Intelligent System for Fault Diagnosis in Automotive Applications

Dissertation
der Fakultät für Informations- und Kognitionswissenschaften
der Eberhard-Karls-Universität Tübingen
zur Erlangung des Grades eines
Doktors der Naturwissenschaften
(Dr. rer. Nat.)

vorgelegt von
M.Sc./Dipl.–Ing. Mashud Kabir
aus Bangladesh

Tübingen
2008

Intelligent System for Fault Diagnosis in Automotive Applications

Tag der mündlichen Qualifikation:	19.12.2008
Dekan:	Prof. Dr.-Ing. Oliver Kohlbacher
1. Berichterstatter:	Prof. Dr. Herbert Klaeren
2. Berichterstatter:	Prof. Dr. Wolfgang Rosenstiel

Acknowledgements

I would like to thank my adviser Prof Dr Herbert Klaeren for his guidance, help and regular feedback for this work. It would have not been possible to finish my work without his professional mentorship. I would also like to thank Prof Dr Wolfgang Rosenstiel to review my dissertation work. It is my pleasure to thank Dr. Uwe Roettgermann of BMW for his help to build up the defect database and his feedback on my work. I would also like to thank my mother Mamtaj Begum and father Wali Ullah for their confidence and encouragement in achieving my goal. I thank all of my family members who were always supportive during my work.

Abstract

This work presents an intelligent system for fault diagnosis in automotive applications. The system is proposed to handle faults in a running car as well as in a car which is in the process of being developed. The main goal of this work is to use the vast knowledge previously acquired by system experts, to visualize, localize and fix a fault in any of the life phases of a car. The existing techniques and systems which are in use for fault diagnosis are investigated. There is no system which can be used in all of the phases of a car life. The newly proposed system, which will be discussed in this study, overcomes this limitation. Infotainment system in automotive areas is a new area of development. This system is adapted to the new infotainment system requirements. Optimized techniques in different stages in fault handling are proposed. Probable fixing solutions are provided for a fault by making use of the latest technologies. The quality of the results is assessed and the importance and appropriateness of the system especially for automotive infotainment system is demonstrated.

Table of Contents

Chapter 1

Introduction

Fault diagnosis has become a very important area of research during the last decade due to the advancement of mechanical and electrical systems in industries. The automobile is a crucial field where fault diagnosis is given a special attention. Due to the increasing complexity and newly added features in vehicles, a comprehensive study has to be performed in order to achieve an appropriate diagnosis model.

A diagnosis system is capable of identifying the faults of a system by investigating the observable effects (or symptoms). The system is referred to as *intelligent* if it can provide similar diagnosis results as a human *expert* could for the same fault. The Intelligent System categorizes the fault into a diagnosis class and identifies a probable cause based on the supplied fault symptoms.

The development of diagnosis classes is done by making use of previous experience, knowledge or information within an application area. The necessary information used may come from several sources of knowledge, such as from system analysis.

In previous work many approaches have been proposed to develop a diagnosis system. For example, Artificial Intelligence (AI) can contribute to the design of a fault detection and diagnosis system. An appropriate approach to build up an Intelligent System is chosen based on the nature of application knowledge.

The aim of this work is to develop an Intelligent System to diagnose faults. The complete Infotainment & Multimedia System of a vehicle is chosen as an application of the Intelligent Diagnosis System.

An appropriate scheme to describe faults in Infotainment & Multimedia must be formulated in order to realize them. A thorough investigation of Infotainment & Multimedia system should be performed to obtain the knowledge or information about the system.

A fault classification model should be built based upon the system knowledge and experience. When the diagnosis system is provided with a fault, it searches for a similar fault. Then it decides whether or not the fault has already been observed to locate the exact cause. The system compares the *input symptoms* with the already stored symptoms. Then a probable cause is identified. To be more exact about the fault the system user could be asked some additional questions.

An investigation needs be carried out to determine the form of the output of diagnosis (diagnosis statement). One additional feature of the Intelligent System is to advise the user on how to fix the fault.

Once the system is developed it has been tested and validated. A comparison has been made with an existing diagnosis system to evaluate the novelty of this study's system.

Automotive infotainment system is growing rapidly. More and more technologies and devices are integrated into the system. The main advantage of developing an Intelligent Diagnosis System for Infotainment & Multimedia is that human knowledge can be integrated into a machine. The system knowledge and its normal and faulty behaviors are known by some experts in the area. Those experiences are to be formulated in the system. Like a human expert, who can locate and advise about a fault when the symptoms are provided, the intelligent system can also do the job in an efficient way.

Until the last few years, automotive devices were mainly developed using CAN technology. Due to the slow speed of data transfer in CAN network (generally up to few hundred Kilo Bytes per second), a new technology named as MOST was introduced. For the last few years, more and more technologies including Navigation, Multi Media Player have been integrated into the MOST system. Use of MOST results in high speed data transfer (generally up to 25MByte per second).

To the best of our knowledge, there is not any fault diagnosis system which can handle new automotive systems. The complexity of this new system grows day by day. This makes it essential to build an efficient diagnosis system. The aim is to develop a new fault diagnosis system which will be able to handle the faults of the new devices and systems based on MOST.

So far all the diagnosis systems in the automotive area were developed based mainly upon the faults experienced during normal uses on streets. It means the diagnosis system is used in a completely developed automotive system. But this system is useful in any of the phases of automotive development, from planning to street use.

Chapter 2

Related Work

In this chapter an overview of available literature on fault diagnosis is provided so as to design and model an optimized fault diagnosis system for automotive infotainment systems.

2.1 Definitions of Common Terms

A *fault* can be defined as an abnormal condition that causes a functional unit to fail to perform its expected behaviors. A fault occurrence is related to several factors. The unexpected behaviors are visible to a user by some *symptoms*. Fault diagnosis is the investigation of the relationship between the factors, faults and symptoms.

2.2 Necessity of Automated Fault Handling

The importance of automating the process of fault handling and fault diagnosis is immense. Automated fault diagnosis process has many advantages:

- The system designers receive exact feedback of the system behaviors.
- Human knowledge can be integrated into a machine. The system knowledge and its normal and faulty behaviors are known by some experts in the area. Those experiences are stored in the diagnosis system. Thus the knowledge can be preserved.
- A human expert of a system can advise on how to solve a fault. Similarly, an intelligent system can also do the same job when the symptoms of a fault are provided.
- Possible problems are identified much earlier in the design process [48].

2.3 Infotainment System Description

In this section the Infotainment and Multimedia system, its components and their interfaces are described.

2.3.1 Database System

All the Infotainment & Multimedia faults are stored in a database. This database is one of the main sources for acquiring knowledge to utilize in the **I**ntelligent **S**ystem for **Fa**ult **D**iagnosis (ISFAD). Information and experience stored in the database are not well formatted. This necessitates a comprehensive study and analysis of the existing database system.

2.3.2 Infotainment Architecture

The infotainment system can be characterized as a distributed, heterogeneous hardware/software system, which provides functionalities and services to the customer via the interaction of several devices. The hardware components are interconnected by standardized car networks (e.g. MOST, CAN bus) and by wireless communication channels (e.g. Bluetooth).

An overview of the main components of Infotainment & Multimedia in a passenger car is presented below:

Head Unit

Head Unit (HU) is regarded as the master in the network. It contains one or more CPUs. The HU has at least one CPU for real time communication purposes (e.g. infotainment gate way) and at least one CPU for the user interface to interact with infotainment applications (Main CPU).

Infotainment & Multimedia Gateway

Infotainment & Multimedia Gateway contains at least a CPU (Gateway CPU) for real time communication with other peripheral components like memory, CAN transceivers, MOST transceiver and other components. The gateway CPU is connected to the Main CPU with an internal bus. As The Main CPU is not connected directly to the MOST command/control channel and the CAN network, all messages from the Main CPU to the outside world are sent and received over the gateway CPU.

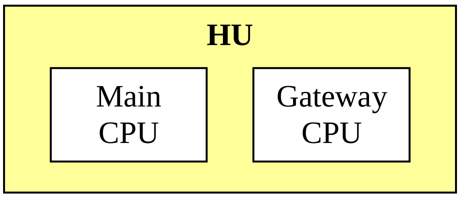


Figure 2.1: Head Unit Processors

CAN Network

One of the main tasks of HU is to control the basic vehicle functions. The gateway is not connected directly to the control units of vehicle functions. It is connected to the Central Gateway (it manages the other CAN (Controller Area Network) networks in the vehicle) by a high speed CAN (Backbone CAN). The HU is connected with the HU-CAN. Some control devices on the HU-CAN communicate only with the HU. There are some other devices for which the messages are only routed through the gateway from the HU-CAN to the Backbone-CAN and vice versa. The CAN messages and signals are specified in some matrices.

MOST Network

MOST (Media Oriented System Transport) is a high-speed network, which connects infotainment components with the central unit (HU) through the gateway. It is a closed-loop network. The gateway must be able to handle command/control data of MOST.

A MOST device is a physical entity which may include multiple functional components. These components are called Function Blocks (FBlock). For example, the MOST device Rear Seat Entertainment (RSE) consists of several FBlocks such as MultiMediaPlayer, HeadPhoneAmplifier, and AuxiliaryInput etc. An FBlock is identified with special ID names such as FBlockID.

In a MOST system, a specific FBlock may exist more than one time. To avoid any conflict among them, each FBlock is assigned with an instance identifier referred to as InstID. The combination of FBlockID and InstID makes an entity unique for addressing. This is termed as functional address. An FBlock consists of many functions, like Mute, Volume, Allocate etc in MOST FBlock HeadPhoneAmplifier. In order to distinguish between different functions, every function has its own Identity which is called FktID.

With the help of a function, certain operations are performed on respective properties or methods. The kind of operation is expressed by OPType, e.g. Start, Get, StartResult etc. Each OPType may contain several parameters. To perform a specific operation on a specific function block, a message of the following format is used:

<FBlockID.InstID.FktID.OpType (Data)>

To realize an application of a MOST device, the above structure is implemented according to the detailed guidelines in a function catalog. A function catalog lists all the FBlocks, instances, functions, OpTypes and parameters.

From the MOST application point of view, a MOST device can be said to be error free if the complete function catalog functions correctly. Therefore, it is of prime importance to test the function catalog thoroughly. This test is also called a Static Function Test, as a fixed set of functions are tested here.

The test of the function catalog starts by reading the Device Identity such as hardware/software/boot loader version from the MOST devices. Then all the implemented and non-implemented FBlockIDs (0x00 – 0xFF) are verified. Non-implemented FBlockIDs are tested to fulfill negative requirements where an appropriate error message is expected.

Each Existing FBlock in a MOST device has certain InstID(s). InstIDs are verified by keeping in mind the assigned instances for an FBlock, wild-cards (normally InstID 0x01n and 0xFF are defined as wild-cards) and all incorrect instances.

The FktIDs are verified for positive and negative test cases to ensure that project specific FktIDs exist, and the other FktIDs do not exist in the FBlock. In addition to the project specific function catalog, a general function catalog obtained from MOST Cooperation is used to decide which functions should be taken to test negative cases. Usually the unused functions in the general function catalog are considered for negative tests. The same test strategy is also applied for OPType verification.

Automotive infotainment system consists of several ECUs (Electrical Control Units). Some defined protocols should be developed to manage the connected devices. These sets of protocols can be termed as Network Management. Network management tasks include startup, shutdown, wakeup, power moding, addressing, allocation, de-allocation, audio management etc.

Startup behavior manages the by-pass close/open, timely response for NetBlock queries, registration of FBlocks, configuration control, resource de-allocation etc. NetBlock is a special function block, which provides functions related to the entire device. Each MOST device must contain this FBlock.

Shutdown behaviors deal with the proper saving of current settings, communication ability, attachment of FBlocks in the central registry, light off behavior, whereas wakeup behaviors set the guidelines for light travel time from input to output in ECUs, lock/unlock, light off etc.

Power moding behaviors deal with the device characteristics in under voltage and over voltage. It also defines the consumption of power during normal mode and sleep mode. Network management addressing ensures the correct sending and receiving of messages in the system.

Allocation, de-allocation and audio management are the areas where sources, sinks, resource management are handled.

The HU contains the MOST FBlocks and controllers, which are necessary for controlling the network behaviors of the HU.

Below is a list of logical interfaces for some common devices:

- Head Unit (through MOST)
- Speech Control System (through MOST)
- Mobile Phone (cradle or Bluetooth)
- Instrument Cluster (through HeadUnit)
- Amplifier/Sound (through MOST)

A typical MOST ring contains the following components:

i. Amplifier: Amplifier is a mandatory ECU in MOST infotainment system. This works as a sink of audio data.

ii. Telephony: Telephony contains a mobile Interface. This is an optional device in a MOST ring. The interface connects the Bluetooth-enabled mobile phone and other MOST components.

iii. Rear Seat Entertainment: Rear Seat Entertainment (RSE) is an optional device. It may contain MultiMediaPlayer, HeadPhoneAmplifier, and AuxiliaryInput.

iv. TVTuner: TVTuner provides reception for video and audio. This is an optional component in a MOST ring. A TVTuner can have variants of digital, analog or hybrid (analog/digital). The video output is sent to display and audio is sent to amplifier.

v. Gateway: This is a mandatory component in MOST infotainment system. Gateway is the interface between MOST and CAN network. MOST commands are translated into CAN commands whenever any command is sent from MOST to CAN. Similarly, CAN commands are translated into MOST commands if the target of the command is an element of the MOST network.

