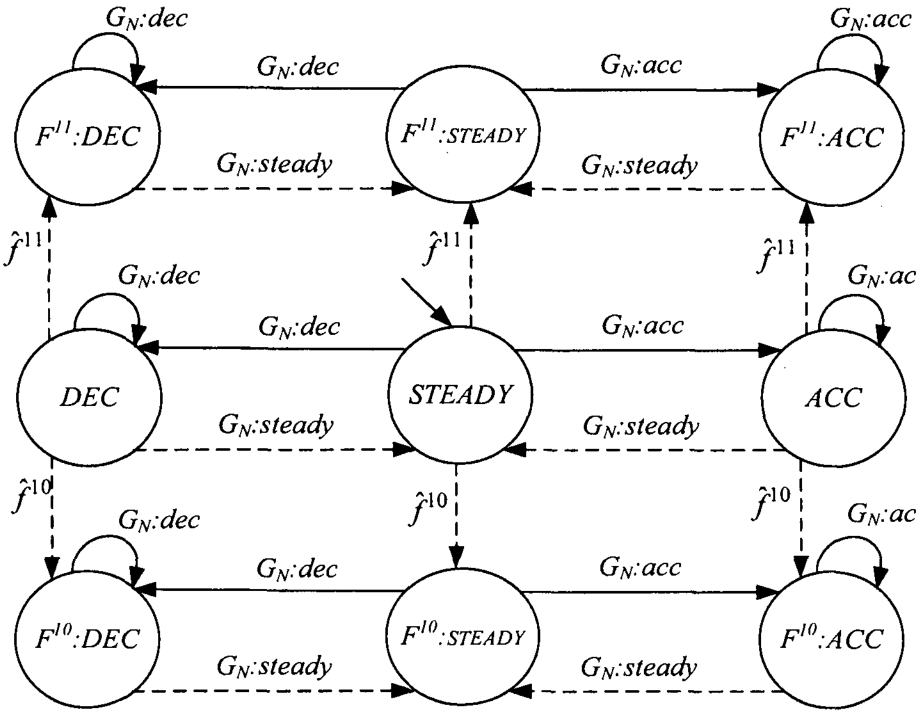


Figure 6.16: The FSA modeling the governors G_N.

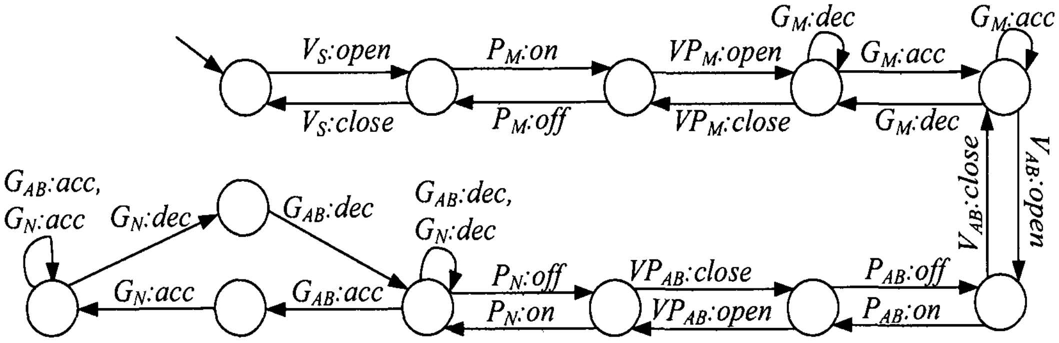


Figure 6.17: The FSA modeling the sequence of the events generated by the supervisory controller: *SequenceController*.

Interactions among the components

The following interactions are present among the components. The interactions can be modeled by FSA.

- The pressure in the pipe of the main fuel system passes the threshold T_r only if the valve V_S is open and the pump P_M is operational, and it goes below T if V_S becomes closed or fails stuck-closed, or P_M is turned off or fails. The FSA $V_S P_M IntPS_M$ shown in Figure 6.18 models this interaction. The pressure in the pipe of the afterburner fuel system passes the threshold T_r only if the valves V_S and V_{AB} are open and the pump P_{AB} is turned on, and it goes below T if V_S or V_{AB} become closed or fail stuck-closed, or P_{AB} is turned off or fails. The FSA $V_S V_{AB} P_{AB} IntPS_{AB}$ shown in Figure 6.19 models this interaction. Moreover, the pressure in the pipe of the nozzle actuator passes the threshold T_r only if the pump in the pipe is operational, and it goes below T if the pump is turned off or fails. The FSA $P_N IntPS_N$ shown in Figure 6.20 models the interaction between the pump and the pressure sensor in the nozzle actuator.

- A pressurizing valve becomes open only if the pressure in the pipe where it is installed is above the threshold T_r. The FSA $VP_M IntPS_M$ and $VP_{AB} IntPS_{AB}$ shown in Figure 6.21 enforce this interaction for the set of the pressurizing

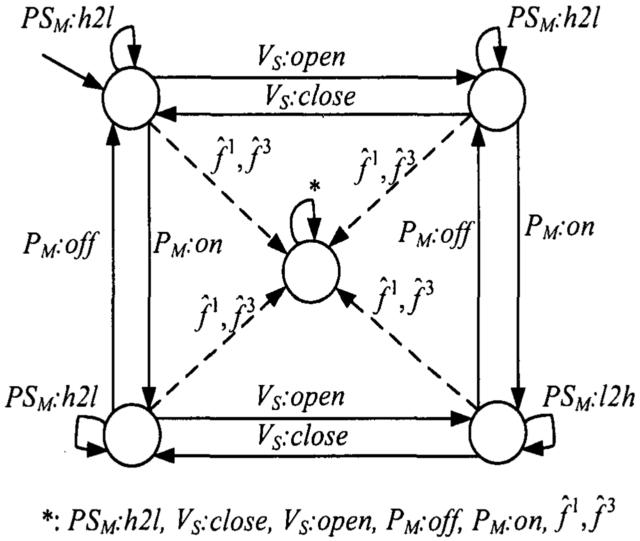


Figure 6.18: The FSA $V_SP_MIntPS_M$ modeling the interaction among V_S, P_M and PS_M.

valve and the sensor in the main fuel supply system and in the afterburner fuel supply system, respectively.

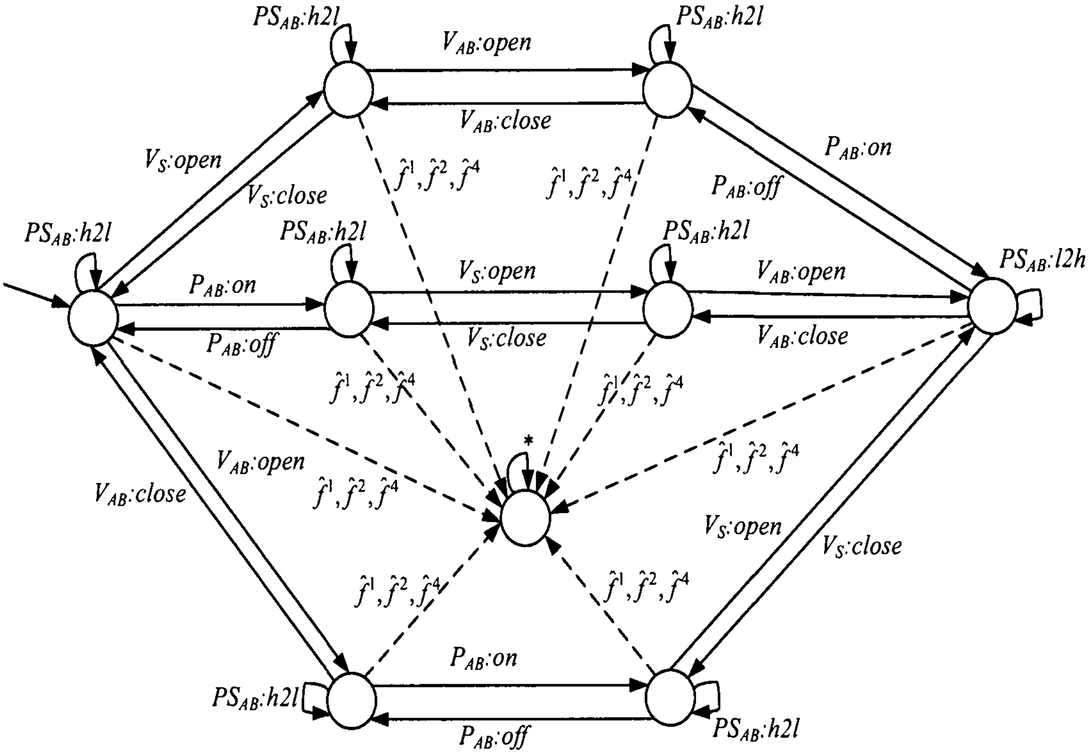
Automata models of the actuator systems

The FSA modeling the actuator systems can be obtained by integrating the models of the components and their interactions and the model of the sequence controller using the synchronous product operator. Let $ActuatorDES$ denote the FSA modeling the actuator systems. $ActuatorDES$ can be obtained as

$$ActuatorDES = \mathbf{sync}(ComponentDES, InteractionsDES, SequenceController)$$

where

$$ComponentDES = \mathbf{sync}(V_S, V_{AB}, VP_M, VP_{AB}, P_M, P_{AB}, P_N, PS_M, PS_{AB}, PS_N, G_M, G_{AB}, G_N)$$



*: PS_{AB}:h2l, V_S:close, V_S:open, V_S:close, V_S:open, P_{AB}:off, P_{AB}:on, $\hat{f}^1, \hat{f}^2, \hat{f}^4$

Figure 6.19: The FSA $V_S V_{AB} P_{AB} Int PS_{AB}$ modeling the interaction among V_S, V_{AB}, P_{AB} and PS_{AB}.

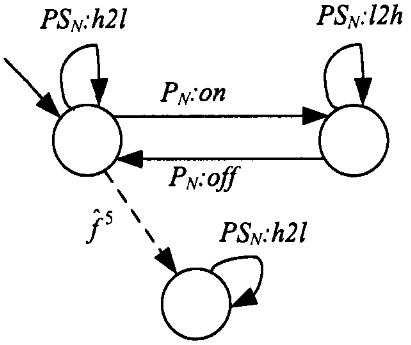
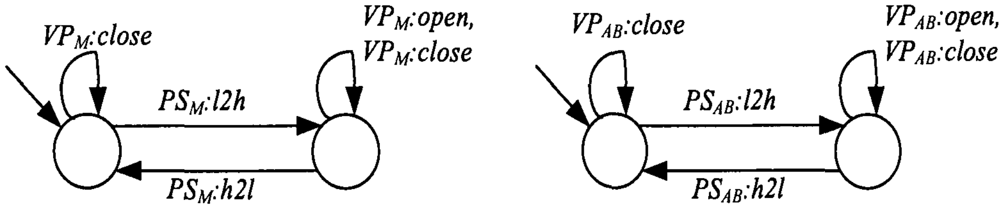


Figure 6.20: The FSA $P_N Int PS_N$ modeling the interaction between P_N and PS_N.



(a) $VP_M IntPS_M$ modeling the interaction between VP_M and PS_M

(b) $VP_{AB} IntPS_{AB}$ modeling the interaction between VP_{AB} and PS_{AB}

Figure 6.21: The FSA ensuring that the pressurizing valves do not open unless the pressure in the pipes is sufficiently high.

and

$$InteractionsDES = \textbf{sync}(V_S P_M IntPS_M, V_S V_{AB} P_{AB} IntPS_{AB}, P_N IntPS_N,$$
$$VP_M IntPS_M, VP_{AB} IntPS_{AB})$$

6.2.4 Hybrid Modeling

The discrete-event evolution of the components of the actuator systems is related to the operating regime of the engine. The status of the valves, pumps and the effective area of the governors change when the operating regime of the engine changes. Tables 6.3 - 6.10 present the state transitions of the components of the actuator systems that must occur when the operating regime of the engine changes. We assume that initially when the engine is shutdown, the valves are closed, the pumps are off, the pressure in the pipes are below the threshold T_r and the governors are at their STEADY discrete states.

It should be noted that while the engine evolves in an operating regime, the governors may have transitions '$STEADY \rightarrow ACC \rightarrow STEADY$' and '$STEADY \rightarrow DEC \rightarrow STEADY$'.

The FSA *EngineSupervisor* shown in Figure 6.22 models the engine supervisor. Each state in *EngineSupervisor* specifies an operating regime. The events of

<table>
<tr><th colspan="2">Regime Transition:
Shutdown to Starting</th></tr>
<tr><th>Component</th><th>State Transition</th></tr>
<tr><td>V_S</td><td>$CLOSED \rightarrow OPEN$</td></tr>
<tr><td>V_{AB}</td><td>$CLOSED$</td></tr>
<tr><td>P_M</td><td>$OFF \rightarrow ON$</td></tr>
<tr><td>P_{AB}</td><td>OFF</td></tr>
<tr><td>P_N</td><td>OFF</td></tr>
<tr><td>PS_M</td><td>$LOW \rightarrow HIGH$</td></tr>
<tr><td>PS_{AB}</td><td>LOW</td></tr>
<tr><td>PS_N</td><td>LOW</td></tr>
<tr><td>VP_M</td><td>$CLOSED \rightarrow OPEN$</td></tr>
<tr><td>VP_{AB}</td><td>$CLOSED$</td></tr>
<tr><td>G_M</td><td>$STEADY \rightarrow ACC \rightarrow STEADY$</td></tr>
<tr><td>G_{AB}</td><td>$STEADY$</td></tr>
<tr><td>G_N</td><td>$STEADY$</td></tr>
</table>

Table 6.3: State transition of the components of the actuator systems when the engine starts.

<table>
<tr><th colspan="2">Regime Transition:
Starting to Idle Thrust
Idle Thrust to Acceleration
Acceleration to Cruise Thrust to Maximum Power
Cruise Thrust</th></tr>
<tr><th>Component</th><th>State Transition</th></tr>
<tr><td>V_S</td><td>$OPEN$</td></tr>
<tr><td>V_{AB}</td><td>$CLOSED$</td></tr>
<tr><td>P_M</td><td>ON</td></tr>
<tr><td>P_{AB}</td><td>OFF</td></tr>
<tr><td>P_N</td><td>OFF</td></tr>
<tr><td>PS_M</td><td>$HIGH$</td></tr>
<tr><td>PS_{AB}</td><td>LOW</td></tr>
<tr><td>PS_N</td><td>LOW</td></tr>
<tr><td>VP_M</td><td>$OPEN$</td></tr>
<tr><td>VP_{AB}</td><td>$CLOSED$</td></tr>
<tr><td>G_M</td><td>$STEADY \rightarrow ACC \rightarrow STEADY$</td></tr>
<tr><td>G_{AB}</td><td>$STEADY$</td></tr>
<tr><td>G_N</td><td>$STEADY$</td></tr>
</table>

Table 6.4: State transition of the components of the actuator systems when the operating regime changes from Starting to Idle Thrust/ Idle Thrust to Acceleration/ Acceleration to Cruise Thrust/ Cruise Thrust to Maximum Power.

Regime Transition: Maximum Power to Afterburner On	
Component	State Transition
V_S	$OPEN$
V_{AB}	$CLOSED \rightarrow OPEN$
P_M	ON
P_{AB}	$OFF \rightarrow ON$
P_N	$OFF \rightarrow ON$
PS_M	$HIGH$
PS_{AB}	$LOW \rightarrow HIGH$
PS_N	$LOW \rightarrow HIGH$
VP_M	$OPEN$
VP_{AB}	$CLOSED \rightarrow OPEN$
G_M	$STEADY$
G_{AB}	$STEADY \rightarrow ACC \rightarrow STEADY$
G_N	$STEADY \rightarrow ACC \rightarrow STEADY$

Table 6.5: State transition of the components of the actuator systems when the operating regime changes from Maximum Power to Afterburner On.

Regime Transition: Afterburner On to Maximum Afterburner Thrust	
Component	State Transition
V_S	$OPEN$
V_{AB}	$OPEN$
P_M	ON
P_{AB}	ON
P_N	ON
PS_M	$HIGH$
PS_{AB}	$HIGH$
PS_N	$HIGH$
VP_M	$OPEN$
VP_{AB}	$OPEN$
G_M	$STEADY$
G_{AB}	$STEADY \rightarrow ACC \rightarrow STEADY$
G_N	$STEADY \rightarrow ACC \rightarrow STEADY$

Table 6.6: State transition of the components of the actuator systems when the operating regime changes from Afterburner On to Maximum Afterburner Thrust.