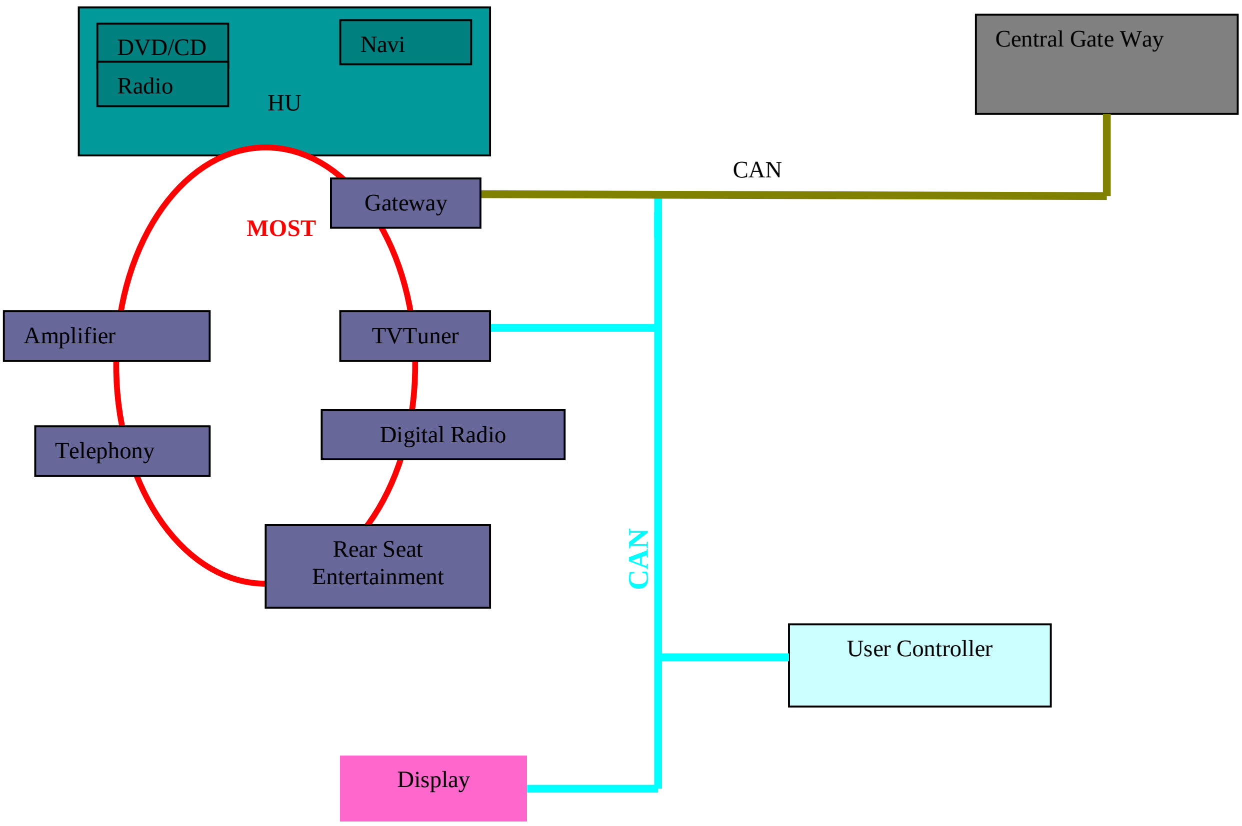


Figure 2.2: Infotainment & Multimedia Architecture

vi. Radio: Radio is a mandatory element of MOST. It can be an analog or digital tuner. This is a source for audio data which is transferred to an audio amplifier.

Software Download and User Controller (UC) are two commonly used terms which are briefly described below:

Software Download

Software download means flashing software into Infotainment & Multimedia devices. Providing an interface to download software into Infotainment & Multimedia devices permits the separation, to a certain degree, of the hardware development cycle from the software development cycle and adds new (updated or additional) functionalities without changing the hardware.

User Controller (UC)

The user control (UC) is a mechatronic interface between the head unit and the user. User commands can be provided through the movement of the buttons and pressure-operated rotary actuator. The commands are transmitted to the HU via CAN-bus.

2.3.3 Measuring Tools

In Infotainment & Multimedia systems, there are some tools and devices which are commonly used for the purpose of measuring different quantities which express different states and running conditions of the automotive system. Below is a brief description of the tools:

Optolyzer

Optolyzer is a very important tool to perform tests for Infotainment & Multimedia systems. It can be used to monitor the status of MOST bus. In a MOST ring, it can be used as spy, master and slave modes based on the types of handling of a message. There are several tools to analyze the messages of a MOST bus. OptoLyzer4MOST [51] is an example of such a tool, which can be used to analyze MOST messages. Based on the status message of the MOST bus, the system state can be determined. Optolyzer can trace almost all of the system behaviors.

CANalyzer

CANalyzer [50] is a universal development tool for CAN bus systems. This is used in observing, analyzing and supplementing data traffic on the bus line.

There are options in CANalyzer to work on byte level with bus-like raw data format and on the application level with the logical/physical data representation. A CAN database is used to convert raw data. CANalyzer contains a user-friendly database management program CANdb++. The basic built-in functions provide for an abundance of possible applications. These include listing bus data traffic (Tracing), displaying data segments of specific messages, transmitting predefined messages and replaying recorded messages, statistically evaluating messages, and acquiring statistics on bus loading and bus disturbances, as well as recording messages for replay or offline evaluation.

There is the possibility to add additional required functionalities with CANalyzer by means of user programming. A program block can be inserted at any point in the data flow diagram. An application-oriented language called CAPL (CAN Access Programming Language) is used as the programming language. Creation, modification and compilation of a CAPL can be done easily with an interactive development environment in CANalyzer.

CANoe

CANoe is a widely used development, test and analysis environment for CAN bus systems. Based on the input and output signals, the characteristics of a network node are described with the help of environment variables. Switch positions and dimensions (e.g. temperature, speed) of the network can be discovered with the environment variables. Control panels in the CANoe provide a user-friendly interface for the environment variables.

One can send commands to CAN and MOST bus using CANoe. With CANoe, the status of CAN and MOST buses can also be observed.

The data from CANoe is important for us as it stores test results. Test results reflect the real behaviors of the infotainment system. The test results are important in gaining knowledge for the development of the intelligent fault diagnosis system.

2.3.4 Communication Layers

This section describes the standard communication layer model. Taking this model as the basis, the communication layers of Infotainment & Multimedia in automobiles are described.

2.3.4.1 OSI Layer

In the OSI communication layer model, there are seven layers:

The **Physical Layer** describes the physical properties of the various communications media, electrical properties and interpretation of the exchanged signals. For example, this layer defines the size of Ethernet coaxial cable.

The **Data Link Layer** defines the logical organization of data bits transmitted on a particular medium. For example, this layer describes the framing and addressing of Ethernet packets.

Application Layer
Presentation Layer
Session Layer
Transport Layer
Network Layer
Data Link Layer
Physical Layer

Figure 2.3: Communication layer model

The **Network Layer** explains how a series of exchanges over various data links can deliver data between any two nodes in a network. For example, defining the addressing and routing structure of the Internet is one of the main tasks handled by this layer.

The **Transport Layer** describes the quality and nature of the data delivery. For example, this layer defines if and how retransmissions will be used to ensure data delivery.

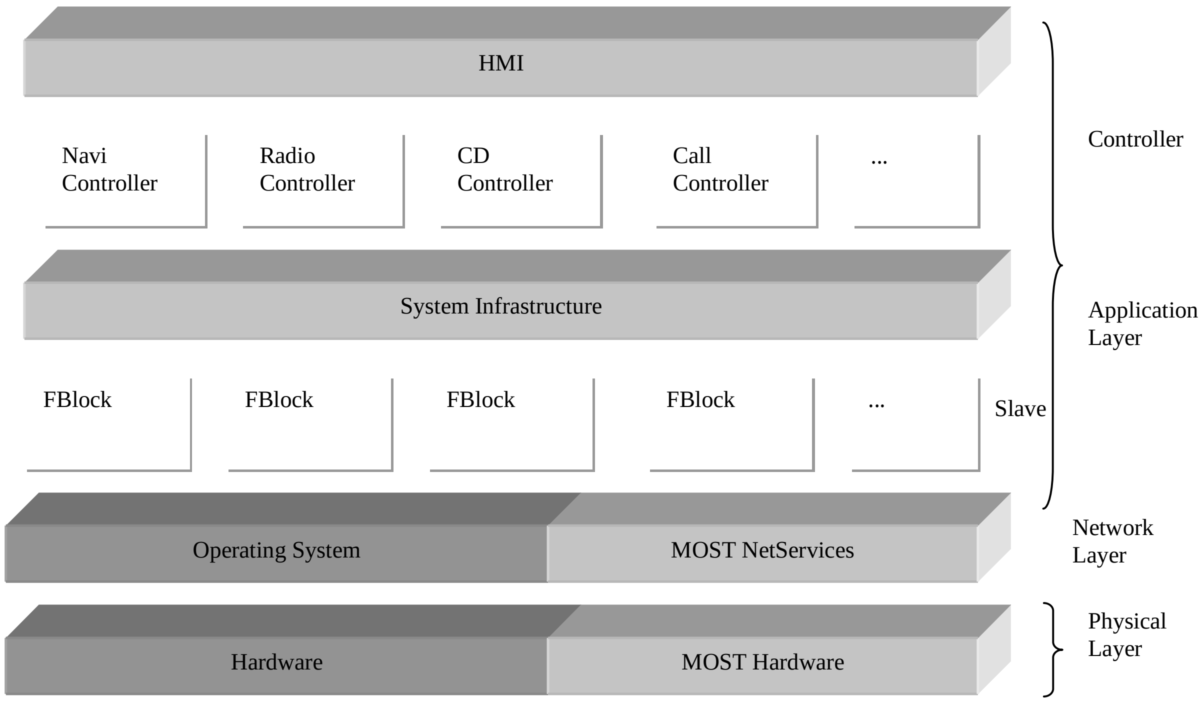


Figure 2.4: Communication layers in Infotainment & Multimedia

The **Session Layer** describes the organization of data sequences that are larger than the packets handled by lower layers. The tasks of this layer include how request and reply packets are paired in a remote procedure call.

The **Presentation Layer** describes the syntax of data being transferred. For example, this layer describes how floating-point numbers can be exchanged between hosts with different math formats.

The **Application Layer** describes how real work actually gets done. As for example, this layer implements file system operations.

Based on the above discussion regarding the communication layer model, a general layer model for Infotainment & Multimedia can be explained.

2.3.4.2 Infotainment & Multimedia Layer

This section discusses the Infotainment and Multimedia layering in an automotive system.

i. Physical Layer

Several Infotainment & Multimedia hardware platforms are distributed over the car and in the infrastructure. The devices integrated in the vehicle are interconnected by MOST and CAN buses. These in-car platforms constitute the physical layer.

The physical layer deals with the following:

Transmitter
Receiver
Device Connector
Cabling
Power Budgeting
Signal Timing
Diagnostic Requirements
Component Specification
System Description
Coordination

ii. Network Layer

The communication stacks and MOST/CAN network management functions as a whole constitute the network layer. The network layer deals with the following:

Net Services
Communication between devices is realized in the majority of cases via the MOST NetServices. There is also a Gateway to communicate with CAN devices. For further communication purposes via other networks additional protocol stacks, e.g. WAP, are implemented.

Transport Protocols - Transport protocols deal with control and segmented data communication over the control channel, the amount of data for such a connection, MOST High Protocol as transport protection for streaming and application data on transport layer level.

Boundary - The boundary for the distribution of synchronous and asynchronous channels is to set to a fixed number.

MOST network management level - MOST Address Handling, InstID Handling, and FBlock Naming.

Network Master - After the MOST network on a basic level is ready for communication (NetOn), the network master checks if it has a valid logical node address. It also manages the following behaviors of a device:

IP Address Handling
MAC Address Handling
Low Voltage Device Behavior

iii. Application Layer

The **application layer** is at the top of the system layer. It consists of the system infrastructure - including infrastructure software components (e.g. Resource Management - providing functionality) generally required in application components or required to ensure that the system as a whole functions properly.
It consists of general requirements, architectural requirements, and a detailed specification of functionalities and interfaces.

The application layer deals with the following:

- application components (controllers and slaves) focus on the functionalities and logics required for a specific application.
- the MMI framework is responsible for the handling of interaction between the infotainment and multimedia system and its users.

2.4 Infotainment and Multimedia Features

All features that the Infotainment and Multimedia system provides are realized by software components which are deployed on car devices and service centers. The features can be grouped as follows:

i. entertainment services (e.g. Radio, DVD, TV, Digital Radio (DAB))

ii. Information services (e.g. Navigation including route planning, dynamic route guiding and travel guidance)

iii. Telecommunication services (e.g. voice telephony, SMS, E-Mail, Remote Climate Control and Remote Door Lock/Unlock, Browser List)

iv. Advanced use of car functions (e.g. adjustment of external mirrors, climate control, seat adjustment)

v. Safety services (e.g. Emergency-call, Car Tracking)

vi. System administration services (e.g. resource management, channel allocation/Deallocation, software download, and engineering mode)

2.5 Fault Handling

To develop the intelligent system, it is necessary to analyze existing faults that are already stored in the Data Base (DB). These faults are experienced in different phases and in different times during and after development of an automotive infotainment system. One important task for this project is to devise a model to describe and classify the faults.

2.5.1 Fault Description

It is necessary to develop a well-devised fault structure to extract relevant information in such a complex system as automotive infotainment. There should be sets of guidelines and rules to narrate a fault. Advanced planning is the primitive condition for achieving the goal of storing faults in a better and more accurate way. Fault description schemes are proposed in the following chapters.

2.5.2 Fault Classification

Mark A. Thurber et al. [40], Puneet Kukkal et al. [39], O. Anton et al. [38], [35], D.Y. Chen et al. [34], Haapanen Pentti et al. [33], Friedemann Bitsch [22] have discussed the descriptive languages, effects and modes of faults to classify them. Classification is the basic task to develop an organized fault diagnosis system. The following chapters discuss the possible ways to classify faults:

2.5.2.1 General Faults

Mark A. Thurber et al. [40] have classified faults into *active* and *previously active* or *inactive* faults.

Active Faults
The newly stored faults are regarded as the active faults. When an abnormal condition is present within a system, there is an active fault.

Inactive Faults
An active fault becomes inactive if this does not exist anymore in the system. The fault has already been resolved. A record of inactive faults is preserved for future references.

Stable and Unstable Faults
For a reliable fault diagnosis system, fault recognition should be executed in an appropriate way. To achieve this goal, Mark A. Thurber et al. [40] have classified faults into stable and unstable faults. An over-sensitive fault detection method could indicate unstable faults (such as electrically unstable faults) that are not any real faults in the system. Therefore, it becomes important to determine stable faults. Faults that are determined by measuring parameters are considered stable faults when the estimating parameter exceeds the limiting value. Faults that are new should be recorded in the fault database. If the fault is already in the fault database, it is not considered and recorded again.

2.5.2.2 Faults in Communication Systems

D.Y. Chen et al. [34] have classified faults of a communication system based on their *nature,* *origin*, *persistence* and their *location* in the protocol stack.

Nature based Faults
According to the nature of faults, they can be classified as *accidental* faults and *intentional* faults. Accidental faults occur fortuitously. An example of this kind of fault is a failure of a node due to fire. Intentional faults are created deliberately. An example of intentional faults is breaking a node into parts so that the execution of some services is denied.

Origin Based Faults
According to the origin of faults, they can be classified into three categories - *phenomenological causes, system boundary* and *phase of creation*.

Phenomenological Faults
Faults from phenomenological causes can be classified into *physical faults* and *software faults*.

Physical Faults
Physical faults occur due to the abnormal conditions in physical components. *Node faults, power faults* and *communication link faults* are the physical faults. Node faults may occur in communication switches, routers and in other physical components. Power faults are due to the interruption of power supply. Communication link faults are due to the interruption of information transmission between two nodes. Link faults may occur due to cable-cut, interference and fading.

Software Faults
The software faults occur due to bugs and resource exhaustion. These can be classified as Bohrbugs, Heisenbugs, and faults due to resource exhaustion.

Bohrbugs always cause a failure for a particular operation [41]. Bohrbug is also called a permanent fault. These faults are easily reproducible, and they can be easily removed.

Heisenbug may or may not cause a fault for a particular operation [41]. Heisenbug is also called as a transient or intermittent fault. This error may vanish when the operation which caused the failure is retried.

Faults due to *resource exhaustion* may persist in the operating system, middleware, and the application software. Operating system resources (e.g. swap space and free memory available) are progressively depleted due to defects in software such as memory leaks and incomplete cleanup of resources after use. Periodically rebooting a process or a node may help prevent the software from crashing due to this kind of faults.

System Boundary Point of View Classification of Faults
Faults from the *system boundary point of view* can be of internal faults and external faults.

Phase of Creation Point of View Classification of Faults
Faults from the *phase of creation point of view* can be of DMI (Design/Manufacture/Integration) faults or operational faults.

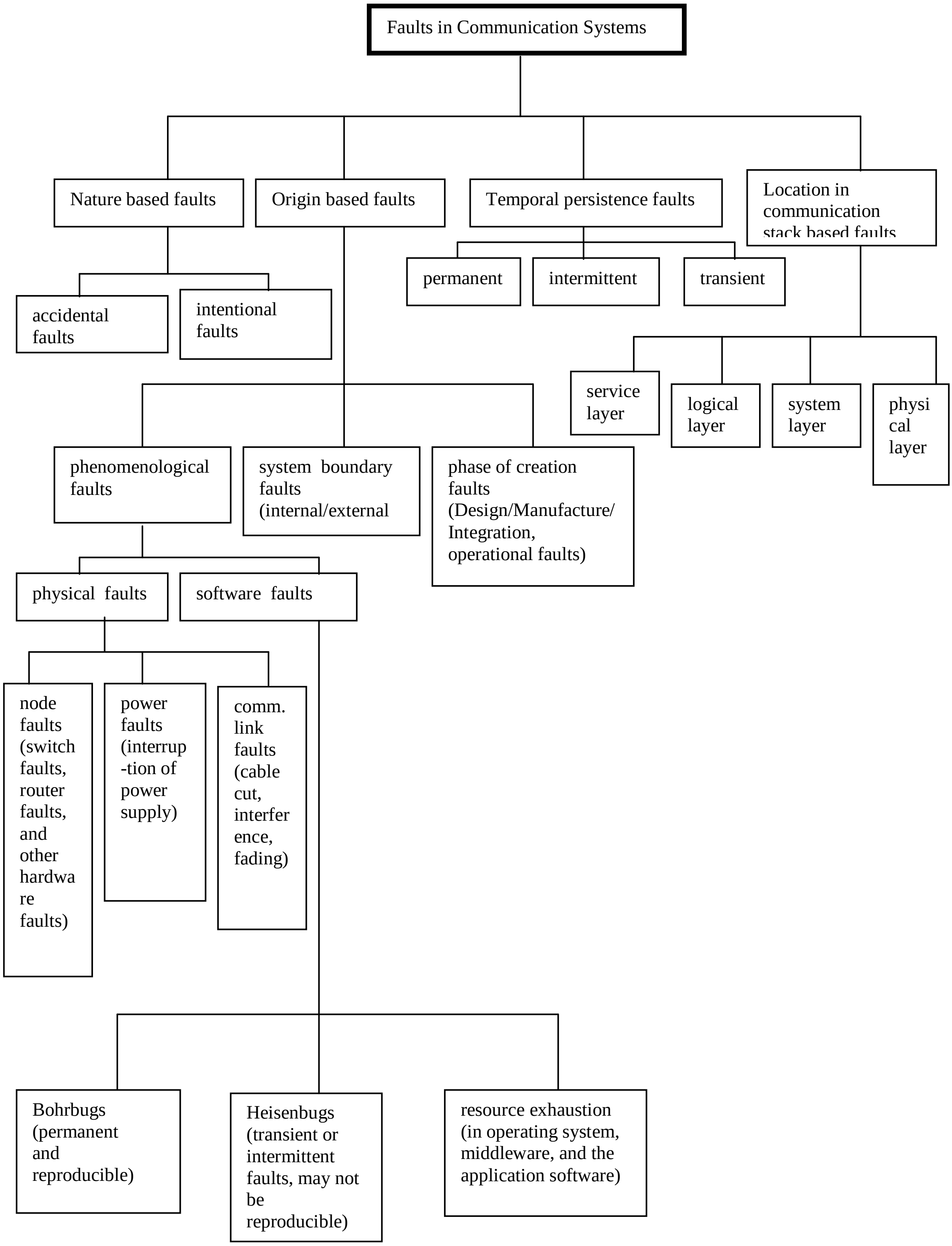


Figure 2.3 Classification of faults in a communication system

Temporal Persistence Based Classification of Faults
According to the temporal persistence of faults, they can be classified as - permanent faults, intermittent faults and transient faults.
Permanent faults are independent of any conditions. *Intermittent faults* depend on certain conditions and they occur for some extent of time.
Transient faults depend on certain conditions, but for a negligible time period.

Communication Protocol Stack Based Classification of Faults
According to the perspective of communication protocol stacks, faults can be classified into - service layer, logical layer, system layer, and physical layer [34].

Service layer faults may occur in switches, signaling points, and advanced intelligent network nodes.

Logical layer satisfies the needs of all services using the physical capacity. Transmission equipments are the nodes of logical layer and the channels are the links. Abnormal conditions in the components of logical layer cause logical layer faults.
System layer handles bandwidth and diversity of communication links. This layer is capable of reconfiguring around system and physical layer failures.

The *physical layer* provides physical and electromagnetic protection upon failures.
The classifications of faults in communication systems are summarized in Figure 2.3

2.5.2.3 Fault Classifications of Automation Systems

The authors of [33] have discussed different failure modes. These are summarized here as they are useful for classification of faults in Infotainment & Multimedia system.
Based on the analysis of three software projects, D.J. Reifer listed the following failure modes:

- Computational
- Logical
- Data I/O
- Data Handling
- Interface
- Data Definition
- Data Base
- Other

L. Ristord & C. Esmenjuad have listed five general-purpose failure modes at processing unit level:

- the operating system stops
- the program stops with a clear message
- the program stops without clear message
- the program runs but with wrong results
- the program runs with apparently right , but practically wrong results

R.R. Lutz et al. [42] have considered the following failure modes based on each input and each output of the software component:

- missing data such as lost message, data loss due to hardware failure
- incorrect data such as inaccurate data, spurious data
- timing of data such as obsolete data, too early arrival of data for processing
- extra data such as redundant data, overflow of data.

R.R. Lutz and R.M. Woodhouse classify faults based on each event (step in processing):

- halt/abnormal termination, such as hung or deadlocked at this point
- omitted event, such as event does not take place , but execution continues
- incorrect logic, such as inaccurate preconditions
- timing/order, such as event occurs in inappropriate order, event occurs too early or too late.

J.C. Becker & G. Flick have provided the following classes of failure modes:

- hardware/software stop
- hardware/software crash
- slow response
- startup failure
- faulty message
- checkpoint file failure
- internal capacity exceeded
- loss of services

2.5.2.4 Common Failure Modes

The paper by Princeton Plasma Physics Laboratory on guidelines for performance of a FMEA [35] has summarized general failure modes. The following are of interest:

- fails to remain in position
- fails to open
- fails to close
- fails to stop or start
- fails to switch
- erratic operation
- erroneous indication
- premature operation
- delayed operation
- erroneous input
- erroneous output
- loss of output
- shorted (electrical)

- open (electrical)
- leakage

2.6 Acquiring Knowledge from Database

To develop the ISFAD system, a vast comprehension of the Infotainment & Multimedia system should be acquired. Normal behaviors, faulty behaviors, fault symptoms and causes of faults of the system are of prime importance.

In Infotainment & Multimedia, there is a database system where all of the component and systems test results are stored. The database system is called Test Manager.

Test Manager
TestManager (TM) is a potential system for ISFAD. The experience of TM can be used as the basis for ISFAD.

Infotainment & Multimedia is a complex system with many features and components. Here the testing of each component is done first. Once the component test is complete, the system test is done. Tests could be software test, hardware test, service test etc. A tester knows the system behaviors. Expected behaviors and test behaviors are compared. If there is any dissimilarity, it is regarded as a fault. The symptoms and the causes of a fault should be stored in the TM.

Figure 2.4 and Figure 2.5 depict the basic features of TM. The first one describes the name of the device under test (DUT), features, occurrence and status of the fault. The second figure is more important for this study as it describes the fault in more detail. It has fault ID with fault heading, preconditions, misbehavior, expected behavior, specification reference and Manufacturer/supplier comments.

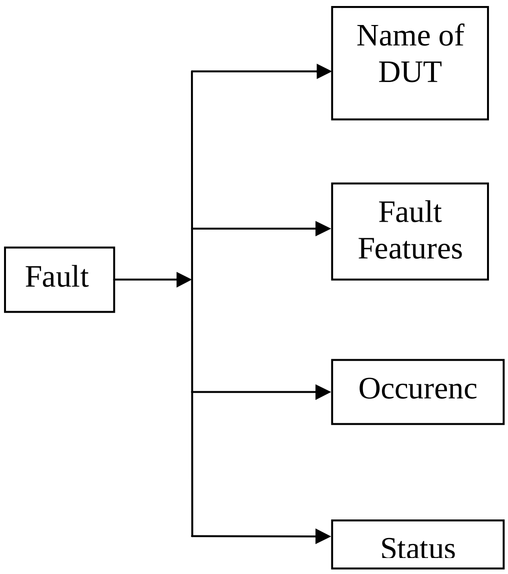


Figure 2.4: Fault Description (Short way)

The format of fault data in TM should contain the following characteristics.

A) There is a field in TM named as Actions. This is a mandatory field. This is important for describing the fault path. Necessary conditions are written in the Actions field in order to perform the test, which yields a fault. Actions should contain as many key words as possible.