Regime Transition: Maximum Afterburner Thrust to Afterburner On	
Component	State Transition
V_S	*OPEN*
V_{AB}	*OPEN*
P_M	*ON*
P_{AB}	*ON*
P_N	*ON*
PS_M	*HIGH*
PS_{AB}	*HIGH*
PS_N	*HIGH*
VP_M	*OPEN*
VP_{AB}	*OPEN*
G_M	*STEADY*
G_{AB}	*STEADY* → *DEC* → *STEADY*
G_N	*STEADY* → *DEC* → *STEADY*

Table 6.7: State transition of the components of the actuator systems when the operating regime changes from Maximum Afterburner Thrust to Afterburner On.

Regime Transition: Afterburner On to Maximum Power	
Component	State Transition
V_S	*OPEN*
V_{AB}	*OPEN* → *CLOSED*
P_M	*ON*
P_{AB}	*ON* → *OFF*
P_N	*ON* → *OFF*
PS_M	*HIGH*
PS_{AB}	*HIGH* → *LOW*
PS_N	*HIGH* → *LOW*
VP_M	*OPEN*
VP_{AB}	*OPEN* → *CLOSED*
G_M	*STEADY*
G_{AB}	*STEADY*
G_N	*STEADY*

Table 6.8: State transition of the components of the actuator systems when the operating regime changes from Afterburner On to Maximum Power.

Regime Transition: Maximum Power to Cruise Thrust Cruise Thrust to Deceleration Deceleration to Idle Thrust Idle Thrust to Starting	
Component	State Transition
V_S	$OPEN$
V_{AB}	$CLOSED$
P_M	ON
P_{AB}	OFF
P_N	OFF
PS_M	$HIGH$
PS_{AB}	LOW
PS_N	LOW
VP_M	$OPEN$
VP_{AB}	$CLOSED$
G_M	$STEADY \rightarrow DEC \rightarrow STEADY$
G_{AB}	$STEADY$
G_N	$STEADY$

Table 6.9: State transition of the components of the actuator systems when the operating regime changes from Maximum Power to Cruise Thrust/ Cruise Thrust to Deceleration/ Deceleration to Idle Thrust/ Idle Thrust to Starting.

Regime Transition: Starting to Shutdown	
Component	State Transition
V_S	$OPEN \rightarrow CLOSED$
V_{AB}	$CLOSED$
P_M	$ON \rightarrow OFF$
P_{AB}	OFF
P_N	OFF
PS_M	$HIGH \rightarrow LOW$
PS_{AB}	LOW
PS_N	LOW
VP_M	$OPEN \rightarrow CLOSED$
VP_{AB}	$CLOSED$
G_M	$STEADY \rightarrow DEC \rightarrow STEADY$
G_{AB}	$STEADY$
G_N	$STEADY$

Table 6.10: State transition of the components of the actuator systems when the operating regime changes from Starting to Shutdown.

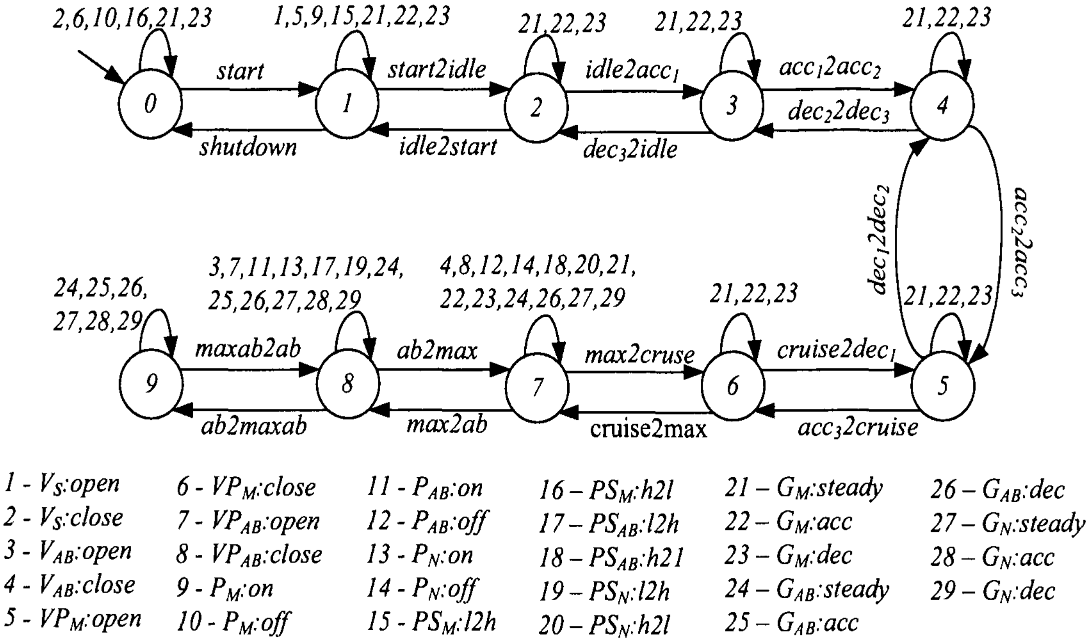


Figure 6.22: The FSA modeling the engine supervisor: *EngineSupervisor*.

the actuator components that can occur in each operating regime are represented as self-loop events in that operating regime.

The DES abstraction of the engine can be obtained by the synchronous product of the FSA modeling the actuator systems and the FSA modeling the interactions between the actuator systems and the operating sections. Let the FSA

$$H_{abs}^{Eng} = (Q_{abs}^{Eng}, \Sigma_{abs}^{Eng}, T_{abs}^{Eng}, D_{abs}^{Eng}, \lambda_{abs}^{Eng}, q_{abs,0}^{Eng})$$

denote the DES abstraction of the engine. We have

$$H_{abs}^{Eng} = \mathbf{sync}(ActuatorDES, EngineSupervisor)$$

The hybrid automaton modeling the engine is a tuple $H^{Eng} = (Q^{Eng}, \mathcal{X}^{Eng}, \mathcal{U}^{Eng},$

$\mathcal{FT}^{Eng}, \mathcal{Y}^{Eng}, Init^{Eng}, S^{Eng}, \Sigma^{Eng}, \mathcal{T}^{Eng}, G^{Eng}, \rho^{Eng}, D^{Eng}, \lambda^{Eng}, q_0^{Eng})$ where

$$\begin{aligned}
&\mathcal{X}^{Eng}, \mathcal{U}^{Eng}, \mathcal{FT}^{Eng}, \mathcal{Y}^{Eng}, Init^{Eng} \subset \mathbb{R}^3; \\
&Q^{Eng} = Q_{abs}^{Eng}; \\
&\Sigma^{Eng} = \Sigma_{abs}^{Eng}; \\
&\mathcal{T}^{Eng} = \mathcal{T}_{abs}^{Eng}; \\
&D^{Eng} = D_{abs}^{Eng}; \\
&\lambda^{Eng} = \lambda_{abs}^{Eng};
\end{aligned} \tag{6.42}$$

and

$$q_0^{Eng} = q_{abs,0}^{Eng} \tag{6.43}$$

Function ρ^{Eng} is a unity reset map that maps the states of the engine to themselves. Function G^{Eng} becomes true when the boundaries of the partitions corresponding to the operating sections are passed, and S is a set of linear systems each modeling the engine dynamics at a discrete state.

Totally, there are 13 components in the actuator systems. Each discrete state of H^{Eng} can be represented by a 14-tuple $q = (q_1, \cdots q_{14})$, where each of $q_1, \cdots, q_{13}$ corresponds to a component model and q_{14} specifies the operating regime of the engine in q.

All the faults considered in this work affect the control input signals. Hence, the faults can be modeled by additive faut type signals. Let f^1, f^2 and f^3 be the fault type signals modeling the faults in main fuel supply system, afterburner fuel supply system and nozzle actuator, respectively. Therefore, we have $FT^{Eng} = \{f^1, f^2, f^3\}$, $\xi(f^1) = \{F^1, F^3, F^6, F^7\}$, $\xi(f^2) = \{F^1, F^2, F^4, F^8, F^9\}$ and $\xi(f^3) = \{F^5, F^{10}, F^{11}\}$.

The dynamics of the engine in the discrete state q, S_q, is represented by a linear system model

$$S_q = \begin{cases} \dot{x} = A_q x + B_q u + L_q f \\ y = C_q x \end{cases} \tag{6.44}$$

where A_q, B_q and C_q are the system matrices and belong to one of the operating sections $i \in \{0, \cdots, 9\}$, $f = [f^1, f^2, f^3]^T$, and L_q is the vector of the fault signatures in the discrete state q. For simplicity, in this work we take $L_q = B_q$. Totally, there are 10 sets of EM-similar (here ABC-similar) discrete states each corresponding to an operating section. Let Q^i be the set of ABC-similar discrete states corresponding to the operating section i. We have $Q^{Eng} = Q^0 \bigcup \cdots \bigcup Q^9$. The set Q^i includes all the discrete states whose A, B and C matrices are the same as those of the operating section i. Fault signatures at the discrete states of Q^i can be different.

In the following section, we explain our proposed hybrid diagnoser design for the hybrid system model of the engine.

6.3 Fault Diagnosis in Jet Engines

First we explain isolator (residual generator) design based on the continuous dynamics of the engine.

6.3.1 Isolator Design

The continuous dynamics of the engine at any discrete state of the hybrid model is represented by a linear system. The A, B and C matrices of the system at any discrete state belong to the dynamics of one of the operating sections. We have assumed that the fault signature at any faulty discrete state is equal to the B matrix of that discrete state. Therefore, we develop a bank of isolators based on the dynamics of the operating sections.

As explained in Section 6.1, the measurable variables in our work are T_{03}, p_{03} and N. These output variables change when the input variables $\dot{m}_f$, $\dot{m}_{fAB}$ and A_n change. It can be shown that the input variables $\dot{m}_{fAB}$ and A_n have opposite impact on the output variables. For example, when the mass flow rate of the afterburner fuel increases the turbine inlet temperature T_{03} increases. But when A_n increases, T_{03} decreases. We also observe that the effect of these two inputs on the outputs (for the same value of inputs) are proportional to each other. In other words, matrix B in the developed linear system models is not full rank, i.e.,

$$rank(B_q) = 2, \text{ for all } q \in Q^{Eng}$$

More precisely,

$$rank([b_q^2 b_q^3]) = 1$$

where b_q^2 and b_q^3 are the vectors corresponding to $\dot{m}_{fAB}$ and A_n, respectively, in matrix B_q.

Since we have $L_q = B_q$ in the dynamics of faulty discrete states, the solvability conditions for the existence of the isolators discussed in Section 2.4 are satisfied for f^1 but not for f^2 and f^3. Fault types f^1, f^2 and f^3 are the signals modeling the faults in main fuel supply system, afterburner fuel supply system and nozzle actuator, respectively. Thus, we cannot design isolators that isolate faults in the afterburner fuel supply system from the faults in the nozzle actuator. However, the solvability conditions will be satisfied if f^2 and f^3 are considered together. In other words, we can design isolators that isolate faults in the main fuel supply system from faults in the afterburner fuel supply system and the nozzle actuator, and isolators that isolate faults in the afterburner fuel supply system and the nozzle actuator from faults in the main fuel supply system. Moreover, the states of the linear systems are all observable, and we can design full observers for the normal mode of operation in each operating section.

The discrete states of Q^0 correspond to the Shutdown operating regime. The dynamics of the engine in these discrete states is $\dot{x} = 0$. We have designed three isolators for each state set Q^i ($i \in \{1, \cdots, 9\}$). In total, we have a bank of 27 isolators for fault diagnosis at the continuous level. Each isolator will be modeled with an FSA. Let $Is^{Q^i}(\{f^1\})$ be the isolator deigned for Q^i to distinguish faults in the main fuel supply system from faults in the afterburner fuel supply system and the nozzle actuator. Also let $Is^{Q^i}\{f^2, f^3\})$ denote the isolator deigned to distinguish faults in the afterburner fuel supply system or the nozzle actuator from the faults in the main fuel supply system while the engine is in a discrete state of Q^i. Furthermore, let $Obs^i = Is^{Q^i}(\{f^1, f^2, f^3\})$ be the observer designed for the normal (non-faulty) dynamics of Q^i.

6.3.2 Constructing the EDESA of the Engine and Isolators

Let $\overline{Is}^{Q^i}(W)$ be the FSA modeling the isolator $Is^{Q^i}(W)$ for any $W \in \{\{f^1\}, \{f^2, f^3\}, \{f^1, f^2, f^3\}\}$. The EDESA developed for the engine H^{Eng} can be obtained from

$$\tilde{H}^{Eng} = \mathbf{sync}(\hat{H}^{Eng}_{abs}, \overline{Is}^{Eng}, ASM^{Eng})$$

where $\hat{H}^{Eng}_{abs}$ is the modified form of H^{Eng}_{abs} satisfying the consistency specifications (described in Section 3.2). Furthermore,

$$\overline{Is}^{Eng} = \mathbf{sync}(\overline{Is}^{Q^1}(\{f^1\}), \overline{Is}^{Q^1}(\{f^2, f^3\}), Obs^{Q^1}, \cdots \overline{Is}^{Q^9}(\{f^1\}), \overline{Is}^{Q^9}(\{f^2, f^3\}), Obs^{Q^9})$$

and

$$\begin{aligned} ASM^{Eng} = \mathbf{sync}(&ASM_{Is^{Q^1}(\{f^1\})}, ASM_{Is^{Q^1}(\{f^2,f^3\})}, ASM_{Obs^{Q^1}}, \cdots \\ &\cdots, ASM_{Is^{Q^9}(\{f^1\})}, ASM_{Is^{Q^9}(\{f^2,f^3\})}, ASM_{Obs^{Q^9}}) \end{aligned}$$

Here, the FSM components of ASM^{Eng} enforce the assumption that an event will be generated by an isolator between every two consecutive events generated by the system (described in Section 3.2).

The discrete output of the EDESA is an array with 30 elements. The first three elements comes from the discrete sensors and the rest comes from the isolators. A

Atmospheric Condition	Description	Value
H	Aircraft altitude	0
p_a	Ambient pressure	10^5 Pa (1Bar)
T_a	Ambient temperature	288K

Table 6.11: Atmospheric conditions.

DES diagnoser is designed for the EDESA of the engine and the isolators based on the method described in Section 2.2.

In the following section, we describe the simulation method and present the simulation results.

6.4 Simulation Results

We used the MATLAB/SIMULINK software for conducting the engine dynamics simulations. We used TTCT software [2] for performing DES simulations. The hybrid automaton model of the engine has $1,969,920$ discrete states and $13,882,752$ transitions.

6.4.1 Engine Parameters

Table 6.11 shows the atmospheric conditions used for the simulations. Table 6.12 shows the gas parameters and Table 6.13 shows the engine parameters used in our work. We assume that the ambient operating conditions and power settings do not change, and the aircraft is static at all times ($Ma = 0$).

6.4.2 Simulation Results

Engine dynamic simulations

First, we present the continuous output variables of the engine for a ramp PLA command and a staircase PLA command. Figure 6.23 shows a ramp PLA starting

Gas Parameter	Description	Value	Unit
R	Specific gas constant	287	$\frac{J}{KgK}$
c_{pa}	Specific heat capacity at constant pressure for air	1005	$\frac{J}{KgK}$
c_{va}	Specific heat capacity at constant volume for air	718	$\frac{J}{KgK}$
γ_a	Specific heat capacity ratio for air	1.4	–
c_{pg}	Specific heat capacity at constant pressure for the gases in the combustion chamber and afterburner	1148	$\frac{J}{KgK}$
c_{vg}	Specific heat capacity at constant volume for the gases in the combustion chamber and afterburner	861	$\frac{J}{KgK}$
γ_g	Specific heat capacity ratio for the gases in the combustion chamber and afterburner	1.333	–

Table 6.12: Gas parameters.

Engine Parameter	Description	Value	Unit
J	Rotor moment of inertia	7.80	KgM^2
V_{comb}	Volume of the main combustion chamber	0.4	M^3
A_n^{max}	Maximum of the nozzle area	1.2	M^2
$\dot{m}_f^{max}$	Maximum of the main fuel flow rate	0.8	$\frac{Kg}{sec.}-$
$\dot{m}_{fAB}^{max}$	Maximum of the afterburner fuel flow rate	2.4	$\frac{Kg}{sec.}$
H_u	Low calorific value of fuel (Kerosene)	47×10^6	$\frac{J}{Kg}$
Δp_b	Percentage of pressure loss in the main combustion chamber	4	–
η_i	Isentropic intake efficiency	0.95	–
η_c	Isentropic compressor efficiency	0.87	–
η_t	Isentropic turbine efficiency	0.90	–
η_j	Isentropic nozzle efficiency	0.95	–
η_m	Mechanical transmission efficiency of the turbine	0.99	–
η_b	Efficiency of the main combustion chamber	0.98	–
η_{AB}	Efficiency of the afterburner combustion chamber	0.65	–

Table 6.13: Engine parameters.