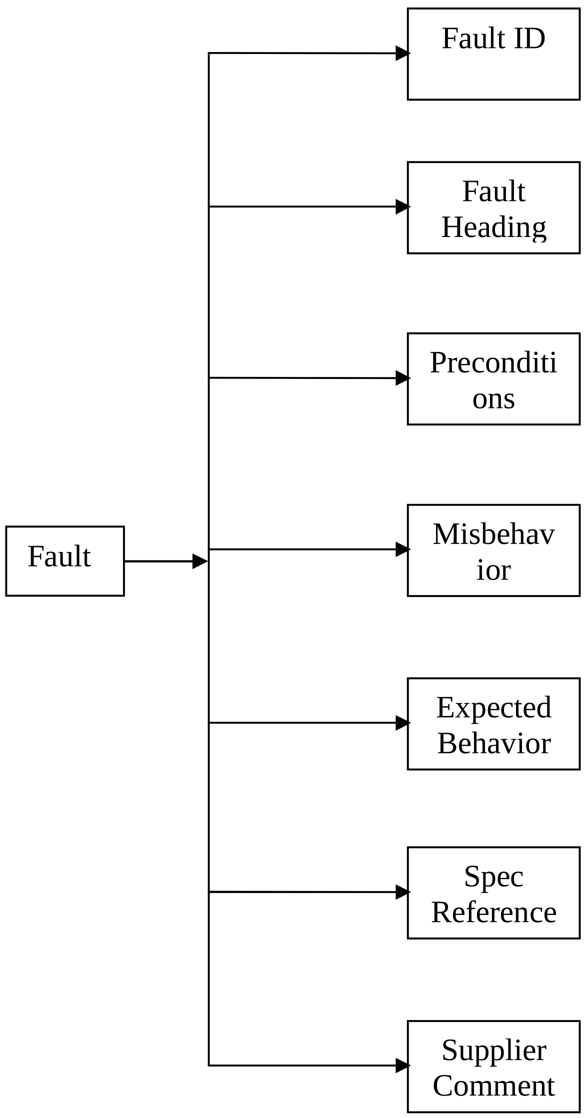


Figure 2.5: Fault Description (Detailed)

B) A Misbehavior/Reaction field describes the symptoms of a fault. This field should be filled with some predefined keywords. For example, a fault symptom could be described as “HU does not wakeup”. Generally a textual description of a fault is not efficient for a system like ISFAD. It would make the ISFAD system more efficient if information can be provided as precisely as possible. Describing a fault symptom with some predefined key words (e.g. choosing words from lists of words) makes the fault more suitable for direct use in ISFAD. For the above example, *HU, NOT* and *WAKEUP* are the key words to describe the symptom. Due to its importance for fault description, this field should also be made mandatory.

C) The Expected Behavior field is useful for distinguishing the component or system behavior in normal condition. Expected Behavior should be made mandatory. This information could be useful during system design. This could be described with texts.

D) The TM should be organized in such a way that the stored test results can be used more efficiently and effectively. In the current form of TM, a fault is described with its symptoms. However, the main drawback of TM is that generally one can not know the cause of a fault. It is difficult to rectify a fault without knowing the cause. This makes TM an ineffective system to store complete information about a fault. This necessitates adding a new field in the TM named as Defect Cause. This field could also be filled with some predefined key words.

2.7 Document Classification Method

For this study, one needs to classify faults based on the nature of a fault. The textual description of a fault is taken as input to make decisions regarding classifications. Document classification techniques are now explored and it is necessary to perform a comparative study to make use of the similar techniques for fault classifications in automotive infotainment systems. Currently, much research is being carried out in the field of document classification. Heide Bruecher et al. [55] and Dumais et al. [56] have compared and summarized different document classification methods and techniques.

The first step in document classification is to preprocess the document. Preprocessing includes feature extraction, feature selection and document representation.

In feature selection, a list of terms is generated to describe the document properly and correctly without undermining its expressions. The document is parted and a list of words is noted. These words are primary features. Then the unnecessary and meaningless (with respect to information content) words are filtered out from the document. Stop word removal [57] is one of the most popular techniques to increase the information contents in the document. With this method a list of words are tabled. Then the same words are eliminated from the document. A stop word list is created in Appendix I for automotive system.

Word stemming is another method to increase the information content of the features [58] [59]. Word stemming extracts the root word from a word in other forms. Words with suffix or prefixes may have same root. These different words are considered to be a single feature. The words plays, playing, played all have the root play. A list of stemming words for automotive system has been built up in appendix II. In the list, the similar words are stated inside the first bracket. To clarify the list, let us take the example of word ‘Play’. Words playing, plays and played are always mapped to word ‘Play’.

After filtering the meaningless words, features are selected according to their content of information. Chih-Ping Wei et al. [60], YongSeog Kim et al. [61] and Mark Last et al. [62] have worked on feature selection method. The most meaningful features are sorted out by some statistical means. Here some common terms such as term frequency (TF), inverse document frequency (IDF) and their combination (multiplication of term frequency and inverse document frequency TF*UDF) are used. Each word receives a score based on TF (important words occur more than the unimportant words), IDF (the rarest word gets the highest informative weighting). Then the combination of TF and IDF are used to determine the top words with the high scores.

After determining the top scored feature words, the document is represented by a feature vector of elements which are the number of features left after applying the selection process. The whole document can be represented by the following vector –

$$A = \begin{bmatrix} A_{11} & A_{12} & A_{13} \cdots & A_{1j} \\ A_{21} & A_{22} & A_{23} \cdots & A_{2j} \\ A_{31} & A_{32} & A_{33} \cdots & A_{3j} \\ . & . & . & . \\ A_{i1} & A_{i2} & A_{i3} \cdots & A_{ij} \end{bmatrix}$$

Where the element A_{ij} represents the occurrence frequency of feature j in document i. The frequency corresponds to the number of TF, IDF and TF*IDF. The values of TF, IDF and TF*IDF greater than zero are considered as 1s. Thus, an element of the vector is represented by binary values. If the feature is present, then the vector element assumes the value of 1 and 0 otherwise.

After determining the above vector, the documents are categorized into different document sets. For categorization decision, different methods such as decision rules, neural network, and fuzzy rules are used.

2.8 Diagnosis

Fault diagnosis has become an important area of research during the few years. The automobile is a crucial field where fault diagnosis is given a special attention. A diagnosis system is capable of identifying the faults of a system by investigating the observable effects (or symptoms). The system is referred to as *intelligent* if it can provide similar diagnosis results as a human *expert* could for the same fault. The Intelligent System categorizes the fault into a diagnosis class and identifies a probable cause based on the fault symptoms supplied.

The aim of diagnosis is to detect a fault and to classify it [48]. Detection is the confirmation whether a fault is a real fault. Classification is to determine the exact cause of a fault. Diagnosis is characterized as fault-class discrimination, where the classes consist of the faults of the various smallest units [48].

Ahmad T. Al-Taani [65] has devised an expert system for car failure diagnosis. This system has several main blocks – user interface, knowledge base, inference engine and language selection.

User interface is a question-answer block where the user is asked questions and he has to answer with true or false (yes/no). Knowledge base is a database where different rules and conditions are defined. Expert knowledge is the main source of this knowledge base. The Inference engine block decides on the rules to be applied based on the nature of the fault. Then it provides an appropriate solution. The shortcomings of this system are its limited handling of a certain number faults and being a time consuming task for creating rules. The goal of the new system ISFAD is to handle any kinds of faults and it is not restricted to providing solutions for a certain number of faults. Therefore, the system developed by Ahmad T. Al-Taani [65] is not suitable to achieve the goal in this study.

F. Cascio et al. [67] have developed an on-board diagnosis system for automotive system. A modified version of qualitative model [68] [69] [70] [71] has been used by F. Cascio et al. [67]. This diagnosis system determines a set of significant cases which are faced on-board. It runs the off-board model-based diagnosis system. After calculating the set of candidate diagnosis for every case, the corresponding fixing action is generated. There is an option to use a learning algorithm for system improvement. To use this system, separate hardware and software are needed to acquire data from Electronic Control Unit (ECU). The goal here in ISFAD is to develop a stand-alone system where a user provides input data. Thus the system proposed by F. Cascio et al. [67] does not fit with the requirements in this study.

Chapter 3

Fault Classification of Automotive Infotainment & Multimedia

In this chapter, new fault classification schemes are proposed and the best scheme for fault diagnosis is chosen.

A fault can be described from different points of views. To understand a fault clearly, it can be associated into different classes of Infotainment & Multimedia system. Based on the different characteristics of a fault, it can fall into different categories of an Infotainment & Multimedia system. The following are different perspectives in which a fault can be divided:

3.1 Fault Classification based on System Component

In this scheme all the faults are classified into six main groups - general problems, MOST defects, HMI Defects, Diagnosis Defects, System Defects and Network Management. This classification is based on the feature list of the Infotainment & Multimedia system.

During the development phase of an infotainment system features are planned. In a broad sense, a fault can be classified into general faults, MOST faults, Man Machine Interface (MMI) faults, diagnosis faults, system faults or network management faults.

General faults may contain wrong documentation, multiple entries of any commands from the head unit, no bug errors, low performance, usability, wrong user input, design error and system malfunction.

MOST faults may contain FBlock not available, FktID not available, no answer, OpType not available, parameter not available, redundant message, undetermined state, unstable system communication, wrong address handling and device de-registration.

MMI faults may contain font graphical interface, hard key, soft key, style guide, user and navigation errors.

Diagnosis faults may contain application CAN protocols, general MOST, protocol service not available, Diagnostic Trouble Code (DTC) errors.

System faults may contain core damp, dead locked, hardware error, not implemented, software defect and unstable system errors.

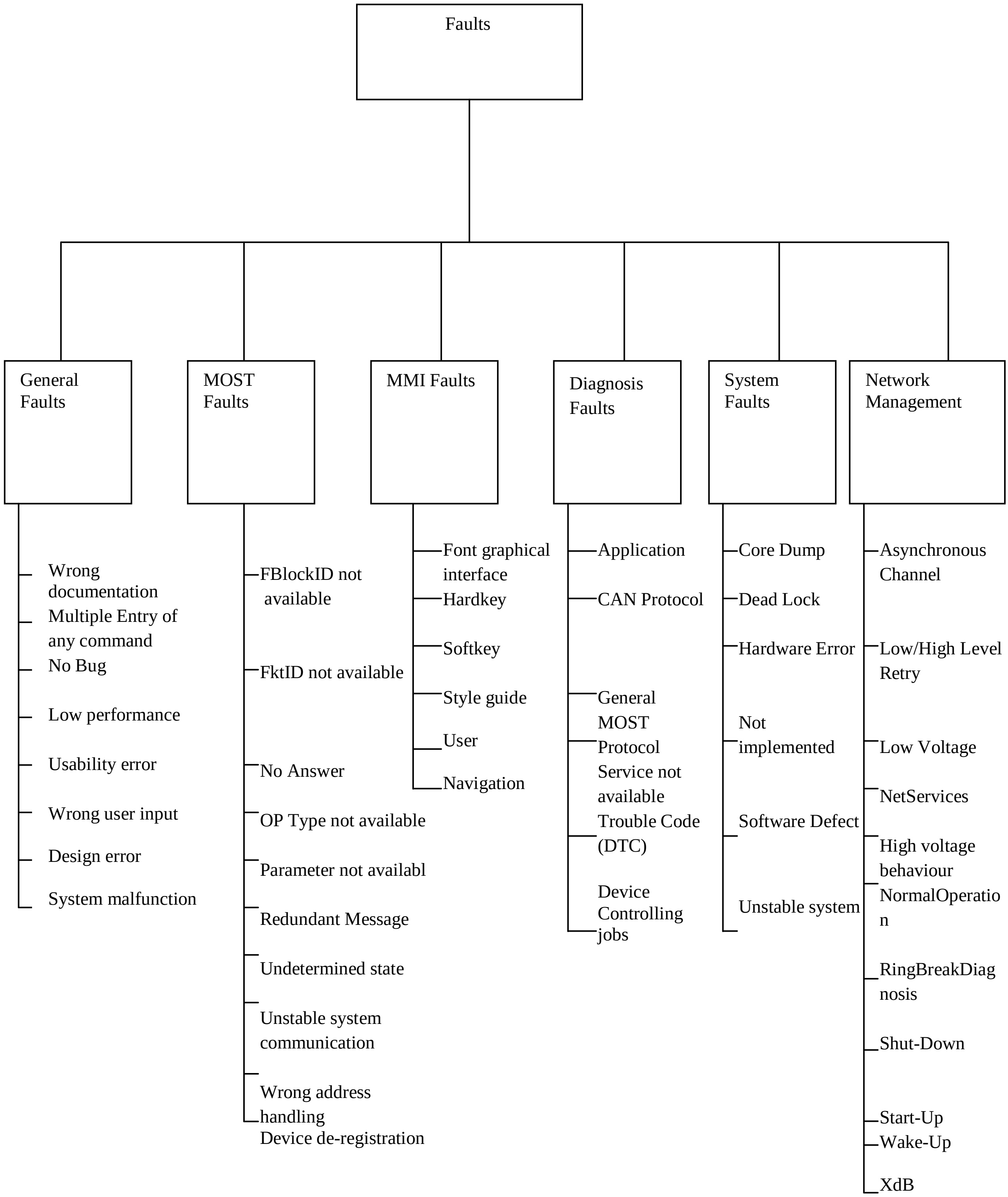


Figure 3.1: Feature-based fault classification

Network management may contain asynchronous channel, Low/High level retry, net services, high voltage, normal operation, ring break diagnosis, shut down, start up, wakeup and xdB errors.

This scheme is based on the earlier version of fault classifications. This is too general rather than being specific. But localization is the prime goal of fault diagnosis. The modern Infotainment & Multimedia system is becoming more and more complex with newly added features. Scheme 1 is too simple. Thus it is unable to diagnose all the faults in the system.

3.2 Fault Classification based on Origin

Describing a fault in a clearer way is very important for the task. A fault should be represented in such a way that there is a straight indication of the origin of the fault. To make a fault more understandable, it could be classified into a fault-category.