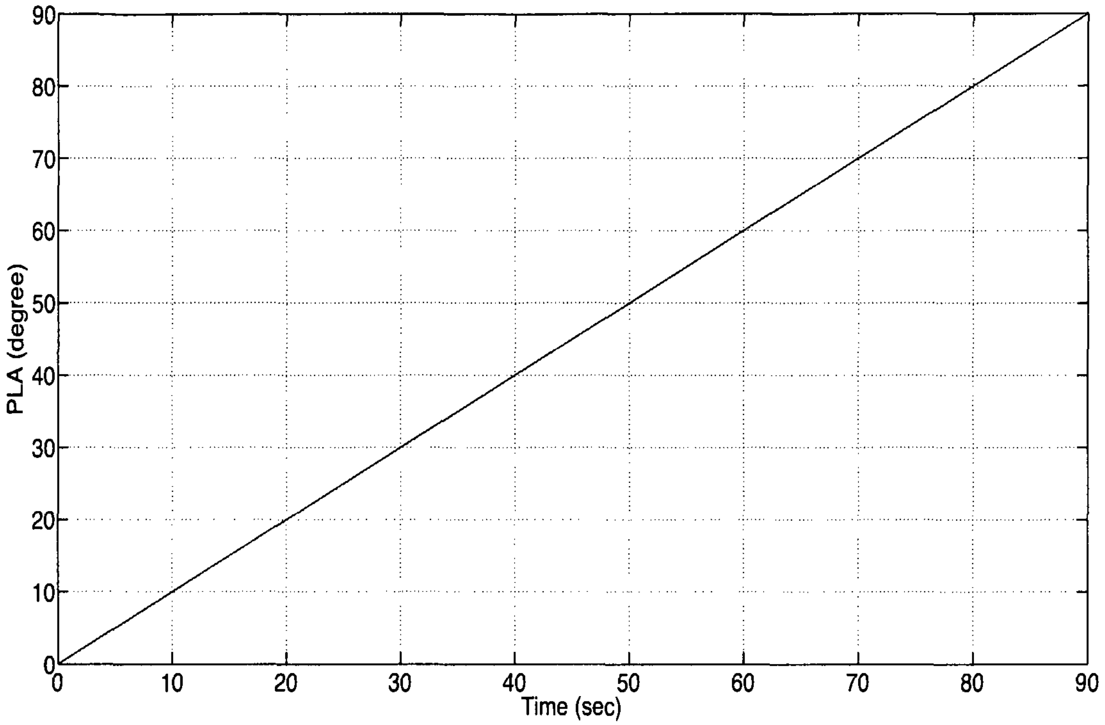


Figure 6.23: A ramp PLA command.

from zero to 90^o. The continuous output variables of the engine (N, T_{03} and p_{03}) generated by the nonlinear model are shown in Figures 6.24, 6.25 and 6.26. As it can be seen from these figures, the maximum nonlinearity is at low PLA values (the PLA below 10^o).

Figure 6.27 depicts a staircase PLA command. The responses of the nonlinear model are shown in Figures 6.28, 6.29 and 6.30.

The operation of the hybrid diagnoser

We present the simulation results for three fault scenarios. Although the isolators are designed based on the linearized models, in the simulations, we use the output of the nonlinear model as the input to the isolators. The hybrid diagnoser is constructed systematically using the EDESA model. Here, we explain the operation of the hybrid diagnosis for the three fault scenarios.

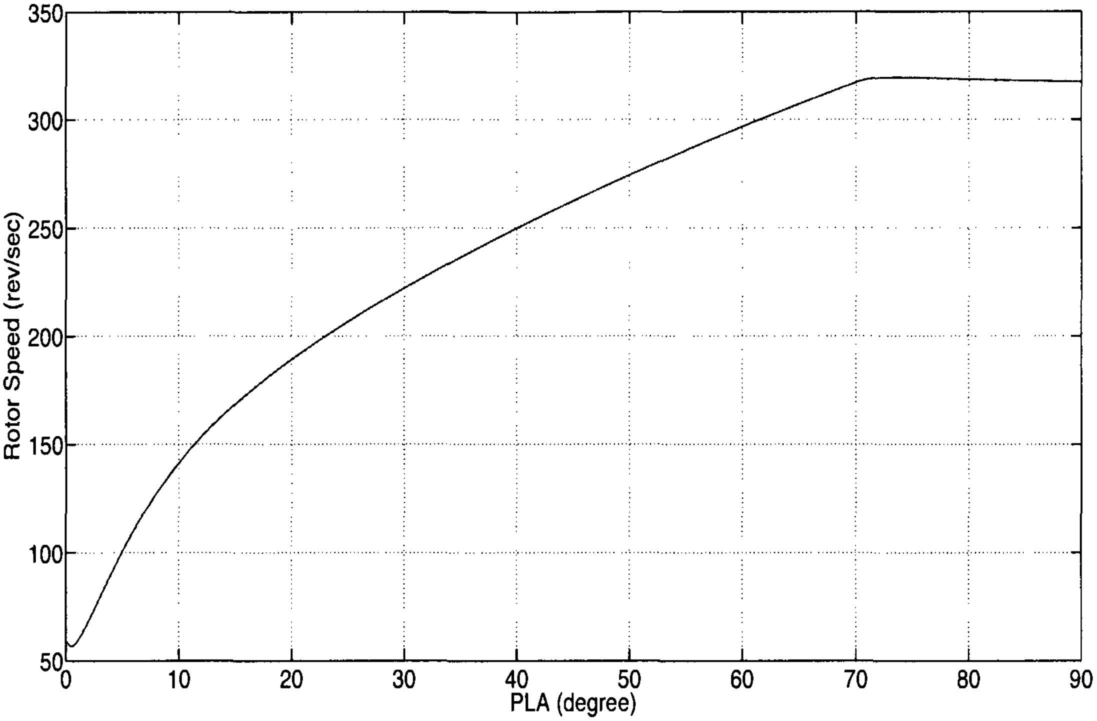


Figure 6.24: Rotor speed (N) for a ramp PLA command.

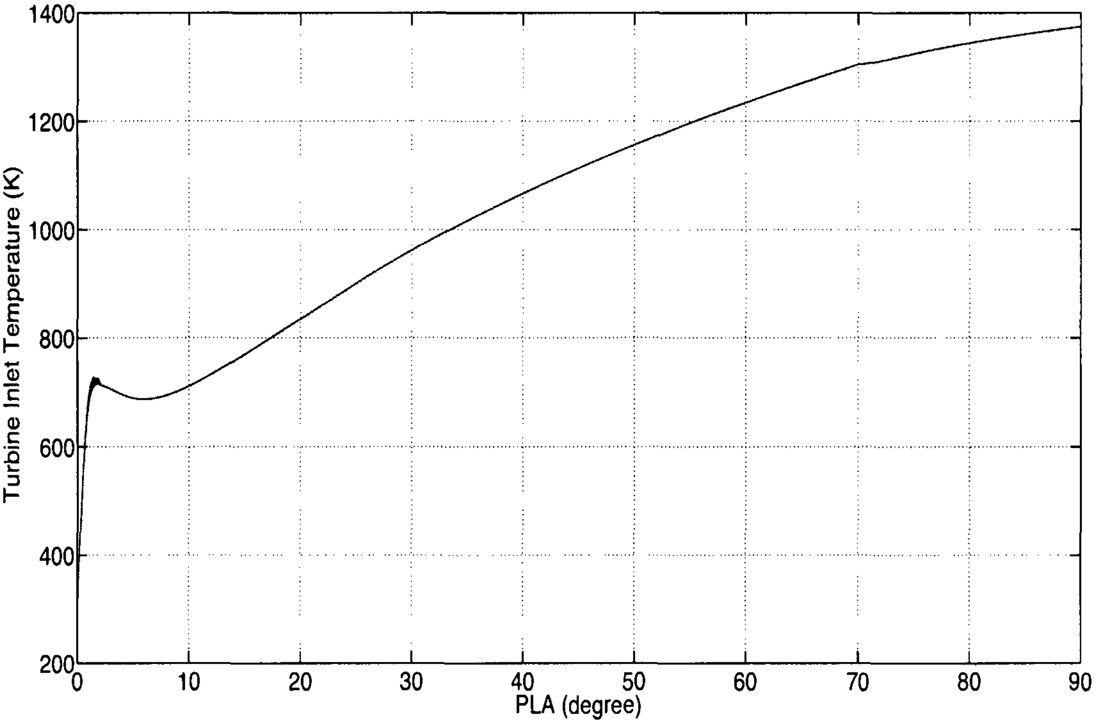


Figure 6.25: Turbine inlet temperature (T_{03}) for a ramp PLA command.

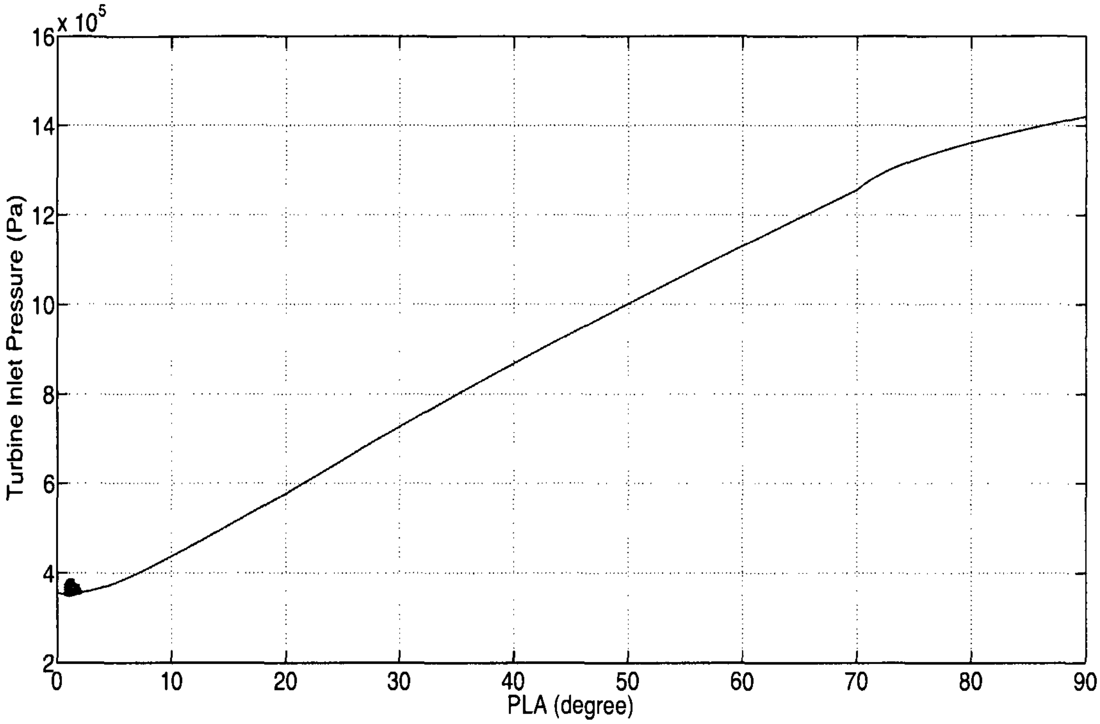


Figure 6.26: Turbine inlet pressure (p_{03}) for a ramp PLA command.

We used the PLA signal as shown in Figure 6.31 for generating the inputs for these fault scenarios. The continuous control input variables of the engine ($\dot{m}_f$, $\dot{m}_{AB}$ and A_n) in normal mode of operation for this PLA signal is showns in Figure 6.32.

Fault scenario 1: 5% loss-of-effectiveness of the main fuel system governor G_M applied at $t = 30$ *sec.*

Figure 6.33 shows the fault type signals corresponding to this fault scenario. Initially, all the fault type signals are zero. At $t = 30$ *sec.*, the value of the fault type corresponding to the main fuel system changes from zero to -0.04 which is 5% of the main fuel flow rate at $t = 30$ *sec.* The fault is assumed permanent. Therefore, the value of the fault type signal does not change during the rest of the simulation period.

Initially, we present the simulation results for the case that no measurement

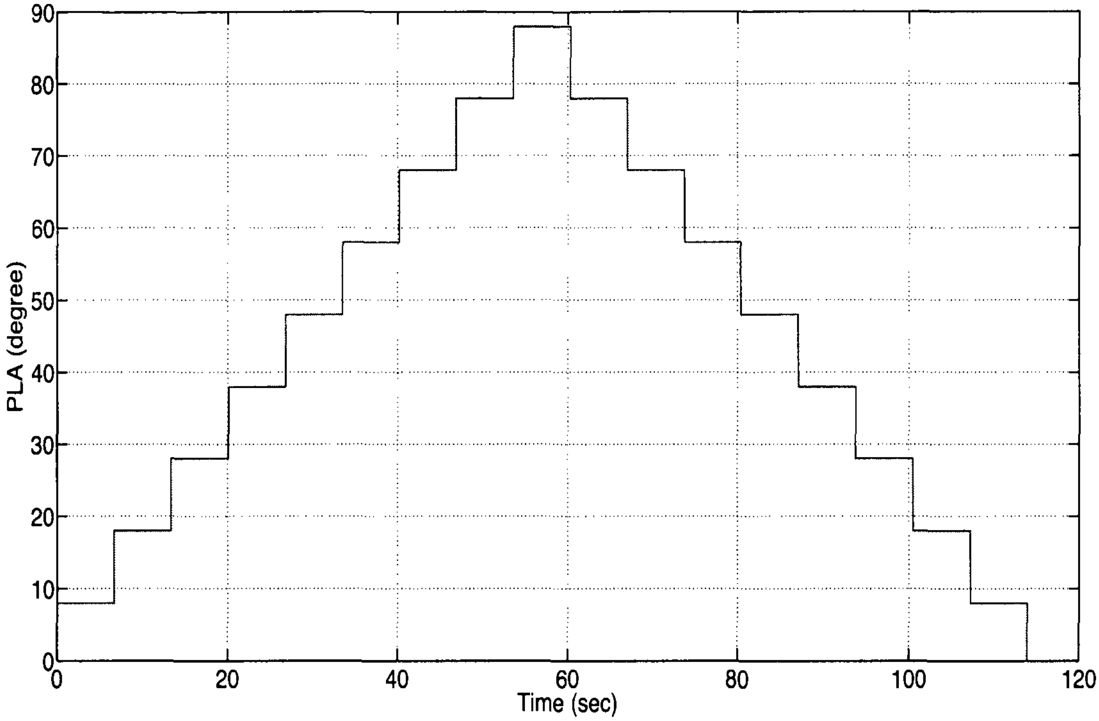


Figure 6.27: A staircase PLA command.

noise is present. The engine enters the operating regime 8 at $t = 9.33$ *sec.*. It can be shown that from $t \simeq 9.33$ *sec.* (when the engine enters the operating regime 8) to $t = 50$ *sec.*, the response of all isolators except the isolators designed for the operating section 8 (i.e, $Is^8(\{f^1\})$, $Is^8(\{f^2, f^3\})$, Obs^8) are one. This implies that the engine is in the operating section 8 (corresponding to the operating regime Afterburner On) during this interval, and no fault has occurred. Figure 6.34 shows the response of the isolators designed for the operating section 8. The isolator $Is^8(\{f^1\})$ generates zero at $t \simeq 9.71$ *sec.* implying that the engine is in the operating section 8. Response of the isolators designed for the operating sections 2 and 5 are shown, as an example of the output of other isolators, in Figures 6.35 and 6.35. The fault is applied at $t = 30$ *sec.* and is detected at $t = 30.17$ *sec.* From $t = 30.17$ *sec.* to $t = 50$ *sec.*, the output of $Is^8(\{f^1\})$ changes to one but the output of $Is^8(\{f^2, f^3\})$ remains at zero implying that the fault type f^1 is active and f^2 or f^3 have not

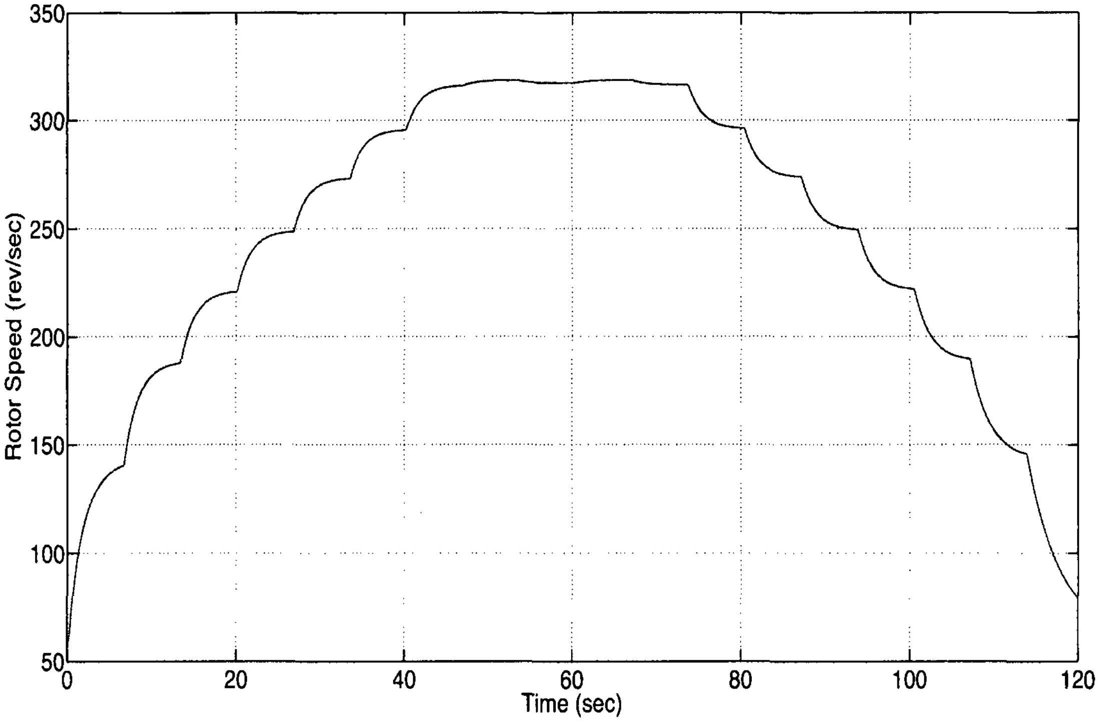


Figure 6.28: Rotor speed (N) for a staircase PLA command.

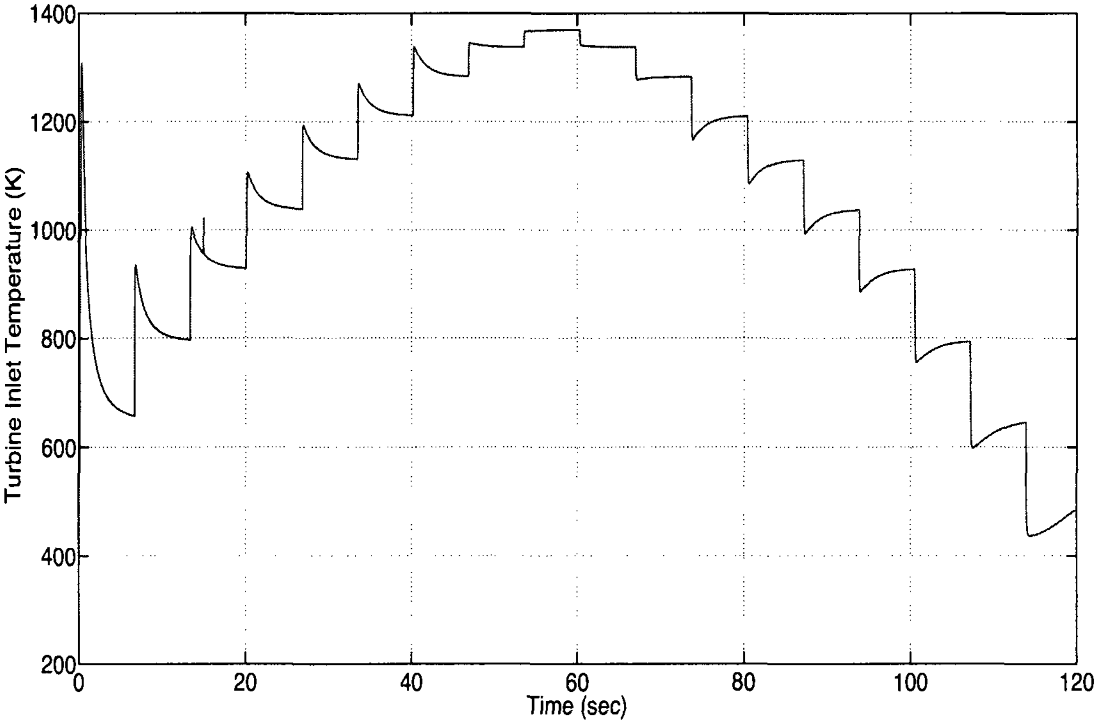


Figure 6.29: Turbine inlet temperature (T_{03}) for a staircase PLA command.

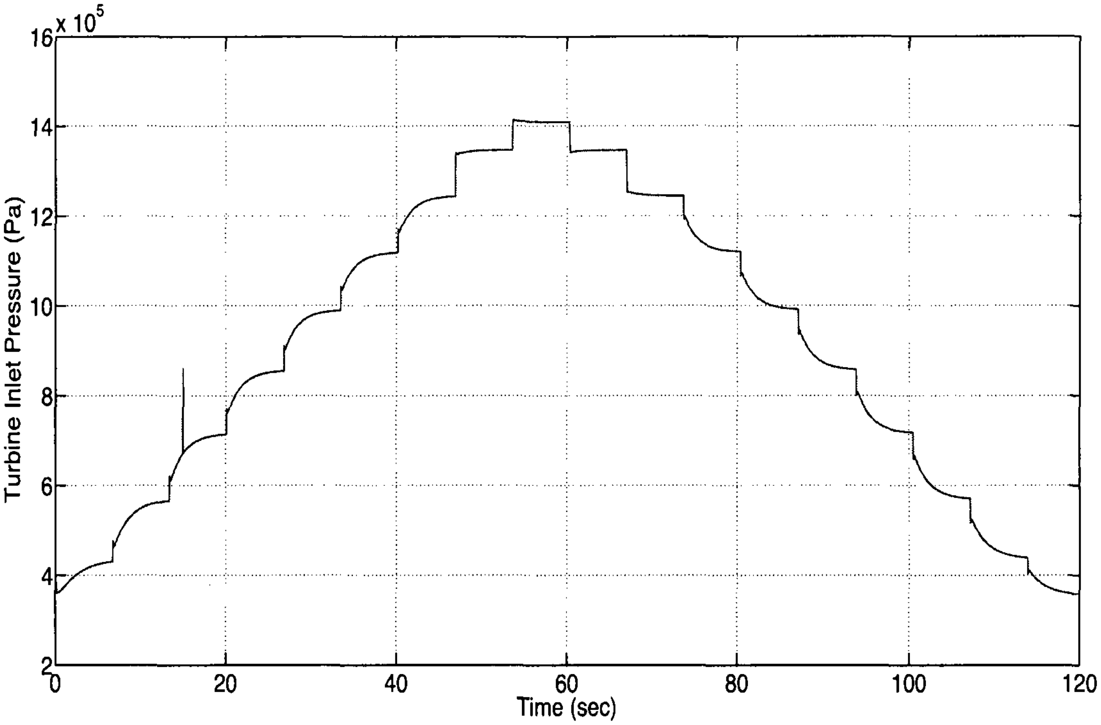


Figure 6.30: Turbine inlet pressure (p_{03}) for a staircase PLA command.

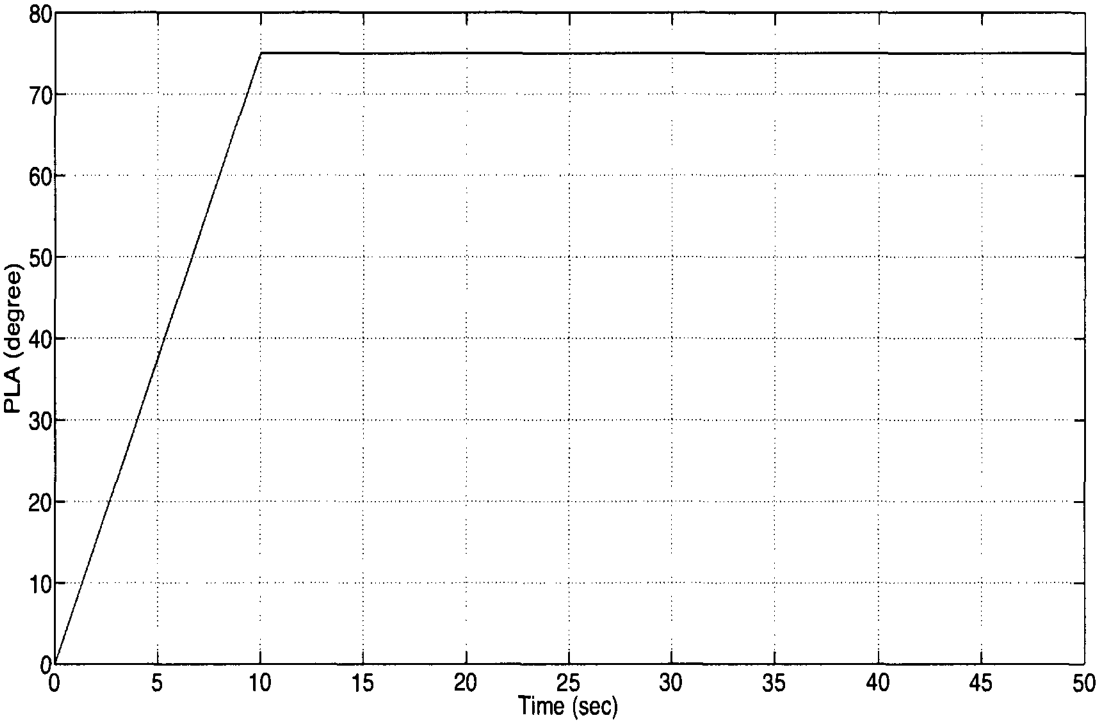


Figure 6.31: The PLA command used for diagnosis simulations.

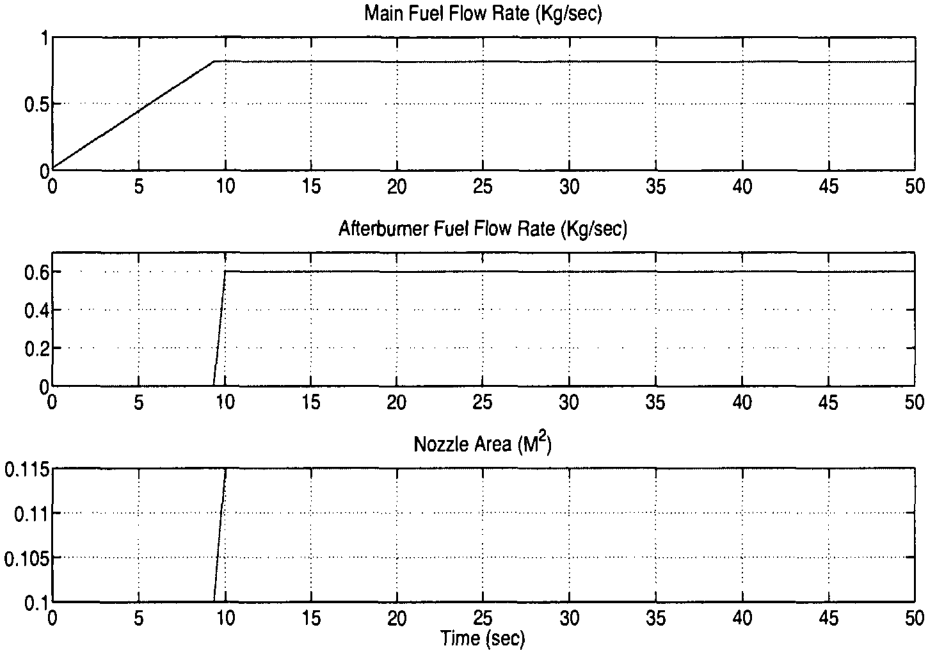


Figure 6.32: Input signals in the normal mode of operation.

occurred. However, one cannot determine which of the failure modes F^1, F^3, F^6, F^7 has occurred[1]. In other words, it cannot be inferred from the output of the isolators if the main fuel system governor has failed or the main shut-off valve is stuck-closed or the main fuel system pump failed while in operation.

We also performed simulations in the presence of measurement noise in the engine. Figure 6.37 shows the output of the isolators designed for the operating regime 8 when measurement nose is present. We have used *Monte Carlo* method for obtaining the threshold values of the isolators in this case. It can be shown that the detection time of the fault (the time that the output of the isolator $Is^8(\{f^1\})$ changes from zero to one) has changed to 30.27 *sec.* implying that it takes longer to detect the fault comparing with the case that no measurement noise was present.

At the DES level of the engine, this fault does not change any discrete output

[1] The shape of the residual signals may allow us to distinguish $\{F^1, F^3\}, F^6$ and F^7.

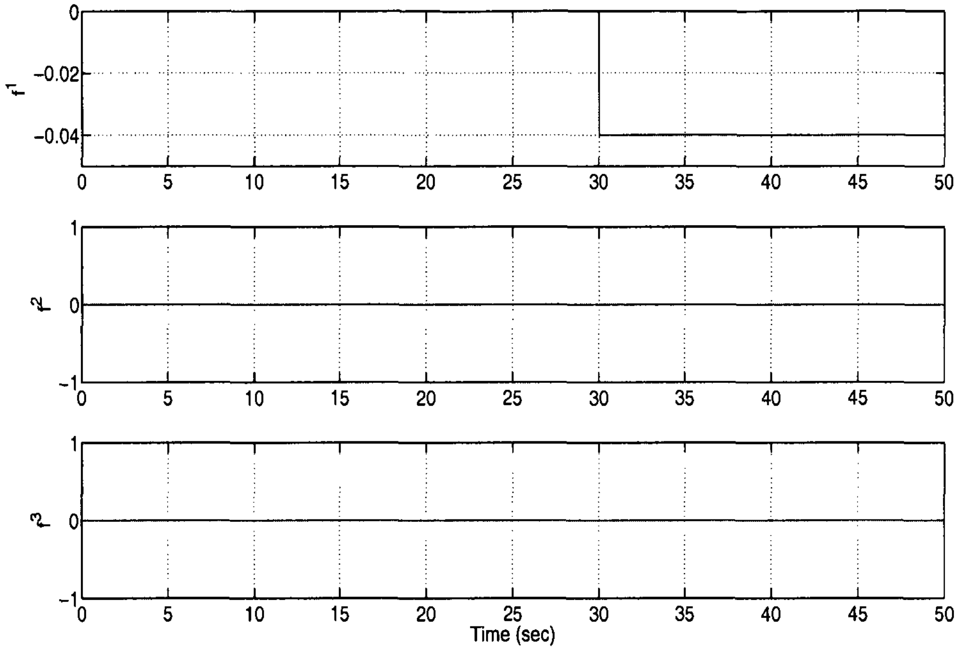


Figure 6.33: Fault type signals for the Fault Scenario 1.

in the operating section 8, and therefore cannot be detected by using only the discrete outputs. Combining the information at the DES level and the information coming from the isolators, the fault can be detected and isolated in the hybrid diagnoser. As described earlier in Section 6.3, the discrete output of the EDESA is an array with 30 elements. The first three elements come from the discrete sensors and the rest comes from the isolators. We observe that for the entire time that the engine is in the operating section 8, the first three elements of the output are '$[PS_M : high, PS_{AB} : high, PS_N : high]$' implying that the fault has not changed the output of the pressure sensor. Hence, the fault cannot be in the main shut-off valve or the main fuel system pump and it has to be in the main fuel system governor.