Fault diagnosis is divided into several steps. The first step is to make a fault classification scheme. One can attain a good grasp of a fault with this scheme. This necessitates drawing a complete path of a fault. In this scheme of fault classification a complete sketch of a fault should be performed.

In this scheme all the Infotainment & Multimedia faults have been classified into three groups - communication system faults, general faults and software based system faults.

General Faults
General faults are active, inactive, stable and unstable faults. A current fault is regarded as the active faults. An earlier fault, which does not exist anymore, can be described as an inactive fault. A fault, which continually exists, is called stable fault. An Infotainment & Multimedia fault that can be seen under some special conditions is called an unstable fault.

Software Based System Faults
Software based system faults can be computational, data management and interface faults.

Problem of this Classification
A fault could be in hardware or software or both. This scheme is unable to answer whether a fault is a software or a hardware fault. Scheme 2 can not always detect the exact point of a fault.

If a fault occurs in Speech System (SS) due to a design problem in software, this classification can not localize the cause of the fault. This is due to the fact that a software fault is under the sub-class of phenomenological faults whereas design fault is under the sub-class of phase of creation faults.

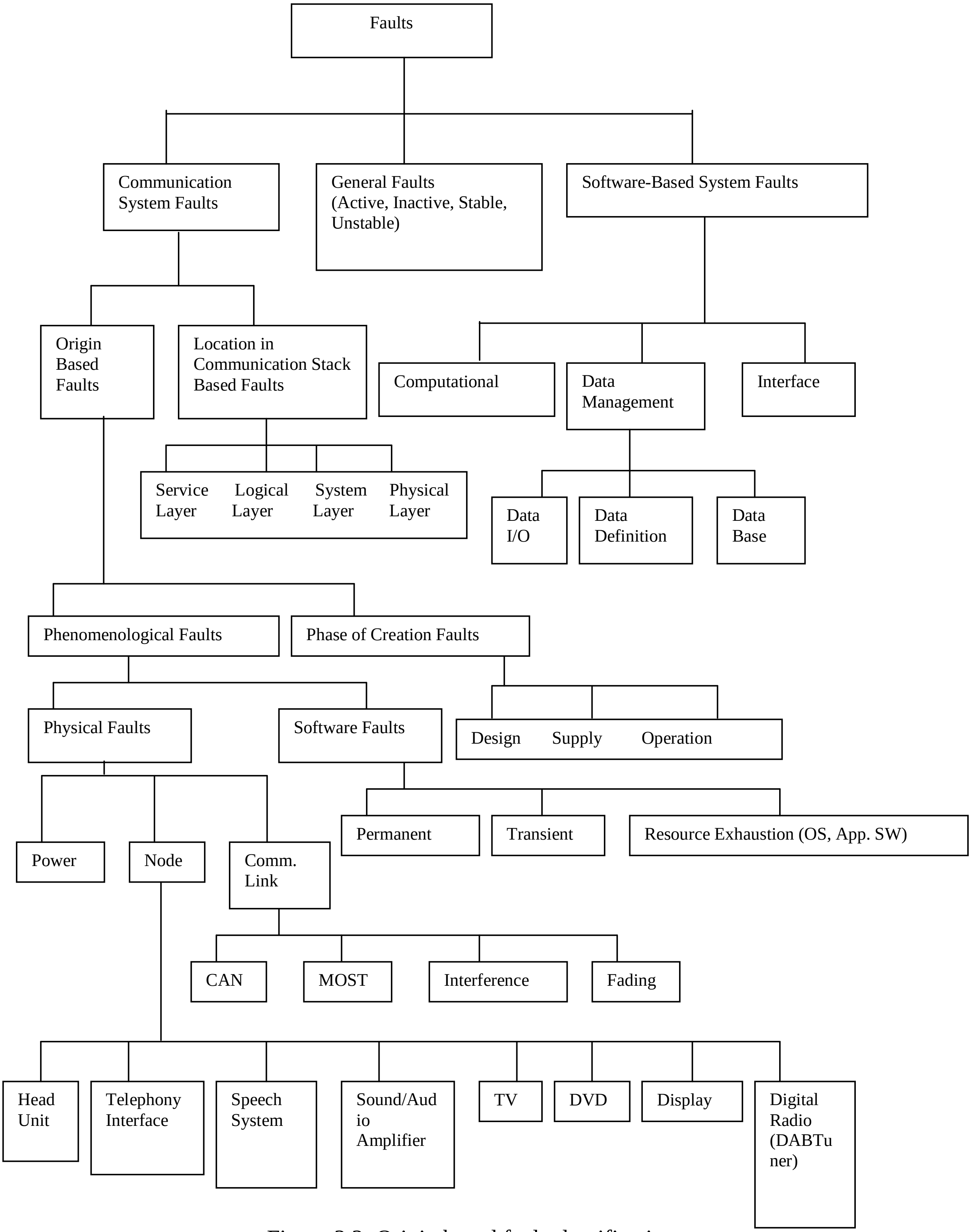


Figure 3.2: Origin based fault classification

3.3 Fault Classification based on Communication in MOST Network

This classification is based on communications among different entities in the infotainment and multimedia MOST network.

Fault classification based on communications firstly categorizes all the faults into two classes - origin based and communication stack based faults.

Origin based faults contain node, power, communication link or interface errors.

Communication stack based faults contain the faults in service layer, logical layer and physical layer.

The errors in node contain faults in TV, speech system, human machine interface, SDARS (Satellite Digital Audio Radio System), DABTuner, Multi Media Player, Rear Seat Entertainment, head unit.

Communication link faults contain CAN or MOST errors. CAN or MOST errors may come from hardware or software errors.

Interface faults may come from communication link to node (e.g. DAB – MOST interface), node to node errors like HU-Gateway.

Table 1: Examples of faults in a wireless network

Layer	Faults
Service	fault of home location register causes loss of roaming service
Logical	loss of base station causes service loss
Physical	loss of voice or data causes service loss

Table 2: Abbreviations

Abbreviation	Meaning
H/W	Hardware
UMI	Universal Mobile Interface
S/W	Software
SS	Speech System
HU	Head Unit
TGW	Telematic Gateway
Navi	Navigation

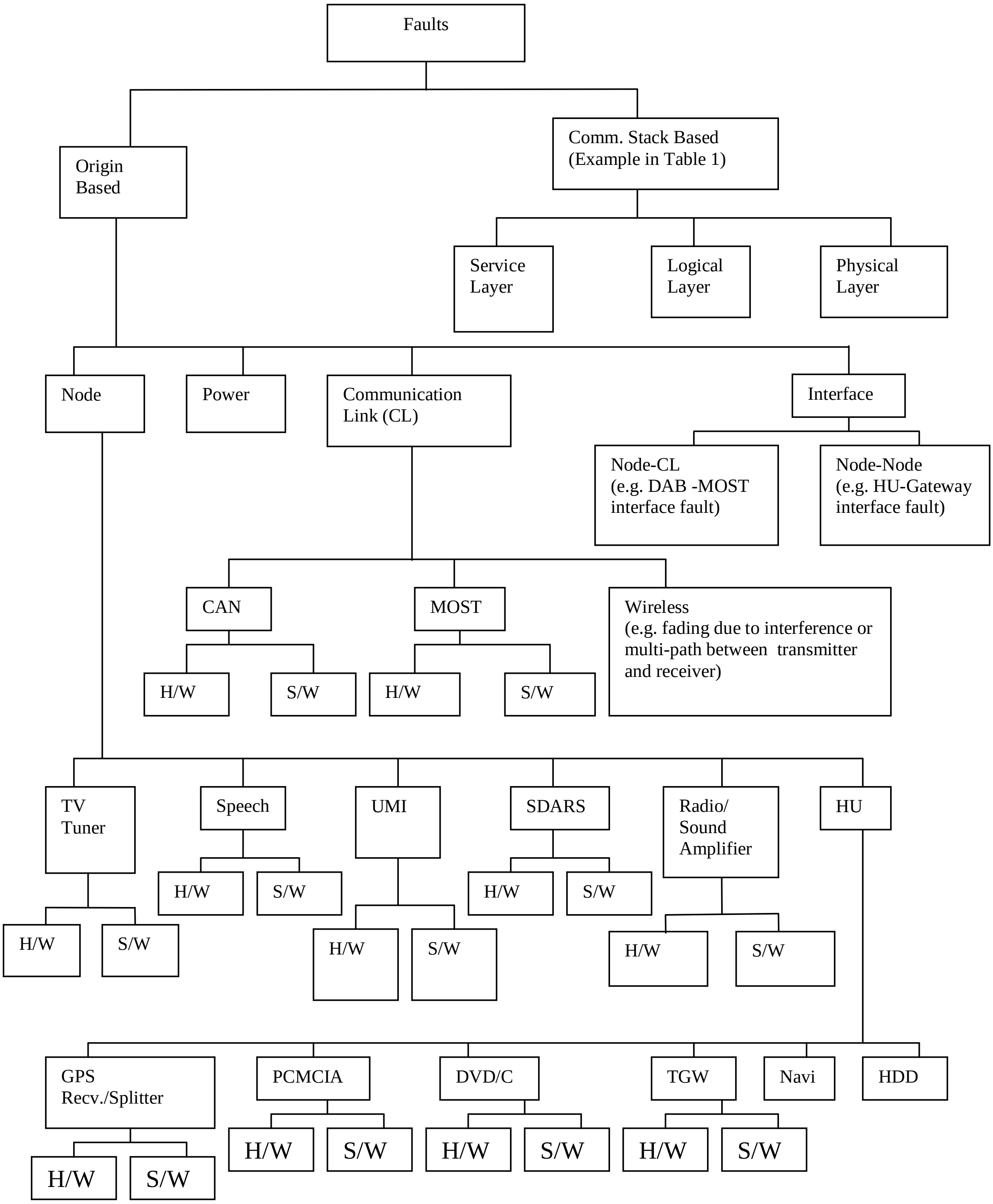


Figure 3.3: Faults based on communication in MOST Network

3.4 Fault Classification based on Severity

Faults can be classified according to their importance. Based on this classification, it can be ascertained whether an infotainment device should still be included in the infotainment system. Figure 3.4 depicts the fault classification scheme based on fault severity.

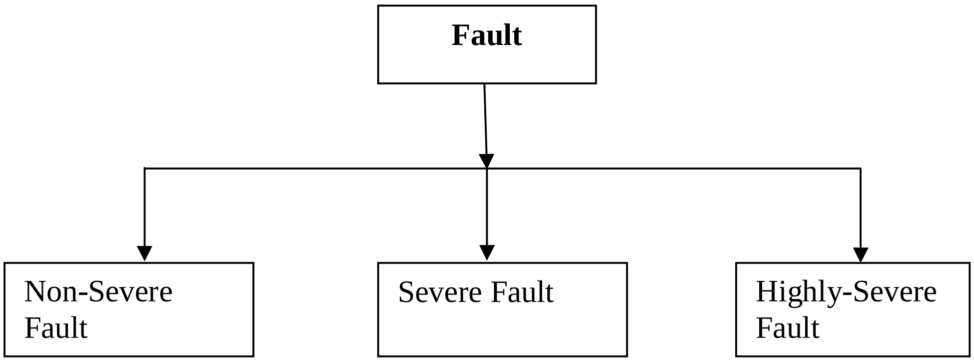


Figure 3.4: Fault scheme based on fault severity

Non-severe faults do not have a major effect on the proper functioning of the system. These faults can be ignored. Severe faults have a substantial effect on the functioning of the system, although the system is still in operation. Highly severe faults make the system inoperable. If there is any highly severe fault, the device responsible for the fault should be removed from the system so that the rest of the system can function properly.

Chapter 4

Fault Visualization in Automotive Infotainment System

4.1 Views of Faults

A fault in the Infotainment & Multimedia system can be analyzed in different ways. To be more certain about a fault, a scheme for describing a fault in different ways should be devised. By using the knowledge of earlier faults it is easy to be certain about it. When a fault is encountered in the Infotainment & Multimedia system, a thorough and comprehensive analysis needs to be carried out, taking previous faults into consideration in order to identify the cause of it. A fault is compared with earlier faults to gain different perspectives of the Infotainment & Multimedia system. A fault can resemble more than one earlier fault. Having compared this fault with other faults for the purpose of gaining different views, a decision can be made. The decision is taken based on the similarity. Using the results of different views for similarity testing, a clear description of the fault can be provided in order to be certain about the cause. We choose the highest “similarity” for the fault. It follows that, based on the causes of the similar faults, a final decision is taken to describe the fault.

4.2 Fault classification based on Test Purposes

Varieties of tests are performed to verify the functionalities of all Infotainment & Multimedia components and systems. Each test has some aims. Based on the purposes of a test, faults can be classified. When a test fails it generates a fault report. This report should contain the purpose of the test. With the information about the purpose of a test, it is easy to analyze the test result if it fails.

It is possible that a similar fault symptom can occur for different purposes of a test.

4.3 Combining Faults of Different Views

Faults described from different points of views should provide a combined result. Some probabilities can be determined for different views.

Figure 4.1 depicts the reliability factor after taking into account all the probabilities from different points of views. A fault is analyzed into four categories –
Feature Point of view – it has the sub groups of entertainment, information, telecommunication, advance functions, safety and system administration.

Communication Layer point of view – it has the sub groups of physical, network and application.
Phase creation point of view – it consist of design, supply and operation.
Temporal persistence point of view – it has sub groups of permanent, intermittent and transient.