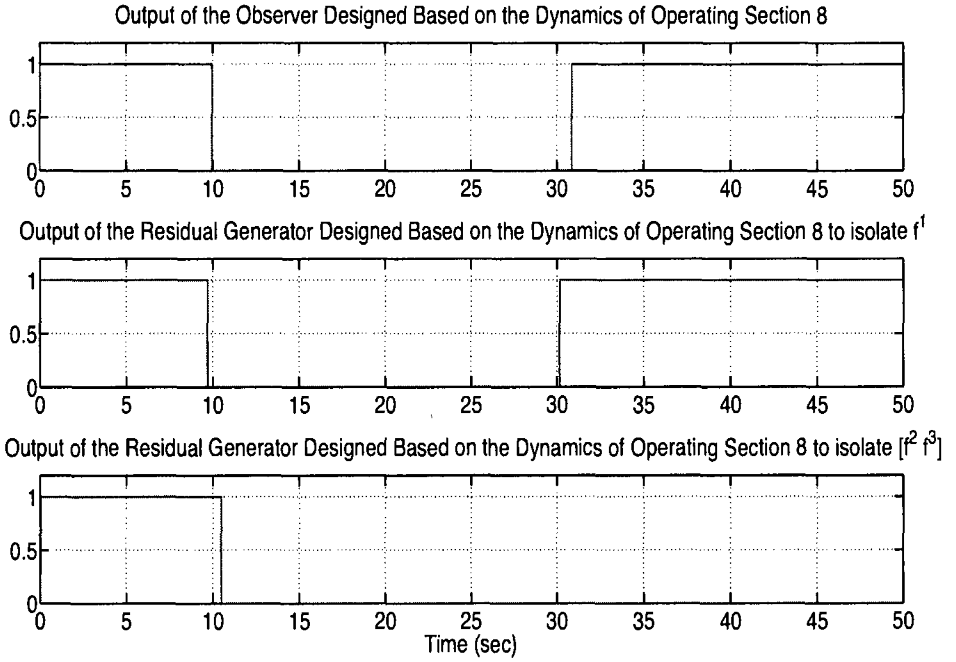


Figure 6.34: Fault Scenario 1: Outputs of the isolators designed based on the dynamics of the operating section 8 when there is no measurement noise present.

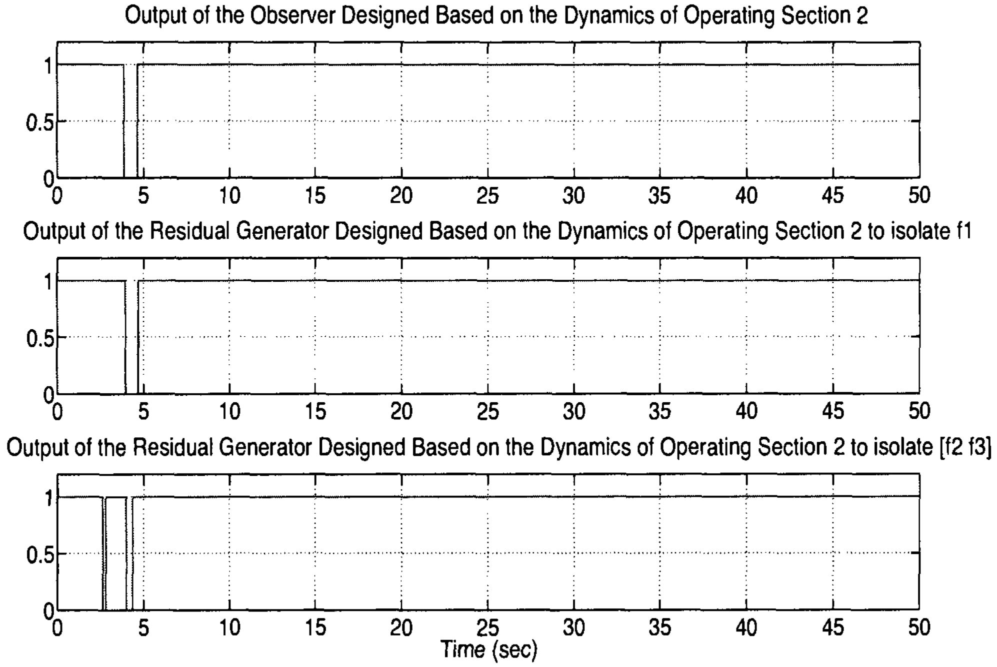


Figure 6.35: Fault Scenario 1: Outputs of the isolators designed based on the dynamics of the operating section 2 when there is no measurement noise present.

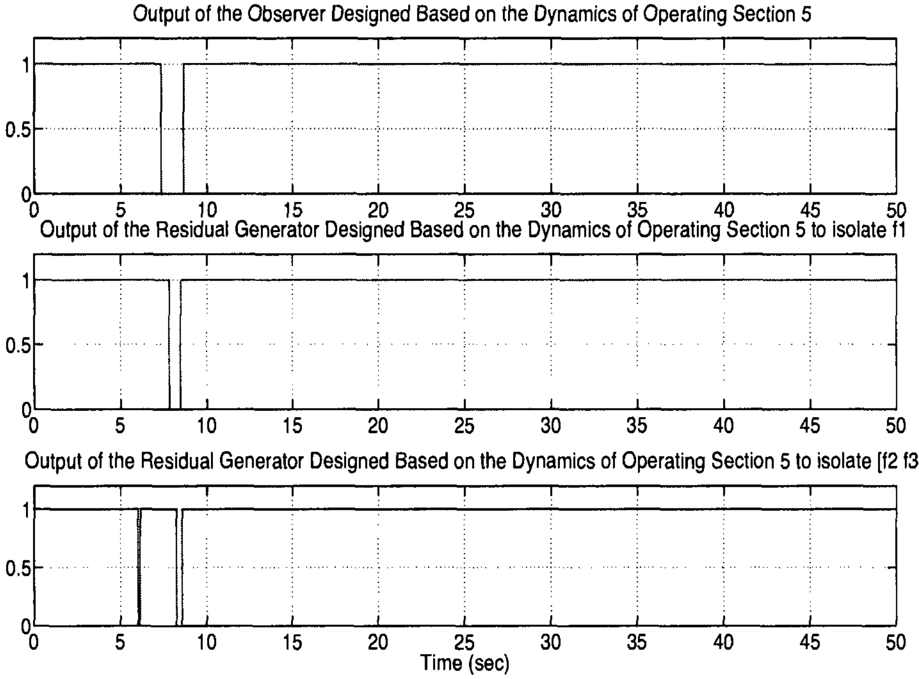


Figure 6.36: Fault Scenario 1: Outputs of the isolators designed based on the dynamics of the operating section 5 when there is no measurement noise present.

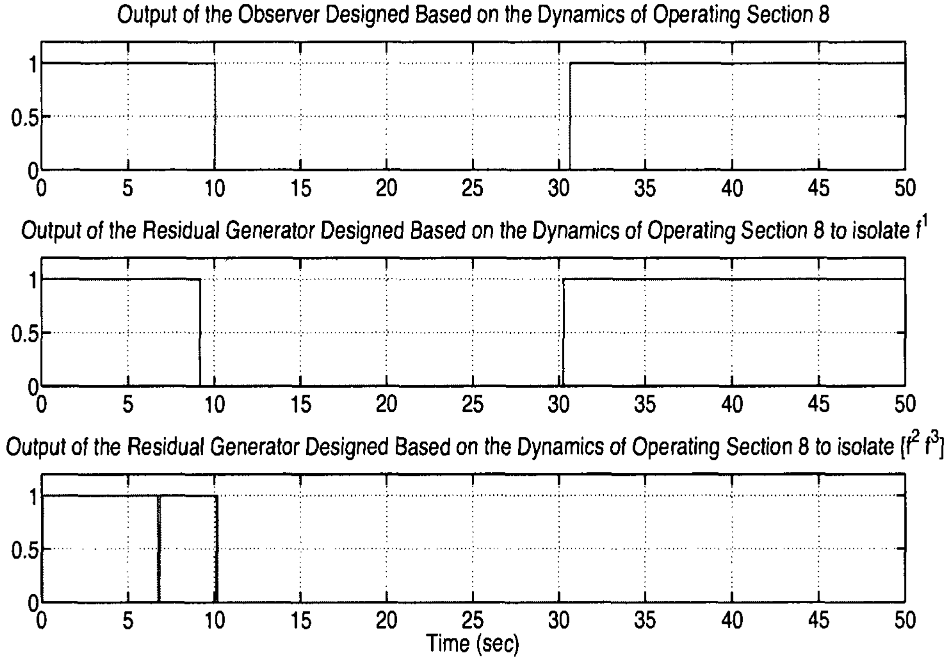


Figure 6.37: Fault Scenario 1: Outputs of the isolators designed based on the dynamics of the operating section 8 in the presence of measurement noise.

Fault scenario 2: Nozzle actuator pump is not energized

Figure 6.38 shows the fault type signals corresponding to this fault scenario. Initially, all the fault type signals are zero. When the engine enters the operating section 8 at $t \simeq 9.33$ *sec.*, the nozzle actuator pump is commanded to turn on. From this time, the value of the fault type corresponding to the nozzle area becomes $f^2 = -A_n$, modeling the failure in the nozzle actuator pump. Figure 6.39 shows the output of the isolators designed for the operating section 8 when there is no measurement noise present in the engine. We observe that from $t \simeq 11.92$ *sec.*, all isolators generate non-zero residuals, but $Is^{Q^8}(\{f^1\})$ generates a zero residual implying that the engine is in the operating section 8 and f^1 has not occurred but f^2 or f^3 has occurred. However, one cannot determine which of the afterburner fuel supply system or the nozzle actuator has failed.

We also performed simulations in the presence of measurement noise in the engine. Figure 6.40 shows the output of the isolators designed for the operating section 8 in the presence of measurement noise. We observe that when measurement noise is present, the isolator $Is^{Q^8}(\{f^1\})$ generates zero output at $t = 9.16$ *sec.*, but when there is no noise present, it generates zero output at $t = 11.92$ *sec.*

We observe that for the entire time that the engine is in the operating section 8, the first three elements of the output of the EDESA are '$[PS_M : high, PS_{AB} : high, PS_N : low]$' implying that the fault is not in the afterburner and it has to be in the nozzle actuator. The discrete output generated by the pressure sensor PS_N can only stay at "low" (while the engine is in the operating section 8) if the pump of the nozzle actuator has failed and the failure mode F^5 is present in the engine. Therefore, the fault type f^3 and the failure mode F^5 are isolated.

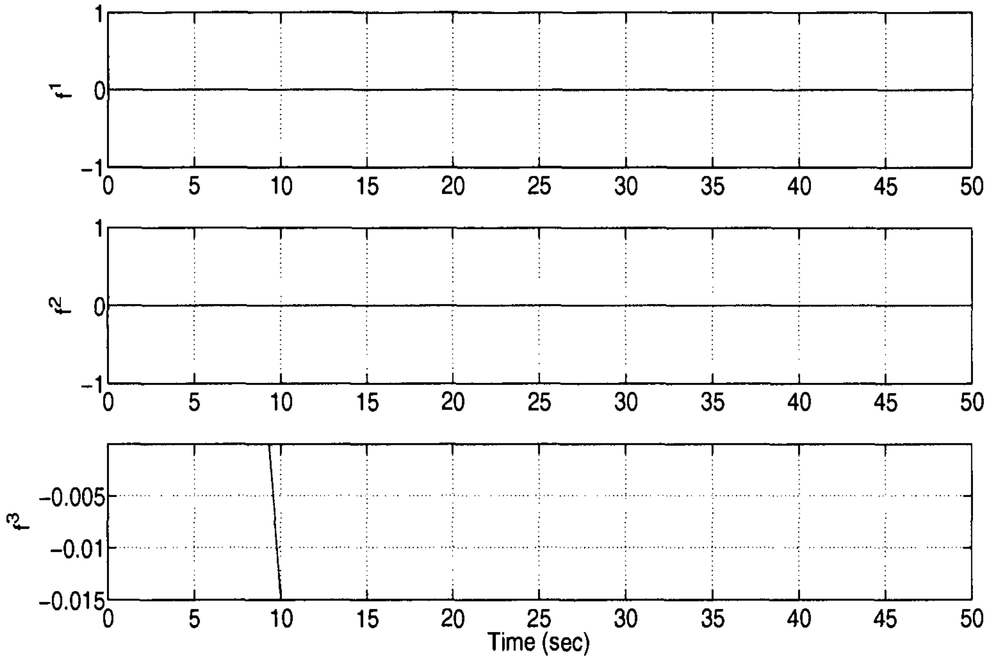


Figure 6.38: Fault type signals for the Fault Scenario 2.

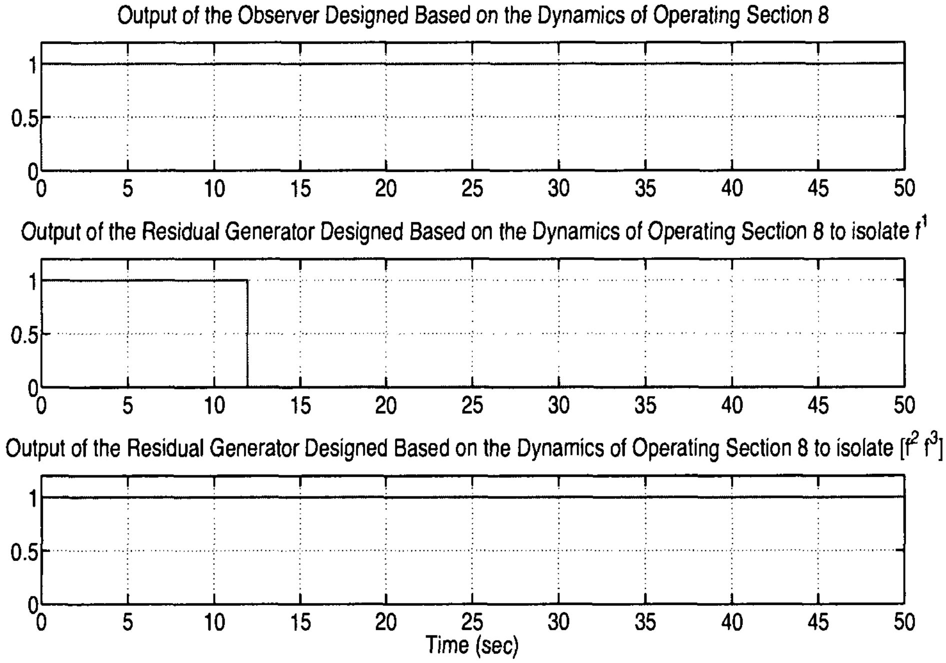


Figure 6.39: Fault Scenario 2: Output of the isolators designed based on the dynamics of the operating section 8 when there is no measurement noise present.

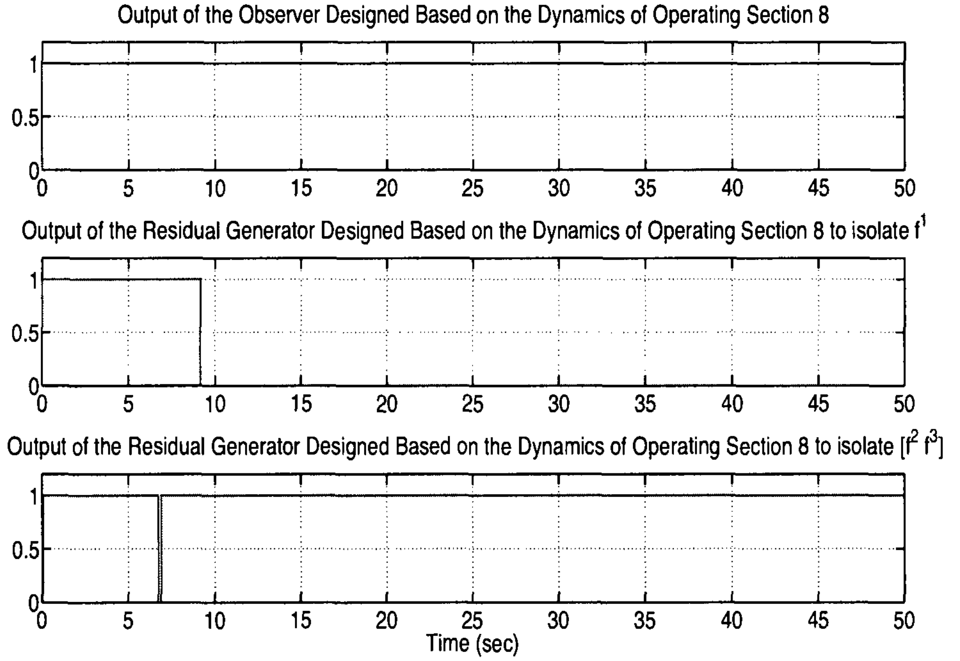


Figure 6.40: Fault Scenario 2: Output of the isolators designed based on the dynamics of the operating section 8 in the presence of measurement noise.

Fault scenario 3: Simultaneous 5% loss-of-effectiveness of main fuel system governor applied at $t = 20$ *sec.* and 5% oversupplying of the afterburner fuel system governor applied at $t = 40$

Figure 6.41 shows the fault type signals corresponding to this fault scenario. Initially, all the fault type signals are zero. At $t = 20$ *sec.*, the value of the fault type corresponding to the main fuel system changes from zero to -0.04 which is 5% of the main fuel flow rate at $t = 20$ *sec.*.

Initially assume that there is no measurement noise present in the engine. Similar to the Fault Scenario 1, we observe that from $t \simeq 9.71$ *sec.* to $t = 20.17$ *sec.*, the response of all isolators except the isolators designed for the operating section 8 (i.e, $Is^8(\{f^1\})$, $Is^8(\{f^2, f^3\})$, Obs^8) are one. This implies that the engine is in the operating section 8 during this interval and no fault has occurred. Figure 6.42 shows the response of the isolators designed for the operating section 8. From $t = 20.17$ *sec.* to $t = 40.03$ *sec.*, only the response of $Is^8(\{f^2, f^3\})$ remains at zero implying that the fault type f^1 is active and f^2 or f^3 have not occurred. However, one cannot determine which of the failure modes F^1, F^3, F^6, F^7 has occurred.

We observe that from $t = 20.17$ *sec.* to $t = 40.03$ *sec.*, the first three elements of the EDESA output are '$[PS_M : high, PS_{AB} : high, PS_N : high]$' implying that the fault has not changed the output of the pressure sensor. Hence, the fault cannot be in the main shut-off valve or the main fuel system pump, and it has to be in the main fuel system governor.

At $t = 40.03$ *sec.*, all the residuals provided by the isolators become nonzero. This can be interpreted as the simultaneous activeness of f^1 and either of f^2 or f^3, or the change of the operating regime of the engine (which consequently changes the operating section). If there was a transition to a discrete state with different dynamics (different operating section), the isolators designed for that operating section to isolate f^2 and f^3 from f^1 will generate a zero output after the effect of the mismatch between its initial condition and the continuous state of the system dies out (in our simulations, this time is less than 2 *sec.*). Since no transition from one to zero is observed from any isolators, it can be concluded that at $t = 40.03$ *sec.*, one of the fault types f^2 and f^3 becomes active. However, one cannot determine which of the afterburner fuel supply system or the nozzle actuator has failed.

We observe that from $t = 40.03$ *sec.* to $t = 50$ *sec.*, the first three elements of the output of the EDESA are '$[PS_M : high, PS_{AB} : high, PS_N : High]$' implying that the fault is not in V_{AB}, P_{AB} and P_N. Therefore, fault has to be in the afterburner fuel system governor or nozzle actuator governor. The information received from the sensors, however, is not sufficient to find the faulty component.

We also performed simulations in the presence of measurement noise in the engine. Figure 6.43 shows the output of the isolators designed for the operating section 8 when measurement nose is present. We observe that the detection time of the first fault (5% loss-of-effectiveness of main fuel system governor) has changed to 20.32 *sec.* Moreover, the detection time of the second fault (5% oversupplying of the afterburner fuel system governor) has changed to 40.10 *sec.* This implies that it takes longer to detect the faults comparing with the case that no measurement noise was present.

6.4.3 Discussion

Diagnosis in the presence of noise

It should be pointed out that occasionally false alarm signals are generate by the isolators when measurement noise is present in the engine. Table 6.14 shows the percentage of the false alarms corresponding to different operating sections. For conducting the simulations, we have performed a Monte Carlo simulation of 50 randomly generated PLA commands in each operating section. The high rate of false alarms particularly for the operating sections 1, 2 and 8 is due to the possible large

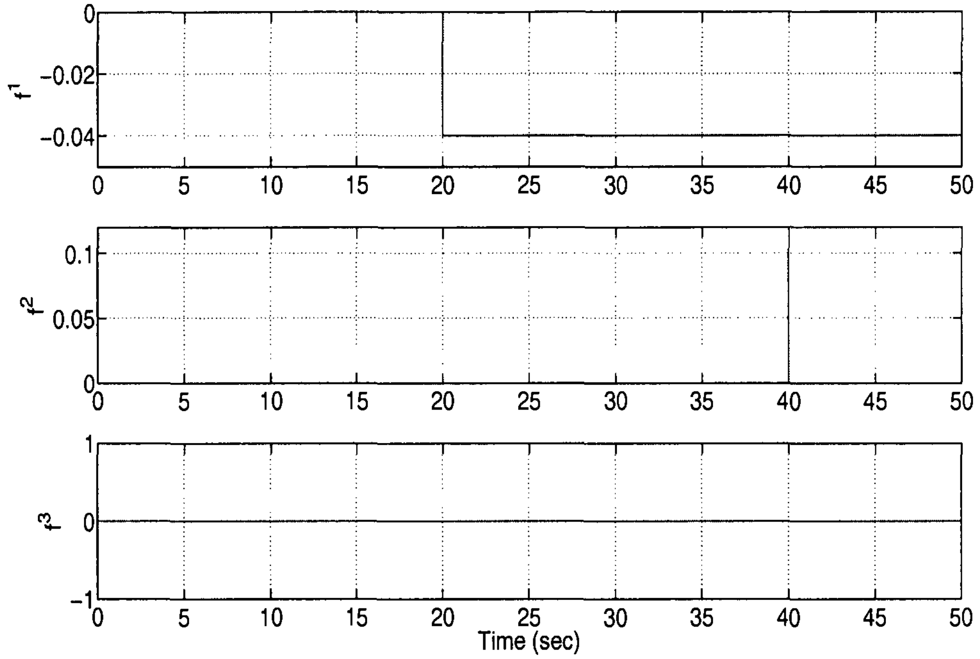


Figure 6.41: Fault type signals for the Fault Scenario 3.

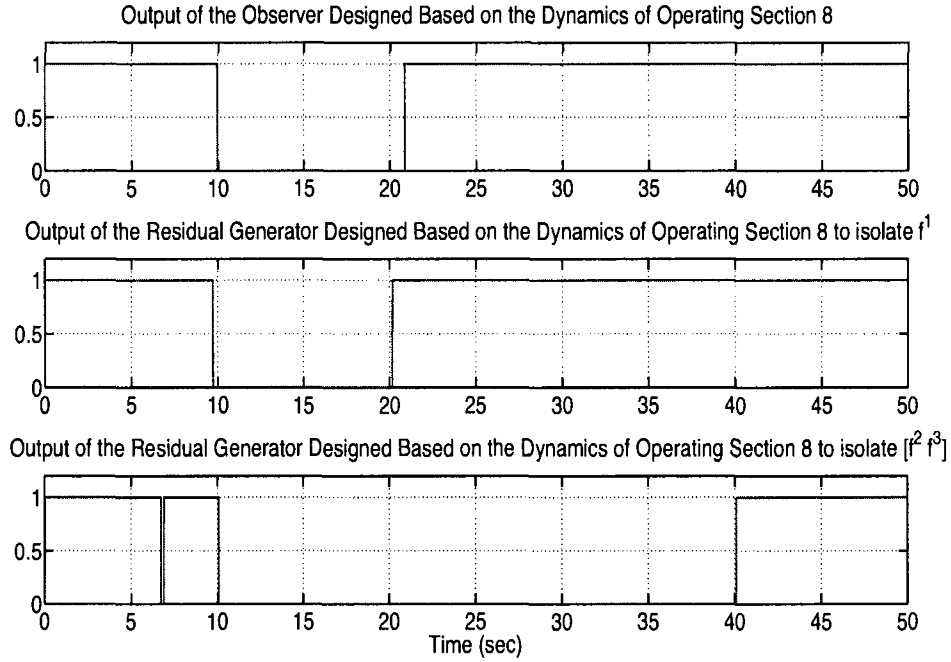


Figure 6.42: Fault Scenario 3: Outputs of the isolators designed based on the dynamics of the operating section 8 when there is no measurement noise present.

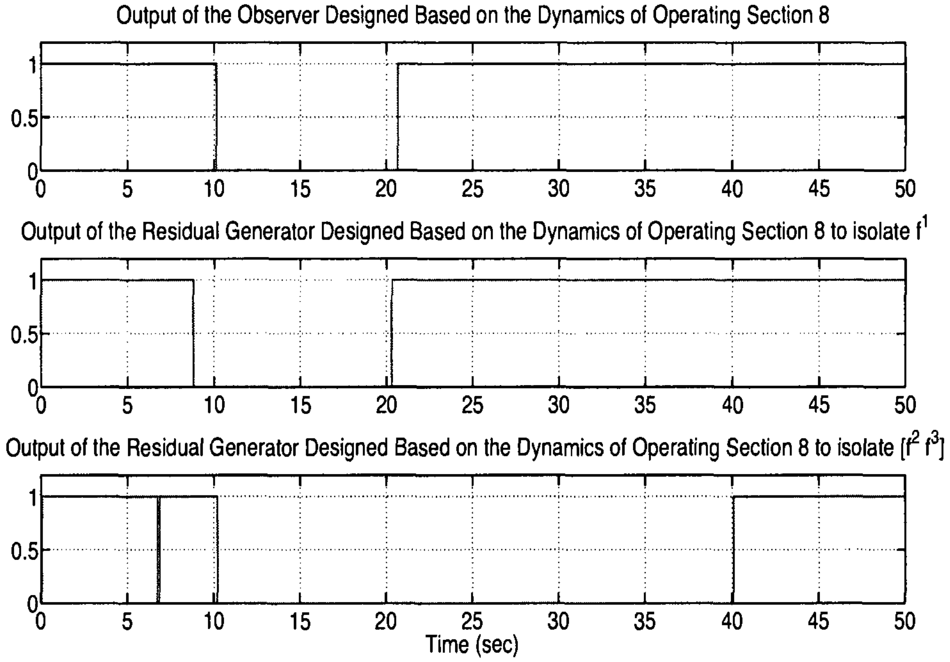


Figure 6.43: Fault Scenario 3: Outputs of the isolators designed based on the dynamics of the operating section 8 in the presence of measurement noise.

Operating section	Rate of false alarms
Section 1	26 %
Section 2	12 %
Section 3	6 %
Section 4	6 %
Section 5	4 %
Section 6	4 %
Section 7	6%
Section 8	11%
Section 9	4%

Table 6.14: Percentage of the false alarms in different operating sections.

discrepancy between the actual nonlinear model of the engine and its approximated linearized models. In this work, we have developed 9 linearized models corresponding to 9 operating sections. In order to reduce the rate of false alarms, one can increase the number of the operating sections and therefore, increase the number of the linearized models. As explained in the fault scenarios, detection time of the faults is also longer in the case that measurements are noisy because thresholds are higher.

It should be noted that residual generators in our framework can be designed according to any model-based technique. In case of noisy measurements, Kalman filter-based isolators may represent a better choice. In case of unmodeled dynamics and uncertainties in the system model, robust H_∞ techniques can be used for designing the residual generators.

Performance of the isolators

Although the isolators are designed based on linear models, we have used the output of the nonlinear model of the engine to drive the isolators. Therefore, the performance of the isolators becomes dependent on the values of the fault types. For example, we observe that the isolator $Is^{Q^i}(\{f^2, f^3\})$ passes its threshold when f^1 is large. Therefore, for large values of f^1, the output of all isolators passes the threshold and diagnosis results are not reliable. Table 6.15 shows the range of the

Operating Section	Range of the size of the fault types that can be detected and isolated
Section 2	$0.065\dot{m}_{f2}^{max} \leq f^1 \leq 0.2\dot{m}_{f2}^{max}, \quad -0.2\dot{m}_{f2}^{max} \leq f^1 \leq -0.11\dot{m}_{f2}^{max}$ $0.05\dot{m}_{fAB}^{max} \leq f^2 \leq \dot{m}_{fAB}^{max}$ $0.055A_n^{max} \leq f^3 \leq A_n^{max}$
Section 7	$0.03\dot{m}_f^{max} \leq f^1 \leq 0.35\dot{m}_f^{max}, \quad -0.35\dot{m}_f^{max} \leq f^1 \leq -0.035\dot{m}_f^{max}$ $0.06\dot{m}_{fAB}^{max} \leq f^2 \leq \dot{m}_{fAB}^{max}$ $0.065A_n^{max} \leq f^3 \leq A_n^{max}$
Section 9	$0.03\dot{m}_f^{max} \leq f^1 \leq 0.2\dot{m}_f^{max}, \quad -0.2\dot{m}_f^{max} \leq f^1 \leq -0.02\dot{m}_f^{max}$ $0.06\dot{m}_{fAB}^{max} \leq f^2 \leq \dot{m}_{fAB}^{max}, \quad -\dot{m}_{fAB}^{max} \leq f^2 \leq -0.07\dot{m}_{fAB}^{max}$ $0.055A_n^{max} \leq f^3 \leq A_n^{max}, \quad -A_n^{max} \leq f^3 \leq -0.05A_n^{max}$

Table 6.15: Performance of the isolators.

values of the fault types that can be detected and isolated (fault type isolation) with the isolators for the three operating sections 2, 7 and 9. In Table 6.15, $\dot{m}_{f2}^{max}$ is the maximum mass flow rate of the main fuel supply system when the engine is in the operating section 2 ($\dot{m}_{f2}^{max} \simeq 2\dot{m}_f^{max}/7$). The nozzle actuator and the afterburner become operational only in the operating sections 8 and 9. Therefore, we have not considered the negative values of fault types f^2 and f^3 in the operating sections 2 and 7.

Diagnosability of failure modes

The autonomous transitions in the hybrid automaton model of the engine occur instantaneously when the guard conditions (conditions on passing the boundaries of the operating sections) become true. It can be verified that all the operating sections are reachable from the initial conditions. Moreover, due to the constraints on the fuel supply, at least one of the autonomous transitions defined at each discrete state is prospective. We also assumed that the controllable events generated by the supervisor are all prospective. Therefore, $\tilde{Q}^{Eng,Inf} = \emptyset$, and a value for τ^{max} can be calculated for the hybrid model based on the PLA signal (as described in Section 4.1, τ^{max} is the maximum time that the hybrid system can stay in a discrete state without becoming stuck in that state).

As described in Fault scenario 1, failure modes of the main fuel governor G_M cannot be distinguished from the failure modes of the main shut-off valve V_S or main fuel pump P_M by using only discrete outputs generated in the engine or by using only the response of the isolators designed for the engine.

Hybrid diagnoser can be used to distinguish failure modes of G_M from failure modes of V_S or P_M. However, using the hybrid diagnoser, one cannot isolate over-supplying of G_M (the failure mode F^7) from loss-of-effectiveness of G_M (the failure mode F^6). Therefore, we group the failure modes F^6 and F^7 together and call them the failure modes of the main fuel governor. Now, we observe that $\{F^6, F^7\}$ is diagnosable using the hybrid diagnoser. Similarly, we denote $\{F^8, F^9\}$ and $\{F^{10}, F^{11}\}$ the failure modes of the afterburner fuel system governor and the failure modes of the nozzle governor, respectively.

Failure modes of the nozzle actuator governor and afterburner fuel system governor cannot be isolated from each other. More sensors are needed to isolate them in our hybrid diagnoser. Table 6.16 shows the resolution of the isolation of failure modes in our hybrid diagnoser for the case of maximum two simultaneous failure modes.

Diagnosability in the presence of isolators generating incorrect output

Assuming that a maximum of two failure modes may occur simultaneously, the failure modes F^1, F^3 and F^5 and the group of failures $\{F^2, F^4\}$ (viewed as one failure mode) can be diagnosed by using only the output of the discrete sensors (as shown in Table 6.16). Therefore, these failure modes are diagnosable even if all the isolators generate incorrect output. The output of the observer designed for each operating section is one if the output of one of the isolators designed for that operating section is one. Therefore, in the case that one of the isolators generates incorrect output, the groups of failures $\{F^6, F^7\}$ and $\{F^8, F^9, F^{10}, F^{11}\}$ remain diagnosable in the

Failure mode(s)	Detection using only isolators	Detection using only discrete outputs	Detection using the hybrid diagnoser	Isolation using only isolators	Isolation using only discrete outputs	Isolation in the hybrid diagnoser
F^1	Yes	Yes	Yes	No	Yes	Yes
F^2	Yes	Yes	Yes	No	No	No
F^3	Yes	Yes	Yes	No	Yes	Yes
F^4	Yes	Yes	Yes	No	No	No
F^5	Yes	Yes	Yes	No	Yes	Yes
F^6	Yes	No	Yes	No	No	No
F^7	Yes	No	Yes	No	No	No
F^8	Yes	No	Yes	No	No	No
F^9	Yes	No	Yes	No	No	No
F^{10}	Yes	No	Yes	No	No	No
F^{11}	Yes	No	Yes	No	No	No
$\{F^2, F^4\}$	Yes	Yes	Yes	No	Yes	Yes
$\{F^6, F^7\}$	Yes	No	Yes	**No**	**No**	**Yes**
$\{F^8, F^9, F^{10}, F^{11}\}$	Yes	No	Yes	**No**	**No**	**Yes**

Table 6.16: Diagnosability of failure modes.

hybrid diagnoser.