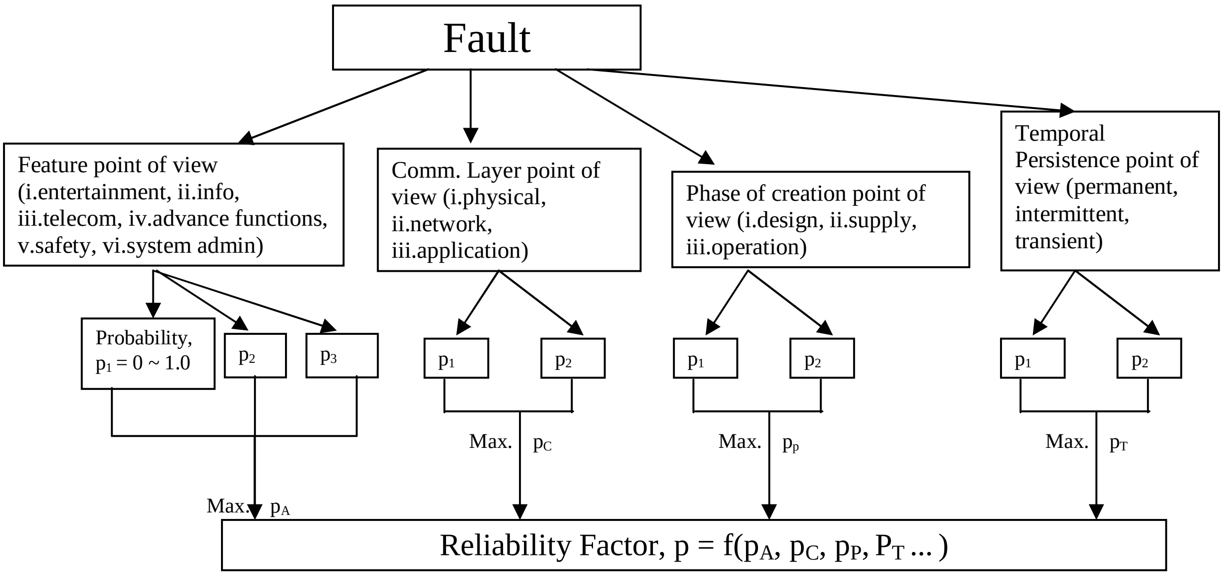


Figure: 4.1 Combining Fault Factors

Each of the four categories provides a probability factor for matching a fault. The factors are then combined mathematically to work out the reliability factor of the fault.

4.4 Fault Description Picture

Suppose the configuration of a system is like in Figure 4.2.

Suppose the behavior of a sound amplifier were to be tested in the above configuration. The proper functionality of the steps to increase sound volume of TV should be checked. It is observed that there is not any increase of sound after a certain volume. To analyze the fault to a further extent one can investigate further into the technical depths. As for an example, assume that there are in total ten volume steps from vol1 to vol10.

To investigate the fault in more detail the following steps are carried out:

1. At first check the MOST message at point A.

    Command Audio_Settings can be seen on the MOST at A?

        If YES then do the following

2. Build a fault picture using “Mute” button in User Controller.

3. Check the above steps of option 1 for another component e.g. Satellite Radio.

1. Checking MOST message:

Checking the MOST messages at point A is the first place to start drawing the fault picture for the above fault case. At first check if the message Audio_Settings is available.

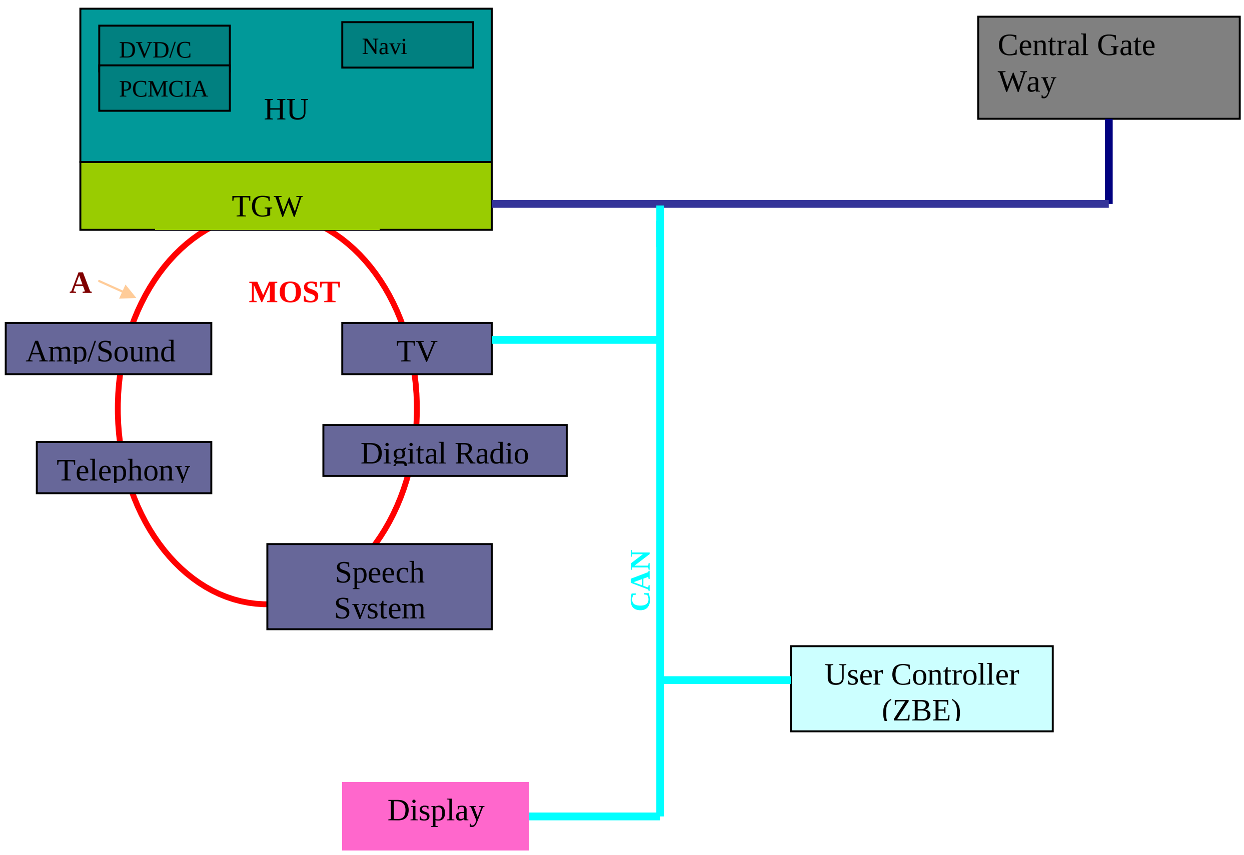


Figure 4.2: Infotainment & Multimedia Components for a particular fault

Case A:
If the message Audio_Settings is available, then there can be two cases -

Case I: Fault is probably at AMP.

Case II: Fault is probably not at – User Controller, CAN, HU, TGW, MOST.

For Case II check parameter at A. Observe the parameter whether it is Vol3 or Vol6. If the value is Vol6, then there is nothing wrong with the other devices except AMP. It means that the AMP is faulty. On the other hand, if the value is Vol3, the fault might be at User Controller or HU (Audio Management). Now check CAN message. If the message contains Vol3, then the fault is at User Controller. If the value is Vol6, HU is the faulty device.

Case B:
If the signal Audio_Settings is not available then do the following:

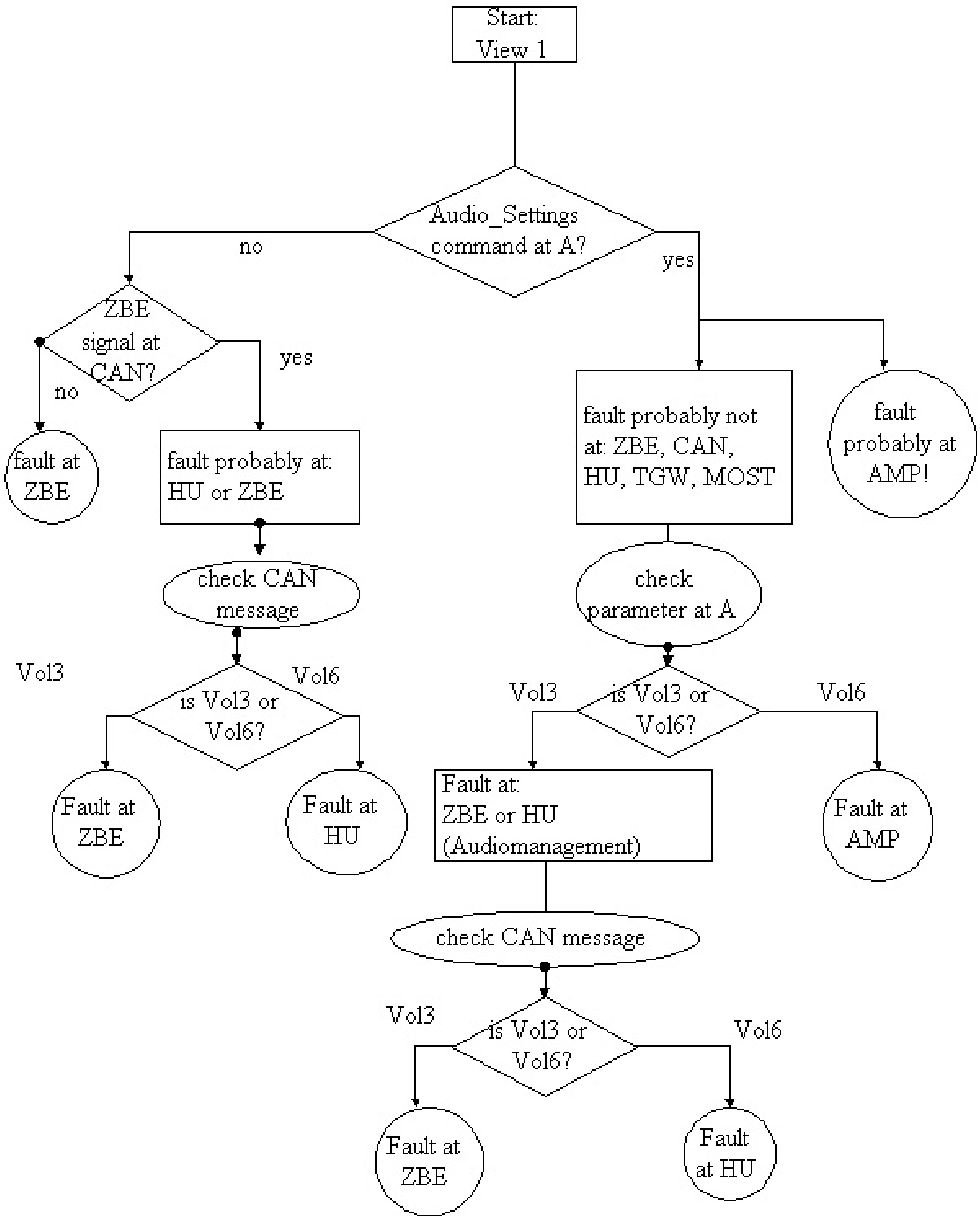


Figure 4.3: Fault Picture

Observe the User Controller message at CAN. If there is no signal at CAN bus, the fault is at User Controller.
If the answer is yes, then do the following:
The fault might be at User Controller or HU (Audio Management). Now check CAN message. If the message contains Vol3, then the fault is at User Controller. If the value is Vol6, HU is the faulty device.

The above description is just an example. It is necessary to devise a complete system with the requirements.

4.5 Fault Handling System

In this chapter a fault handling system is proposed. The steps to be completed for developing this system are:

1. Gather knowledge for the Infotainment & Multimedia components, Infotainment & Multimedia system and different behaviors. The knowledge can be acquired from the Test Manager (TM) for the initial case.

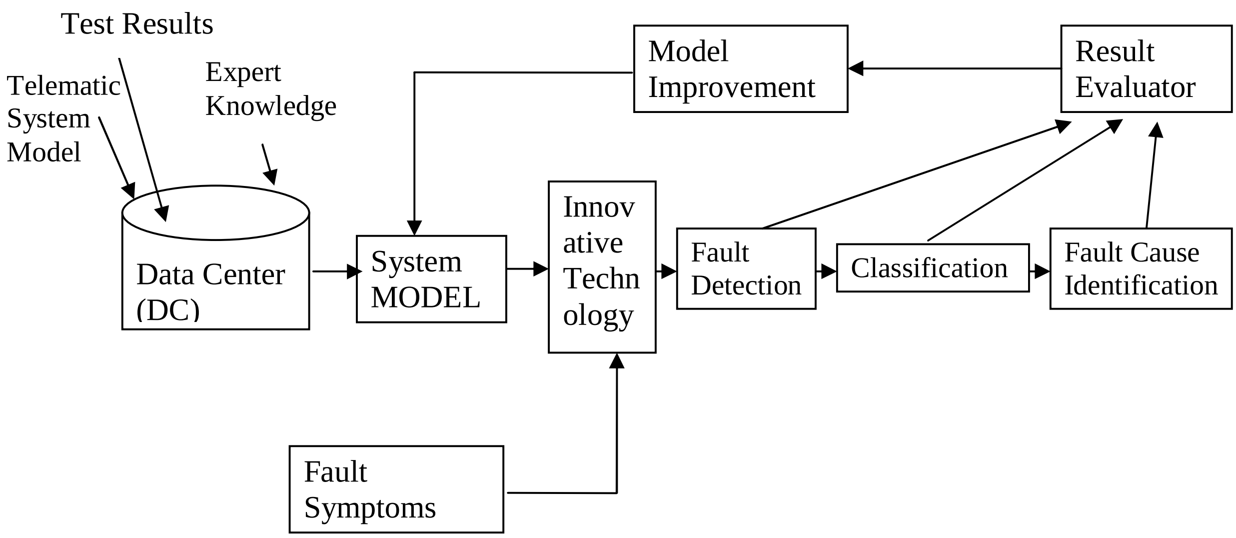


Figure 4.4: Fault Handling System

2. Gather test cases from TestManager. Those test cases work as the basis in the primary phase.

3. With the acquired knowledge, a Data Center (DC) is created.

4. Using Data Center (DC) a System Model is built.

5. Faults symptoms are incorporated into the system model.

6. After analyzing the symptoms facilitated by any technique, the fault is detected. As for an example, Fuzzy Logic (FL) might be used for decision making about a fault.