Isolator selection

Assuming that maximum of two failure modes may occur simultaneously, the failure modes F^1, F^3 and F^5 and the group of failure modes $\{F^2, F^4\}$ can be diagnosed by using only the output of the discrete sensors. Therefore,

$$\mathbf{IS}_{Min}^{F^1} = \mathbf{IS}_{Min}^{F^3} = \mathbf{IS}_{Min}^{F^5} = \mathbf{IS}_{Min}^{\{F^2,F^4\}} = \emptyset$$

As mentioned earlier, one of the isolators designed for each section generates redundant output. Therefore, a minimal set of isolators for diagnosability of the group of failure modes $\{F^6, F^7\}$ is

$$\begin{aligned}\mathbf{IS}_{Min}^{\{F^6,F^7\}} = \{&Is^{Q^1}(\{f^1\}), Is^{Q^1}(\{f^2,f^3\}), Is^{Q^2}(\{f^1\}), Is^{Q^2}(\{f^2,f^3\}), Is^{Q^3}(\{f^1\}),\\ &Is^{Q^3}(\{f^2,f^3\}), Is^{Q^4}(\{f^1\}), Is^{Q^4}(\{f^2,f^3\}), Is^{Q^5}(\{f^1\}), Is^{Q^5}(\{f^2,f^3\}),\\ &Is^{Q^6}(\{f^1\}), Is^{Q^6}(\{f^2,f^3\}), Is^{Q^7}(\{f^1\}), Is^{Q^7}(\{f^2,f^3\}), Is^{Q^8}(\{f^1\}),\\ &Is^{Q^8}(\{f^2,f^3\}), Is^{Q^9}(\{f^1\}), Is^{Q^9}(\{f^2,f^3\})\}\end{aligned}$$

The components of the afterburner fuel supply system and the nozzle actuator become operational only in the operating sections 8 and 9 corresponding to the operating regimes Afterburner On and Maximum Afterburner Thrust. Therefore, diagnosis and diagnosability of the failure modes of the components of the afterburner fuel supply system and the nozzle actuator is only possible in the operating sections 8 and 9. Therefore, a minimal set of isolators for diagnosability of the group of failure modes $\{F^8, F^9, F^{10}, F^{11}\}$ is

$$\mathbf{IS}_{Min}^{\{F^8,F^9,F^{10},F^{11}\}} = \{Is^{Q^8}(\{f^1\}), Is^{Q^8}(\{f^2,f^3\}), Is^{Q^9}(\{f^1\}), Is^{Q^9}(\{f^2,f^3\})\}$$

6.5 Summary

In this chapter, we investigated fault diagnosis in the actuator systems of a single-spool turbojet engine. We described the static thermodynamic relations in the engine components and reviewed dynamical equations for describing the transient behavior of the engine. We described different operating regimes of the engine and developed linear system models for the engine in each operating regime. Moreover, we developed DES models for the fuel supply systems and nozzle actuator. Combining the DES models with the linear system models developed for each operating section, we built a hybrid automaton model for the engine. Based on the hybrid automaton model and the isolators designed based on the dynamics of the operating regime, we constructed a hybrid diagnoser. We presented the simulation results and demonstrated the operation of our hybrid diagnoser for three fault scenarios. We also discussed the diagnosability of failure modes, isolator selection and isolator performance in the hybrid diagnoser developed for the engine. We showed that some failure modes such as the failure modes in the main fuel governor cannot be isolated by using only the output of the discrete sensors or by using only the response of the isolators, however, they can be isolated by using the hybrid diagnoser.

Chapter 7

CONCLUSION

7.1 Summary

Stringent reliability and maintainability requirements for modern complex systems demand development of systematic methods for fault detection and isolation. In this work, we presented a novel framework for fault diagnosis of systems that are modeled by hybrid automata. Many complex systems can be modeled as hybrid automata. The dynamics of a hybrid automaton are characterized by a Discrete-Event System (DES) representing transitions among various modes of operation and a set of continuous models (i.e., differential equations) describing the system's behaviour in the discrete modes. Generally, in a hybrid system, two types of sensors may be available, namely: continuous supplying continuous readings (i.e., real numbers) and threshold sensitive (discrete) supplying discrete outputs (e.g., level high and pressure low).

In our hybrid framework, we assumed a bank of residual generators (detection filters) based on the continuous models of the system is available. We modeled each residual generator by a DES model, and then integrated the DES models of the residual generators and the DES model of the hybrid system to build an

"Extended DES" model. A hybrid diagnoser was constructed based on the extended DES model. The hybrid diagnoser effectively integrates the readings of discrete sensors and the information supplied by the residual generators (which is based on continuous sensors) to determine the health status of the hybrid system.

We also studied the problem of diagnosability of failure modes in hybrid automata. We introduced a notion of diagnosability in hybrid automata and developed a systematic approach for verifying the diagnosability of failure modes in our hybrid diagnosis framework. We showed that for diagnosability of a failure mode in a hybrid automaton, it is sufficient that the failure mode be diagnosable in the extended DES model developed for representing the hybrid automaton and residual generators. We also investigated diagnosability of failure modes in the case that some residual generators produce unreliable outputs in the form of false alarm or false silence signals. Moreover, we investigated the problem of isolator (residual generator) selection and developed procedures for computing a minimal set of isolators to ensure the diagnosability of failure modes.

In this thesis, we employed our hybrid diagnosis approach for investigating faults in the fuel supply system and the nozzle actuator of a single-spool turbojet engine with an afterburner. Components in a fuel supply system and a nozzle actuator such as pumps and solenoid valves behave in a discrete-event manner. The status of these components varies when the operating regime of the engine changes. The discrete-event behavior of these components can be described by DES models. On the other hand, thrust generation in an engine is a continuous process, and operation of engine components such as compressor and turbine can be described by continuous static and dynamic thermodynamic relations (i.e., algebraic and differential equations). We developed linear system models to represent the continuous dynamics of the engine in different operating regimes. Our developed hybrid automaton model for the engine was obtained by integrating the DES models of the fuel supply system and the nozzle actuator with the aforementioned linear system models. A bank of residual generators was then designed based on the linear system models. Each residual generator was modeled by a DES system and an extended DES was constructed by combining the DES models of the residual generators and the DES model of the engine. Based on the extended DES model, a hybrid diagnoser was built. We showed that there are cases when the faults in the fuel supply system and the nozzle actuator cannot be isolated by a purely DES diagnoser or by methods that are based on the residual generators alone. However, the fault can be isolated if the hybrid diagnoser is used. A number of simulation studies were conducted to demonstrate and verify the advantages of our proposed hybrid fault diagnoser.

7.2 Future Research

In this thesis, we showed that a failure mode F^i is diagnosable in the hybrid system if F^i is diagnosable in the extended DES of the integrated system and isolators. In Section 4.1, we showed by an example that the F^i may be undiagnosable in the extended DES, but diagnosable in the hybrid system. Developing conditions that make diagnosability of F^i in the extended DES equivalent to the diagnosability of F^i in the hybrid system will be a subject of future research. This may require a change in the way isolators are defined and designed.

In this work, we assumed that the system is slow enough and stays in each discrete state for at least τ^{min}, and isolators are designed so that the transient response due to the mismatch in the initial conditions of the isolators and the system dies out in less than τ^{min}.) Finding the conditions under which the system can be guaranteed to stay in each discrete state for at least τ^{min} seems important. Furthermore, fault diagnosis in the case that it cannot be ensured that the transient response due to the mismatch in the initial conditions of residual generators and system dies out in less than τ^{min} is an issue for future work.

The diagnosability of failure modes in our work was studied assuming that the continuous dynamics are represented by continuous-time systems. Extending the results to the case that the continuous dynamics are represented by discrete-time systems is a subject of future work. Moreover, investigating the performance of the proposed fault diagnosis scheme in the presence of noise and uncertainties in the system's model is an interesting topic for future research.

Bibliography

[1] "Power plant controls for aero-gas turbine engines," *Advisory Group for Aerospace Research and Development (AGARD) conference proceedings*, no. 151, Ustaoset, Norway, 1974.

[2] TTCT Software Package: available at: http://www.control.utoronto.ca/people/profs/wonham/wonham.html, accessed in April 2009.

[3] http://commons.wikimedia.org/wiki/Image:J85_ge_17a_turbojet_engine.jpg, accessed in April 2009

[4] http://en.wikipedia.org/wiki/Image:Jet_engine.svg, accessed in April 2009.

[5] L. Aguirre-Salas, "Sensor selection for observability in interpreted petri nets: a genetic approach," *Proc. of the 42nd IEEE Conf. on Decision and Control*, Maui, Hawaii USA, vol. 6, pp. 3760-3765, 2003.

[6] R. Alur, C. Courcoubetis, T. A. Henzinger and P.-H. Ho, "Hybrid automata: an algorithmic approach to the specification and verification of hybrid systems," *Hybrid Systems, vol. 736 of Lecture Notes in Computer Science*, pp. 209-229, Springer, 1993.

[7] R. Alur, C. Courcoubetis, N. Halbwachs, T.A. Henzinger, P. Ho, X. Nicollin, A. Olivero, J. Sifakis and S. Yovine, "The algorithmic analysis of hybrid systems," *Theoretical Computer Science*, vol. 138, pp. 3-34, 1995.

[8] R. Alur, T. Dang and F. Ivanid, "Reachability analysis of hybrid systems via predicate abstraction," *Hybrid Systems: Computation and Control, 5th Intl. Workshop, vol. 2289 of Lecture Notes in Computer Science*, pp. 35-48, Springer-Verlag, 2002.

[9] R. Alur and D. L. Dill, "A theory of timed automata," *Theoretical Computer Science*, vol. 126, pp. 183-235, 1994.

[10] P. J. Antsaklis, X. D. Koutsoukos, "Hybrid systems: review and recent progress," *T. Samad Editor: Software Enabled Control: Information Technology for Dynamical Systems*, NY: Wiley-IEEE, pp. 273-298, 2003.

[11] P. J. Antsaklis, X. D. Koutsoukos and J. Zaytoon, "On hybrid control of complex systems: A survey," *European Journal of Automation*, vol. 32, pp. 1023-1045, 1998.

[12] E. Asarin, O. Bournez, T. Dang and O. Maler, "Approximate reachability analysis of piecewise-linear dynamical systems," *Hybrid Systems: Computation and Control, 3rd Intl. Workshop, vol. 1790 of Lecture Notes in Computer Science*, pp. 21-31, Springer-Verlag, 2000,

[13] A. Balluchi, L. Benvenuti, M. D. Di Benedetto and A. L. Sangiovanni-Vincentelli, "Design of observers for hybrid systems," *C. J. Tomlin, M. R. Greenstreet, Editors: Hybrid Systems: Computation and Control, vol.2289 of Lecture Notes in Computer Science*, pp. 76-89, Springer-Verlag, 2002.

[14] A. Balluchi, L. Benvenuti, M.D. Di Benedetto and A.L. Sangiovanni-Vincentelli, "Observability for hybrid systems," *Proc. of the 42nd IEEE Conf. on Decision and Control*, Maui, Hawaii, USA, 2003.

[15] P. Baroni, G. Lamperti, P. Pogliano and M. Zanella, "Diagnosis of large active systems," *Artificial Intelligence*, vol.110, no.1, pp. 135-183, 1999.

[16] M. Basseville, A. Benveniste and L. Tromp, "Diagnosing hybrid dynamical systems: Fault graphs, statistical residuals and viterbi algorithms," *Proc. of the 37th IEEE Conf. on Decision and Control*, Tampa, Florida, USA, pp. 3757-3762, 2000.

[17] S. Bavishi and E. K. Chong, "Automated fault diagnosis using a discrete event systems framework," *Proc. of the 9th IEEE Intl. Symposium on Intelligent Control,* Columbus, Ohio, USA, vol.16, no.18, pp. 213-218, 1994.

[18] R. V. Beard, "Failure accommodation in linear systems through self-reorganization," Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 1971.

[19] E. Benazera, L. Trave-Massuyes and P. Dague, "State tracking of uncertain hybrid concurrent systems," Proc. of the 13th Intl. Workshop on Principles of Diagnosis, Semmering, Austria, pp. 106-114, 2002.

[20] H. A. P. Blom and Y. Bar-Shalom, "The interacting multiple model algorithm for systems with markovian switching coefficients," *IEEE Trans. on Automatic Control*, vol. 33, no. 8, pp. 780-783, 1988.

[21] B. A. Brandin and W. M. Wonham, "Supervisory control of timed discrete-event systems," *IEEE Trans. on Automatic Control*, vol. 39, pp. 329-341, 1994.

[22] M. S. Branicky, "Studies in hybrid systems: Modeling, analysis, and control," Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 1995.

[23] M. Branicky, V. Borkar and S. Mitter, "Unified framework for hybrid control: Model and optimal control theory," *IEEE Trans. on Automatic Control*, vol. 43, no. 1, pp. 31-45, 1998.

[24] J. V. Casamassa and R. D. Bent, *Jet aircraft power systems*, McGraw Hill, 2nd edition, 1967, ISBN-13: 978-0070101999.

[25] J. Chen and R. Patton, *Robust Model-Based Fault Diagnosis for Dynamic Systems*, Kluwer Academic Publishers, 1999.

[26] Y. Chen and G. Provan, "Modeling and diagnosis of timed discrete event systems- A factory automation example," *Proc. of American Automatic Control Conf.*, New Mexico, USA, vol. 1, pp. 31-36, 1997.

[27] E. Y. Chow and A. S. Willsky "Analytical redundancy and the design of robust failure detection systems," *IEEE Trans. on Automatic Control*, vol. AC-29, pp. 603-614, 1984.

[28] P. Collins and J. H. van Schuppen, "Observability of piecewise-affine hybrid systems," *Hybrid Systems: Computation and Control, vol. 2993 of Lecture Notes in Computer Science*, pp. 265-279, Springer-Verlag, Philadelphia, PA, USA, 2004.

[29] B. Crossette, "Jet pilot who saved 304 finds heroism tainted," *The New York Times*, Sept. 10, 2001.

[30] X. Dai, T. Breikin, Z. Gao and W. Hong, "Dynamic modelling and robust fault detection of a gas turbine engine," *Proc. of the American Control Conf.*, Seattle, WA, USA, pp. 2160-2165, 2008.

[31] A. Darwiche, "Model-based diagnosis using causal networks," *Proc. of the Joint Conf. on Artificial Intelligence*, Montreal, Canada, pp. 211-217, 1995.

[32] R. Debouk, S. Lafortune and D. Teneketzis, "On an optimization problem in sensor selection," *Discrete event dynamic systems: Theory and Applications*, vol. 12, no. 4, pp.417-445, 2002.

[33] C. De Persis and A. Isidori, "A geometric approach to nonlinear fault detection and isolation," *IEEE Trans. on Automatic Control*, vol. 46, no. 6, pp. 853-865, 2001.

[34] M. Desai and A. Ray, "A fault detection and isolation methodology," *Proc. of the 20th IEEE Conf. on Decision and Control*, San Diego, CA, USA, pp. 1363-1369, 1981.

[35] M. D. Di Benedetto, S. Di Gennaro and A. D'Innocenzo, "Diagnosability verification for hybrid automata and durational graphs," *Proc. of the 46th IEEE Conf. on Decision and Control*, New Orleans, LA, USA, pp. 1789-1794, 2007.

[36] Y. E. Fattah and G. Provan, "Modeling temporal behavior in the model-based diagnosis of discrete event systems," *Proc. of the Intl. Workshop in Principles of Diagnosis*, Mont-Saint-Michel, France, pp. 43-50, 1997.

[37] G. Ferrari-Trecate, D. Mignone and M. Morari, "Discrete abstractions of hybrid systems," *IEEE Trans. on Automatic Control*, vol. 47, no. 10, pp. 1663-1676, 2001.

[38] G.K. Fourlas, K.J. Kyriakopoulos and N.J. Krikelis, "Diagnosability of hybrid systems," *Proc. of the 10th IEEE Mediterranean Conf. on Control and Automation*, Lisbon, Portugal, 10 pages, 2002.

[39] G.K. Fourlas, K.J. Kyriakopoulos and N.J. Krikelis, "Model-based fault diagnosis of hybrid systems based on hybrid structure hypothesis testing," *Proc.*

of the 11th IEEE Mediterranean Conf. on Control and Automation, Rhodes, Greece, 6 pages, 2003.