7. The system categorizes a fault into classes, considering it from different points of views.

8. Based on the classification results, fault description pictures are generated. With the help of these fault descriptions, a clear understanding of it is achieved. The fault pictures show the state of a fault, what is happening, how and what the causes are, how and why a specific group of people are responsible for the fault.

9. A report on the probable causes of a fault is generated.

10. Based on the results of the system, an evaluation is performed.

11. The evaluation result is used to improve the model. Result Evaluator is the indicator of the performance of the system. Feedback is provided to the system model to improve it. This makes the system model flexible and dynamic. Due to the dynamic property, the system model passes through a continuous modification and improvement. This enables the possibility of expansion for the inclusion of new demands.

Chapter 5

System Design Techniques

In this chapter the possibilities and usefulness of using different techniques for this system are discussed. Then this system, its design procedures and the functionalities are proposed.

5.1 Feature Selection

Feature selection is one of the main steps in similarity matching of faults. As described in chapter 2, we apply a stop word list to filter out the meaningless words. A list of stop words has been created in Appendix I. This list was created keeping in mind the existing standard fault description language in automotive infotainment systems.

Word stemming is a method where lexically similar words are listed together. Here, the words with affixes and suffixes are converted into root words. This methodology overcomes the limitation of words with the same meaning being categorized into different classes. A list of stemming words is created for automotive infotainment system in Appendix II.

Both the lists of stop words and stemming words were developed with the help of experienced system engineers in automotive infotainment system.

5.2 Pattern Recognition

Based on highly developed skill after sensing the surroundings, humans are capable of taking any actions according to their observations. By observing the nature of human intelligence, a machine can be built to do the same job, such as identifying hand writing, post code, voice, finger print, DNA, human face etc.

A pattern is an abstract object such as a set of measurements describing a physical object. This is an entity with a given name such as hand writing, a sentence, human face etc. Pattern recognition consists of several steps such as observation of inputs, learning how to distinguish different patterns and making rationale decisions in categorizing patterns.

Shmuel Brody [63] has summarized the concepts of pattern recognition and their uses in similarity matching. Human detected patterns contain many relevant and irrelevant data. The most important task in pattern recognition is to find out the meaningful patterns and to disregard the irrelevant subject matter. The fields of area of pattern recognition range from data analysis, feature extraction, error estimation, error removal, cluster analysis, grammatical inference and parsing.

Faramarz Valafar [64] has discussed pattern recognition techniques in data analysis. Clustering is one
of the most commonly used recognition techniques. Data are grouped into clusters or groups in clustering. K-means clustering is a widely used algorithm for data clustering. In k-means similar algorithm patterns are partitioned into the same group. All the data are classified into any of the k clusters or classes. Then the mean inter and intra-class distances are determined. The last step is to maximize the intra-class distance and minimize the inter-class distance. This is an iterative procedure where data is moved from one cluster to another. This process continues until optimized distances of intra-class and inter-class are found.

In pattern recognition different techniques are applied for similarity matching. For this work it is necessary to discover optimized techniques and algorithms for similarity matching, fault classifying and fault cause detecting.

5.3 Ranking Algorithms

PageRank algorithm is a widely used algorithm to rank web pages according to their importance. The algorithm is described as following **–**

PageRank is a link analysis algorithm to rank a web page from a set of pages according to its relative importance. It provides a numerical weighting to each of the page elements in the set. This weighting is called PageRank of E which is denoted by PR(E).

PageRank was introduced by Larry Page at Stanford University to develop a new search engine in the web. The ranking of a page depends on the number of links of the other pages to that page.

PageRank is a probability distribution which shows the likelihood that a user randomly clicking on the links finds a specific site. This probability ranges from 0 to 1. A PageRank of 0.8 means that the probability of reaching a specific site by randomly clicking on a set of links is 80%.

A set of five web pages is assumed: A, B, C, D, E. The initial probability is distributed evenly among these pages. Therefore, each of the pages will get a PageRank of 1.0/5.
It means,

$$PR(A) = PR(B) = PR(C) = PR(D) = PR(E) = 0.2 \qquad \text{(i)}$$

Now suppose the following scenario:

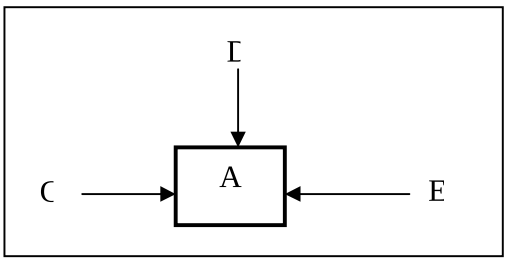


Figure 5.1: Inbound link of page A

Page A has inbound links from Page C, D and E. Thus, the PageRank of page A

$$PR(A) = PR(C) + PR(D) + PR(E) \qquad \text{(ii)}$$

Page C has other outbound links to page E, page D has other outbound links to B, C and E.

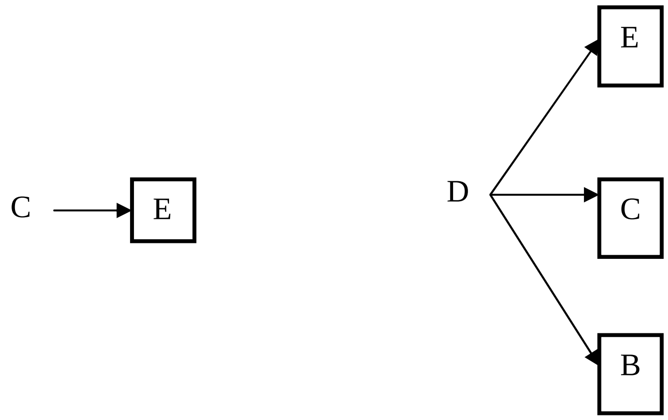


Figure 5.2: The Outbound links of page C and Page D

The value of the link-votes is divided among all the outbound links of a page. Thus, page C contributes a vote weight of 0.2/2 i.e. 0.1 and page D contributes a vote weight of 0.2/4 i.e. 0.05.

Thus, the equation stands in the following form:

$$PR(A) = \frac{PR(C)}{2} + \frac{PR(D)}{4} + \frac{PR(E)}{1} \qquad \text{(iii)}$$

The above equation can be generalized in the following form assuming that the PageRank incurred by an outband link of a page is the page's own PageRank in the set divided by the number of outband links

$$PR(A) = \frac{PR(C)}{L(C)} + \frac{PR(D)}{L(D)} + \frac{PR(E)}{L(E)} \qquad \text{(iv)}$$

The PageRank of any page *i* can be expressed in the following form:

$$PR(i) = \sum_{j \varepsilon S_i} \frac{PR(j)}{N_j} \qquad \text{(v)}$$

Where,

$PR(i)$ = PageRank of page i

$PR(j)$ = PageRank of any other pages except page i.

N_j = Number of pages in the set

$j \varepsilon S_i$ = Inbound pages linking to page i

PageRank algorithm is mainly used for internet applications to find the rank of a page. The basis of the algorithm is that the rank of a page depends on the inbound links of the other pages. To apply this technique we need to compare links among the pages with the links among the features of the fault. But this study requires the ranking of features according to their importance. This makes PageRank algorithm inappropriate for this project.

5.4 Similarity Matching

This chapter describes the similarity matching techniques for strings. Using these techniques, a concept is proposed to search similar faults when the symptoms of a fault are provided.

5.4.1 Matching Algorithms

Edit distance is a common term in matching algorithms. The word distance is used to compare different data for similarity. Edit distance is a measure to estimate differences between input elements. Different methods to calculate edit distance exist:

Levenshtein Distance

Levenshtein distance is named after the Russian scientist Vladimir Levenshtein, who devised the algorithm in 1965. The Levenshtein distance between two strings is given by the minimum number of operations needed to transform one string into the other, where an operation is an insertion, deletion, or substitution of a single character [49].
Levenshtein distance (LD) is a measure of the similarity between two inputs: the source s and the target input t. The distance is the number of deletions, insertions, or substitutions required to transform s into t. For example,
If s is "math" and t is "math", then LD(s,t) = 0, because no transformations are needed.
If s is "math" and t is "mats", then LD(s,t) = 1, because one substitution (change "h" to "s") is sufficient to transform s into t.
The more different the inputs are, the greater the Levenshtein distance is.

Insertion, deletion and substitution are the main criteria for determining Levenshtein Distance. The position of a character plays an important role to determine the distance. In this study, the description of a fault is dealt with. If Levenshtein Distance is applied to find out the similarity of faults it would not give a meaningful result as the positions of the strings should not have importance. That is why this technique will not be used in this study.

Damerau-Levenshtein Distance

Damerau-Levenshtein distance comes from Levenshtein distance that counts transposition as a single edit operation [49]. The Damerau-Levenshtein distance is equal to the minimal number of insertions, deletions, substitutions and transpositions needed to transform one string into the other.

Kukich [52] described several edit distance algorithms which use Damerau-Levenshtein distance. It has been proved that the use of Damerau-Levenshtein metric to calculate the similarity between two words is a slow process. For this reason this method is not well-suited for similarity matching in this project.

Needleman – Wunsch Distance

The Levenshtein distance algorithm assumes that the cost of all insertions, deletions, substitutions or conversions is equal. However, in some scenarios this may not be desirable and may mask the acceptable distances between inputs.

Needleman-Wunsch has modified Levenshtein distance algorithm to add cost matrix as an extra input. This matrix structure contains two cost matrics for each pair of characters to convert from and to. The cost of inserting this character and converting between characters is listed in this matrix.

This approach is not appropriate for use in this study's similarity matching for the same reason stated in Levenshtein approach.

Hamming Distance

The Hamming distance H is defined for the same length inputs. For two inputs s and t, H(s, t) is the number of places in which the two strings differ, i.e., have different characters.

Hamming Distance is used in information theory. This method can not be applied in similarity matching for automotive faults since Hamming Distance only considers the differences among the two inputs.

Weighted Edit Distance

This algorithm differs from the Edit Distance in weighting. A particular weight is imposed for each operation of insertion, deletion and substitution.

The main goal of similarity matching of faults is to find the faults with the similar behaviors. Weighted Edit Distance focuses on providing weight on the operations. This kind of approach is inappropriate for finding similar faults.

Hamming Distance

The Hamming distance is the number of positions for which the corresponding characters differ. It is simply the number of differences between two strings of the same length.

For example:
The Hamming Distance between GERMANY and IRELAND is 5.

To apply this distance between two error features they must be of equal length, which is a rare case. This results in the decision not to use Hamming Distance for similarity matching in this study.

5.5 Similarity Determination

The aim of this chapter is to propose an algorithm to use for similarity matching in text queries. The procedures of this algorithm are as following

A text (query) T is represented by multidimensional vector:
$F(T) = (F_1(T), F_2(T), \ldots F_k(T))$ (occurrence vector)
k = no. of distinct term occurring in database (non-stop word)

Function of frequency of the *i*-th term in T,

$$F_i(T) = \frac{1}{2}\left(1 + \frac{tf_i^T}{\max tf_i^T}\right)\log\frac{N}{n_i}$$

where, tf_i^T = frequency of the *i-th* term in ***T***

$\max tf_i^T$ = no. of database documents where the most frequent term of T occurs
N = no. of database entries
n_i = no. of entries where the *i-th* term occurs

The cosine similarity measure between a query (*A*) and a stored document (*B*) is defined as:

$$Cos_{similarity}(A,B) = \frac{\sum_{i=1}^{N}\alpha_i F_i(A) F_i(B)}{\sqrt{\sum_{i=1}^{N}\alpha_i^2 F_i(A)^2 \sum_{i=1}^{N}\alpha_i^2 F_i(B)^2}}$$

where,

α_i = user-determined parameter (weights) (~***1***)

Cosine similarity method counts the number of different words in two documents. With this method the highest frequency words within any document will have the largest influence on its similarity with other documents. Documents with many occurrences of an unusual word or many different unusual words will have low cosine similarity measures with most other documents. Weighting schemes are frequently used to modify the standard cosine measure. These typically lower the importance of common words.

Below are the results of some input data and their similarities with the existing input database using this algorithm –

Input Database:

This is the database which is already stored in the system. This is compared with the user provided fault symptoms.