[40] G.K. Fourlas, K.J. Kyriakopoulos and N.J. Krikelis, "Power system fault diagnosis based on hybrid system modeling," *Proc. of the 12th IEEE Mediterranean Conf. on Control and Automation*, Kusadasi, Turkey, 6 pages, 2004.

[41] P. M. Frank, "Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy - A survey and some new results," *Automatica*, vol. 26, no. 3, pp. 459-474, 1990.

[42] P. M. Frank, "Analytical and qualitative model based fault diagnosis - A survey and some new results," *European Journal of Control*, vol. 2, pp. 6-28, 1996.

[43] P. M. Frank and X. Ding, "Survey of robust residual generation and evaluation methods in observer-based fault detection systems," *Journal of Process Control*, vol. 7, no. 6, pp. 403-424, 1997.

[44] J. Gertler, "Analytical redundancy methods in fault detection and isolation - A survey and synthesis," *IFAC/IMACS SAFEPROCESS*, pp. 9-21, 1991.

[45] C. Goodrich and J. Kurien, "Continuous measurements and quantitative constraints - Challenge problems for discrete modeling techniques,"*in Proc. of iSAIRAS-2001*, 9 pages, 2001.

[46] C. Hajiyev and F.Kaliskan, "Sensor/actuator fault diagnosis based on statistical analysis of innovation sequence and robust Kalman filtering," *Aerospace Science & Technology*, vol. 4, pp. 415-422, 2000.

[47] W. Hamscher, L. Console and J. de Kleer, "Readings in model-Based diagnosis," San Mateo, CA, Morgan Kaufmann, 1992.

[48] S. Hashtrudi Zad, R.H. Kwong and W.M. Wonham, "Fault diagnosis in discrete-event systems: Framework and model reduction," *IEEE Trans. on Automatic Control*, vol. 48, no. 7, pp. 1199-1212, 2003.

[49] S. Hashtrudi Zad, R.H. Kwong and W.M. Wonham, "Fault diagnosis in discrete-event systems: Incorporating timing information," *IEEE Trans. on Automatic Control*, vol. 50, no. 7, pp. 1010-1015, 2005.

[50] S. Hashtrudi Zad, R.H. Kwong and W.M. Wonham, "Fault diagnosis and consistency in hybrid systems," *Proc. 38th Annual Allerton Conf. on Communication, Control, and Computing*, University of Illinois at Urbana-Champaign, pp. 1135-1144, 2000.

[51] S. Hedlund and A. Rantzer, "Optimal control of hybrid system," *Proc. of the 38th IEEE Conf. on Decision and Control*, pp. 3972-3977, Phoenix, AZ, 1999.

[52] T. A. Henzinger, P. H. Ho and H. Wong-Toi, "A user guide to HyTech," *Proc. of the 1st Workshop on Tools and Algorithms for the Construction and Analysis of Systems, vol. 1019 of Lecture Notes in Computer Science*, pp. 41-71, Springer Verlag, 1995.

[53] M. W. Hofbaur and B.C. Williams, "Hybrid estimation of complex systems," *IEEE Trans. on Systems, Man and Cybernetics, Part B*, vol. 34, no. 5, pp. 2178- 2191, 2004.

[54] M. W. Hofbaur and B.C. Williams, "Mode estimation of probabilistic hybrid systems," *C. J. Tomlin and M.R. Greenstreet, Editors: Hybrid Systems: Computation and Control, vol. 2289 of Lecture Notes in Computer Science*, New York, Springer-Verlag, pp. 253-266, 2002.

[55] J. E. Hopcroft, R. Motwani and J. D. Ullman, *Introduction to automata theory, languages, and computation*, Addison-Wesley, 3rd edition, 2006, ISBN-13: 9780321462251.

[56] E. Isermann, "Process fault detection based on modeling and estimation methods - A survey," *Automatica*, vol. 20, no. 4, pp. 387-404, 1984.

[57] R. Isermann, "Supervision, fault-detection and fault-diagnosis methods - An introduction, *Control Engineering Practice*, vol. 5, no. 5, pp. 639-652, 1997.

[58] L. C. Jaw, "Recent advancements in aircraft engine health management (EHM) technologies and recommendations for the next step," *Proc. of Turbo Expo 2005: 50th ASME Intl. Gas Turbine & Aeroengine Technical Congress*, 13 pages, Reno-Tahoe, Nevada, USA, 2005.

[59] S. Jiang, Z. Huang and R. Kumar, "A polynormial algorithm for testing diagnosability of discrete event systems," *IEEE Trans. on Automatic Control*, vol. 46, no.8, pp. 1318-1321, 2001.

[60] K. H. Johansson, J. Lygeros, S.N. Simic, J. Zhang and S. Sastry, "Dynamical properties of hybrid automata," *IEEE Trans. on Automatic Control*, vol. 48, no. 1, pp. 2-17, 2003.

[61] H. L. Jones, "Failure detection in linear systems," Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 1973.

[62] T. Kobayashi, "Aircraft engine sensor/actuator/component fault diagnosis using a bank of kalman filters," *NASA Technical report*, NASA/CR-2003-212298, 2003, available at http://gltrs.grc.nasa.gov/reports/2003/CR-2003-212298.pdf, accessed in April 2009.

[63] T. Kobayashi and D. L. Simon, "Application of a bank of Kalman filters for aircraft engine fault diagnostics," *Proc. of ASME Turbo Expo, GT2003-38550*, Atlanta, USA, 10 pages, 2003.

[64] T. Koo, F. Hoffmann, H. Shim, B. Sinopoli and S. Sastry, "Hybrid control of an autonomous helicopter," *IFAC Workshop on Motion Control*, Grenoble, France, pp. 285-290, 1998.

[65] X. Koutsoukos, "Estimation of hybrid systems using discrete sensors," *Proc. of the 42th IEEE Conf. on Decision and Control*, Maui, Hawaii, USA, pp. 155-160, 2003.

[66] X. Koutsoukos, P.J. Antsaklis, J.A. Stiver and M.D. Lemmon, "Supervisory control of hybrid systems," *Proc. of the IEEE*, vol. 88, no. 7, pp. 1026-1049, 2000.

[67] X. D. Koutsoukos, P. J. Antsaklis, "Safety and reachability of piecewise linear hybrid dynamical systems based on discrete abstractions," *Discrete Event Dynamic Systems*, vol. 13, no.3, pp. 203-243, 2003.

[68] X. Koutsoukos, J. Kurien and F. Zhao, "Estimation of hybrid systems using particle filtering methods," *Proc. of MTNS 2002*, Notre Dame, IN, USA, 2002.

[69] X. Koutsoukos, J. Kurien and F. Zhao, "Estimation of distributed hybrid systems using particle filtering methods," *Hybrid Systems: Computation and Control, vol. 2623 of Lecture Notes in Computer Science*, pp. 298-313, Springer-Verlag, 2003.

[70] G. G. Kulikov and H. A. Thompson, *Dynamic modeling of gas turbines: Identification, simulation, condition monitoring and optimal control*, Springer, 1st edition, 2004.

[71] D. Lee and M. Yannakakis, "Principles and methods of testing finite state machines - A Survey, *Proc. of the IEEE*, vol. 84, no. 8, pp. 1090-1123, 1996.

[72] D. Liberzon, *Switching in systems and control*, Boston, MA: Birkhuser, 2003, ISBN: 978-0-8176-4297-6.

[73] F. Lin, "Diagnosability of discrete-event systems and its application," *Discrete Event Dynamic systems*, vol. 4, pp. 197-212, 1994.

[74] J. S. Litt, D. L. Simon, S. Garg, T.-H. Guo, C. Mercer, R. Millar, A. Behbahani, A. Bajwa and D. T. Jensen, "A survey of intelligent control and health management technologies for aircraft propulsion systems," *NASA Technical report*, NASA/TM-2005-213622, 2005.

[75] J. Lunze, "Diagnosis of quantised systems by means of timed discrete-event representations," *N. Lynch and B. Krogh, Editors: Hybrid Systems: Computation and Control, vol. 1790 of Lecture Notes in Computer Science*, pp. 258-271, Springer-Verlag, 2000.

[76] J. Lygeros, C. J. Tomlin and S. Sastry, "Controllers for reachability specifications for hybrid systems," *Automatica*, vol. 35, pp. 349-370, 1999.

[77] N.A. Lynch, R. Segala, F. Vaandrager and H. B. Weinberg, "Hybrid I/O automata," *R. Alur, T. A. Henzinger and E. D. Sontag, Editors: Hybrid Systems III*, Springer-Verlag, pp. 496-510, 1996.

[78] E.J. Manders, S. Narasimhan, G. Biswas and P.J. Mosterman. "A combined qualitative/quantitative approach for efficient fault isolation in complex dynamic systems," *Proc. of the 4th symposium on Fault Detection, Supervision, and Safety Processes*, Budapest, Hungary, pp. 512-517, 2000.

[79] S. McIlraith, G. Biswas, D. Clancy and V. Gupta, "Hybrid systems diagnosis," *Hybrid Systems: Computation and Control, vol. 1790 of Lecture Notes in Computer Science*, pp. 282-295, Springer-Verlag, 2000.

[80] M. A. Massoumnia, "A geometric approach to failure detection and identification in linear systems," Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 1986.

[81] M. A. Massoumnia, G.C. Verghese and A.S. Willsky, "Failure detection and identification," *IEEE Trans. on Automatic Control*, vol. AC-34, no. 3, pp. 316-321, 1989.

[82] T. E. Menke and P. S. Maybeck, "Sensor/actuator failure detection in Vista F-16 by multiple model adaptive estimation," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 31, no. 4, pp. 1218-1229, 1995.

[83] W. C. Merrill, J. C. DeLaat and W. M. Bruton, "Advanced detection, isolation and accommodation of sensor failures - realtime evaluation," *Journal of Guidance, Control and Dynamics*, vol. 11, no. 6, pp. 517-526, 1988.

[84] A. Misra, G. Provan, G. Karsai, G. Bloor and E. Scarl, "A generic and symbolic model-based diagnostic reasoner with highly scalable properties," *Proc. of the IEEE Conf. on Systems, Man, and Cybernetics*, vol. 4, pp. 3154 - 3160, San Diego, CA, USA, 1998

[85] P. Mosterman and G. Biswas, "Diagnosis of continuous valued systems in transient operating regions," *IEEE Trans on Systems, Man, and Cybernetics*, vol. 29, pp. 554-565, 1999.

[86] S. Narasimhan and G. Biswas, "Model-based diagnosis of hybrid systems," *IEEE Tran. on System, Man, and Cybernetics–Part A: Systems and Humans*, vol. 37, no. 3, pp. 348-361, 2007.

[87] S. Narasimhan, G. Biswas, G. Karsai, T. Pasternak and F. Zhao. "Building observers to handle fault isolation and control problems in hybrid systems," *Proc. of the IEEE Conf. on Systems, Man, and Cybernetics*, Nashville, TN, USA, pp. 2393-2398, 2000.

[88] F. W. Newburgh, M. A. Bennett, A. F. McLean and R. L. Paquette, *Gas turbine fuel controls analysis and design*, Society of Automotive Engineers (SAE), 1965.

[89] G. C. Oats, *Aerothermodynamics of gas turbine rocket propulsion*, American Institute of Aeronautics & Ast (AIAA), 3rd edition, 1997, ISBN-13: 978-156347241.

[90] J. Pan and S. Hashtrudi-Zad, "Diagnosability analysis and sensor selection in discrete event systems with permanent failures," *Proc. IEEE Conf. on Automation Science and Engineering*, Scottsdale, Arizona, USA, pp. 869-874, 2007.

[91] D. N. Pandalai and L. E. Holloway, "Template languages for fault monitoring of timed discrete event processes," *IEEE Trans. on Automatic Control*, vol. 45, no. 5, pp. 868-882, 2000.

[92] R. J. Patton and J. Chen, "Detection of faulty sensors in aero jet engine systems using robust model-based methods," *IEE Colloquium on Condition Monitoring for Fault Diagnosis*, pp. 2/1-2/22, 1991.

[93] R. J. Patton and J. Chen, "Observer-based fault detection and isolation: robustness and applications," *Control Engineering Practice*, vol.5, no. 5, pp. 671-682, 1997.

[94] R. J. Patton, P. M. Frank and R. N. Clark, *Issues of fault diagnosis for dynamic systems*, Springer, 1st edition, 2000.

[95] I. E. Potter and M. C. Sunman, “Thresholdless redundancy management with arrays of skewed instruments,” *Integrity in Electronic Flight Control Systems*, AGARDOGRAPH-224, pp. 15-11 to 15-25, 1977.

[96] P.J. Ramadge and W.M. Wonham, “Supervision of discrete-event processes, *Proc. of the 21th IEEE Conf. on Decision and Control*, New York, NY, USA, pp. 1228-1229, 1982.

[97] P.J. Ramadge and W.M. Wonham, “The Control of Discrete Event Systems, *Proc. of the IEEE*, vol. 77, no. 1, pp. 81-98, 1989.

[98] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen and D. Teneketzis, “Diagnosability of discrete-event systems,” *IEEE Trans. Automatic Control*, vol. 40, no. 9, pp. 1555-1575, 1995.

[99] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen and D. Teneketzis, “Failure diagnosis using discrete-event models,” *IEEE Trans. on Control System Technology*, vol. 4, no. 2, pp. 105-124, 1996.

[100] M. Sampath, S. Lafortune and D. Teneketzis, “Active diagnosis of discrete-event systems,” *IEEE Trans. on Automatic Control*, vol.43, no.7, pp. 908-929, 1998.

[101] S. Simani, C. Fantuzzi and R. J. Patton, *Model-based fault diagnosis in dynamic systems using identification techniques*, Springer, 1st edition, 2003, ISBN-13: 978-1852336851.

[102] H. I. H. Saravanamuttoo, G. F. C. Rogers and H. Cohen, *Gas turbine theroy*, Prentice Hall, 5th edition, 2001, ISBN-13: 978-0130158475.

[103] A. J. Sobey and A. M. Suggs, *Control of aircraft and missile powerplants*, John Wiley & Sons, 1963, ISBN-13: 978-0471810308

[104] I. E. Treager, *Aircraft gas turbine engine technology*, Career Education, 3rd edition, 1995, ISBN-13: 978-0028018287.

[105] S. Tripakis, "Fault diagnosis for timed automata," *Proc. of the Intl. Symposium on Formal Techniques in Real-Time and Fault-Tolerant Systems, vol. 2469 of Lecture Notes in Computer Science*, pp. 205-224, Springer-Verlag, 2002.

[106] I. Y. Tumer and A. Bajwa, "A survey of aircraft engine health monitoring systems," *35th AIAA/ASME/SAE/ASEE Joint Propulsion Conf. and Exhibit*, Los Angeles, CA, USA, AIAA-99-2528, 8 pages, 1999.

[107] R. Vidal, A. Chiuso, S. Soatto and S. Sastry, "Observability of linear hybrid systems," *Hybrid Systems: Computation and Control, vol. 2623 of Lecture Notes in Computer Science*, pp. 526-539, Springer Verlag, 2003.

[108] B. C. Williams, M. Ingham, S.H. Chung and P.H. Elliott, "Model-based programming of intelligent embedded systems and robotic space explorers," *Proc. of the IEEE*, vol. 9, no. 1, pp. 212-237, 2003.

[109] B. C. Williams and P. Nayak, "A model-based approach to reactive self-configuring systems," *Proc. of the AAAI-96*, pp. 971-978, 1996.

[110] A. S. Willsky, "A survey of design methods for failure detection in dynamic systems," *Automatica*, vol. 12, no. 6, pp. 601-611, 1976.

[111] H. S. Witsenhausen, "A class of hybrid-state continuous time dynamic systems," *IEEE Trans. on Automatic Control*, vol. 11, no. 2, pp. 161-167, 1966.

[112] W. M. Wonham, "Supervisory Control of Discrete Event Systems," Systems Control Group, Dept. of Electrical and Computer Eng., University of Toronto, Canada, 2009; available at http://www.control.utoronto.ca/DES, accessed April 2009.

[113] W.M. Wonham, *Linear multivariable control: A geometric approach*, Springer-Verlag, 3rd edition, New York, 1985.

[114] T. S. Yoo and S. Lafortune, "Polynomial-time verification of diagnosability of partially observed discrete-event systems," *IEEE Trans. on Automatic Control*, vol. 47, no. 9, pp. 1491-1495, 2002.

[115] F. Zhao, X. Koutsoukos, H. Haussecker, J. Reich and P. Cheung, "Monitoring and fault diagnosis of hybrid systems," *IEEE Trans. on Systems, Man and Cybernetics - Part B*, vol. 35, no. 6, pp. 1225-1240, 2005.