Attachment	**Defect ID**	**Fault Characteristics**
	32	*Display ON Signal will be sent, but Display remains dark*
Y	*40*	*Preconditions: radio hu message*
	41	*radio: radio. message; audio hu radio message message*
	42	*Preconditions: message-> hu -> audio*
Y	*44*	*preconditions: hu sds sdars message message radio radio*
	45	*radio hu message message message*
	46	*radio hu no message message*
	47	*radio no hu message message no sds*
	48	*radio hu message message no message sds*
	49	*radio hu*
	50	*radio*
	51	*radio radio*
	52	*radio radio radio*
	53	radio does not receive message from headunit

Result Analysis:

Below is the graphical representation of outputs for determining fault similarities corresponding to user provided fault symptoms. The similarities of fault symptom *radio hu* are 100% (fault id 49) with the database fault *radio hu* and 68% (fault id 40) with the database fault *Preconditions: radio hu message.* The result of this fault matching is shown in figure 5.3.

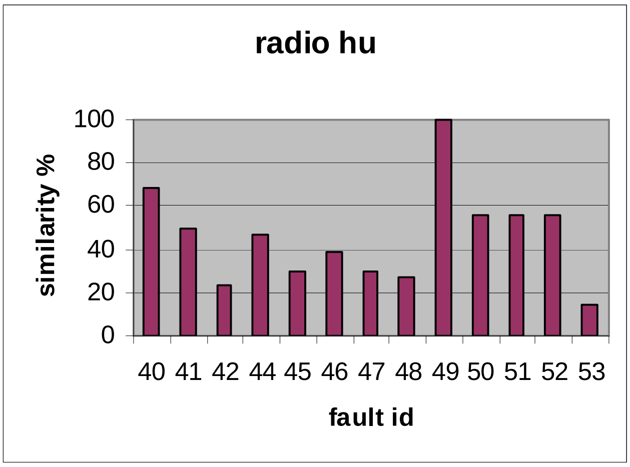


Figure 5.3: Fault similarities with symptom radio hu

The similarities of fault symptom *radio hu message* are 100% (id 40) with database fault *radio hu* and 84% (id 41) with database fault *radio: radio. message; audio hu radio message message* and 84% (id 45) with database fault *radio hu message message messaeg.* The result of this fault matching is shown in figure 5.4.

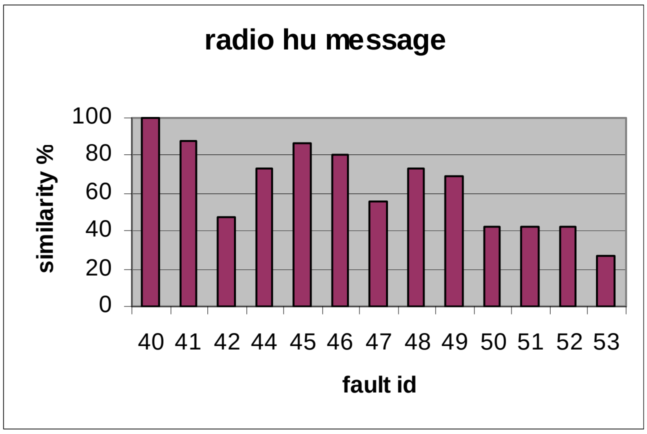


Figure 5.4: Fault similarities with symptom radio hu message

The similarities of fault symptom *radio dvd message* are 61% (id 41) with the database fault *radio: radio. message; audio hu radio message message* and 56% (id 45) with the database fault *radio hu message message message.* The result of this fault matching is shown in figure 5.5.

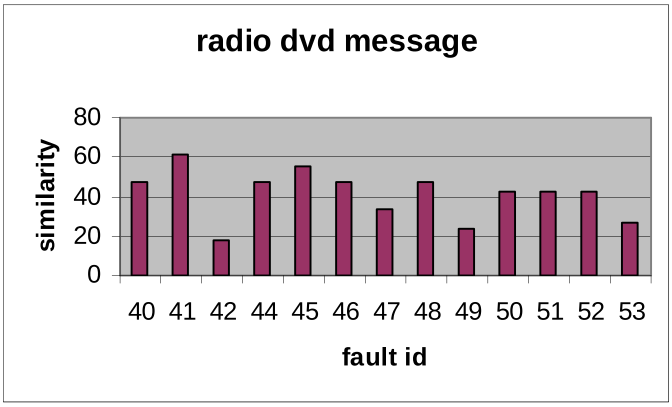


Figure 5.5: Fault similarities with symptom radio dvd message

The similarities of fault symptom *radio dvd* are 58% with database faults *radio* (id 50) and *radio radio*(id 51) and *radio radio radio* (id 52). The result of this fault matching is shown in figure 5.6.

Based on the above result analysis it can be concluded that the similarity of a user provided fault is higher if the symptom of the fault matches more closely with any database fault. It satisfies the requirement of finding similar faults for a fault symptom.

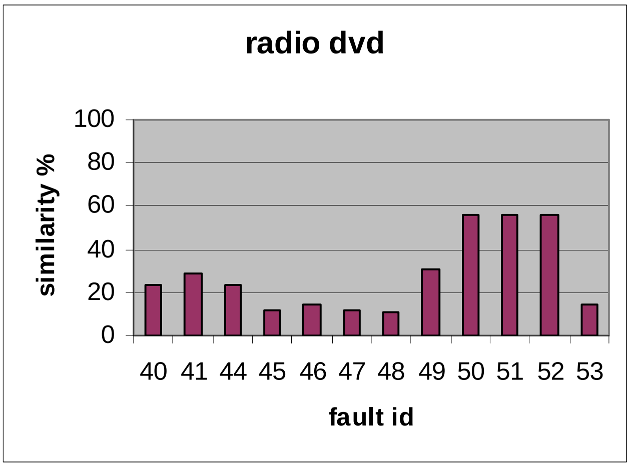


Figure 5.6: Fault similarities with symptom radio dvd

5.6 Fuzzy Logic based Fault Diagnostic System

Fuzzy Logic (FL) is a problem-solving control system methodology [54] which was introduced by Dr. Lotfi Zadeh. It is a generalization of standard logic, where a concept can have a degree of truth from 0.0 to 1.0. Truth degree of 0.0 means the concept is fully false, whereas 1.0 means the concept is fully true. Fuzzy logic makes use of IF X AND Y THEN Z reasoning. Unlike a mathematical model, it relies on the descriptive characters of the system where it is applied. This can be made clear with any example:

For example regarding an infotainment “system” and ”faultiness”. A Fuzzy subset of FAULT that determines “to what degree is a device x faulty” should be designed. A degree of membership in the Fuzzy subset of FAULT must be assigned to each member in infotainment system Set. A membership function based on the sleep current of a device in the system is assumed.

$$FAULT(x) = \{0 \qquad , \text{if } SleepCurrent(x) < 50\mu A,$$

$$= SleepCurrent(x) - 50\mu A/60, \text{ if } 50\mu A <= SleepCurrent(x) <= 110\mu A$$

$$= 1 \qquad , \text{if } SleepCurrent(x) > 110\mu A$$

Now imagine a Rear Seat Entertainment (RSE) device has sleep current of 115µA. It means “RSE is Faulty” = 1. A device TVTuner has sleep current of 80µA means “TVTuner is Faulty” = 0.5.

Zaden describes Fault as LINGUISTIC VARIABLE which represents our cognitive category of Faultiness.

One very important application of Fuzzy membership functions and rules is an expert system for data analysis or data reasoning. To reason on data, the following standard form is used in an expert system

If x is fully-faulty and y is non-faulty then z = faulty
Where x and y are input variables and z is output variable. Fully-faulty is a membership function defined on x which is termed as fuzzy subset; not-faulty is a membership function defined on y which is termed as fuzzy subset and faulty is a membership function defined on z. The conclusion assigns a membership to each of the output variable(s). The set of rules is termed as rule base or knowledge base.

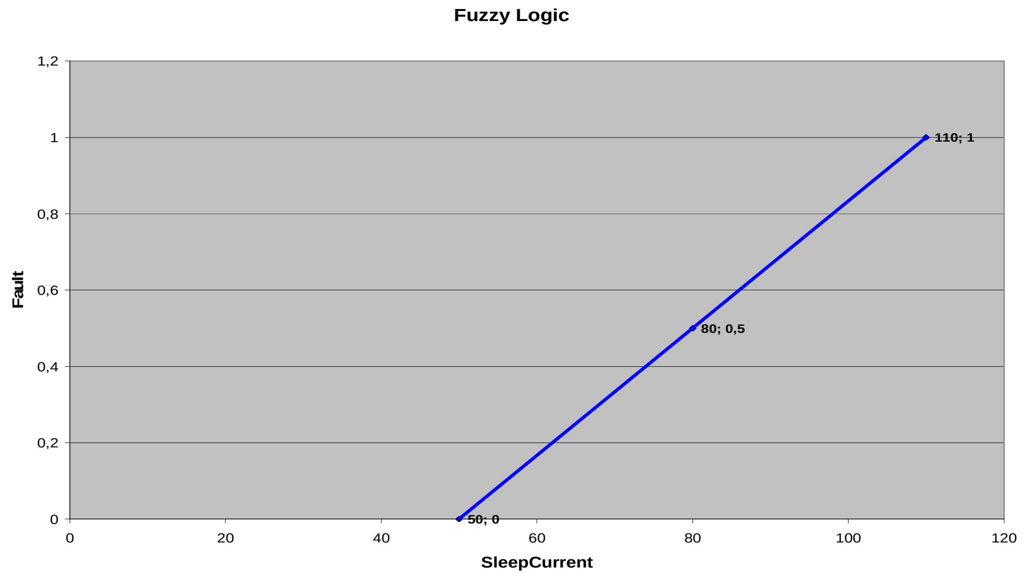


Figure 5.7: Fuzzy Logic for SleepCurrent and Fault for devices

Knowledge or Rule Base: Knowledge base or Rule base is an imprecise form of storage of human experiences. Conditions and Actions in the form of IF …, THEN …, are the main features to describe a human expert's knowledge in a linguistic way.

Fuzzification, Inference, Composition and Defuzzification are the important steps of a Fuzzy Expert system.

Fuzzification: Fuzzification is the process where the membership functions defined on the inputs are applied to their actual values to calculate the degree of truth for each rule premise.

Inference Engine: Uncertainty of information in rule base results in uncertainty in decision. This requires the inference engine to have the capacity of computing to analyze the transmission of uncertainty from the premises to the conclusions. The conclusion should be comprehensible for general users. MIN or PRODUCT is used as inference rule.

Composition: All the Fuzzy subsets assigned to each output variable are combined together to form a single Fuzzy subset for each output variable. MAX or SUM is used for composition. A combined output fuzzy subset is determined by taking the pointwise maximum over all of the fuzzy subsets assigned to variable by the inference rule with the help of MAX. A combined output fuzzy subset is determined by taking the pointwise sum over all of the fuzzy subsets assigned to the output variable by the inference rule with the help of SUM.

Defuzzification: The transformation of fuzzy output set to a crisp number is called Defuzzification. This is the process of computing a scalar from a fuzzy conclusion. This is needed especially for fuzzy control system. Different approaches are available to determine Defuzzification.
A single defuzzified number can be calculated by methods such as average maximum method, weighted average maxima method and a centroid method. In the centroid method the defuzzified value is determined by the center of gravity of the area below the membership function of the grade member. The variable value at which the fuzzy subset has its maximum truth value is taken as the crisp value for the output variable in maximum method. Now the equations for fuzzy systems will be derived:

Rule 1: IF X is A1 and Y is B2 THEN Z is C1
Rule 2: IF X is A2 and Y is B2 THEN Z is C2

The degrees of membership are represented by $\mu A_1(x_0)$ and by $\mu B_1(y_0)$ for Rule1 for fuzzy variables X and Y. x_0 and y_0 represent two specific values on fuzzy variables X and Y.
Similarly, the degrees of membership are represented by $\mu A_2(x_0)$ and by $\mu B_2(y_0)$ for Rule2 for fuzzy variables X and Y.

The degree of truth for Rule 1 is defined with

$$\alpha_1 = min\{\mu_{A1}(x_0), \mu_{B1}(y_0)\}$$

The degree of truth for Rule 2 is defined with

$$\alpha_1 = min\{\mu_{A2}(x_0), \mu_{B2}(y_0)\}$$

The outputs of Rule 1 and Rule 2 are calculated by means of matching the strength of its preconditions on its conclusion:

$$\mu_{C'1}(\omega) = min\{\alpha_1, \mu_{C1}(\omega)\}$$

$$\mu_{C'2}(\omega) = min\{\alpha_1, \mu_{C2}(\omega)\}$$

Here ω is Rule conclusions' value. $\mu_{C'1}(\omega)$ is an output action for Rule 1 and $\mu_{C'2}(\omega)$ is an output action for Rule 2.

Then conflict resolution is determined by the following equation

$$\mu_C(\omega) = max\{\mu_{C'1}(\omega), \mu_{C'2}(\omega)\}$$

$\mu_C(\omega)$ is a pointwise membership function for the combined conclusion of the rules of Rule 1 and Rule 2. The operation max refers to union.

Now the defuzzified crisp number is determined by center of gravity method

$$Z^{*} = \frac{\sum_{i=1}^{k} z_i \mu_C(z_i)}{\sum_{i=1}^{k} \mu_C(z_i)}$$

Where k is the number of quantization levels of the output, Z_i is the amount of system output at the quantization level i and $\mu_C(z_i)$ represents its membership value in C.

5.7 Fuzzy System for Automotive Infotainment System

A fuzzy system accepts the input variables, processes them and then sends an output. Below is a schematic diagram of fuzzy system for automotive infotainment system

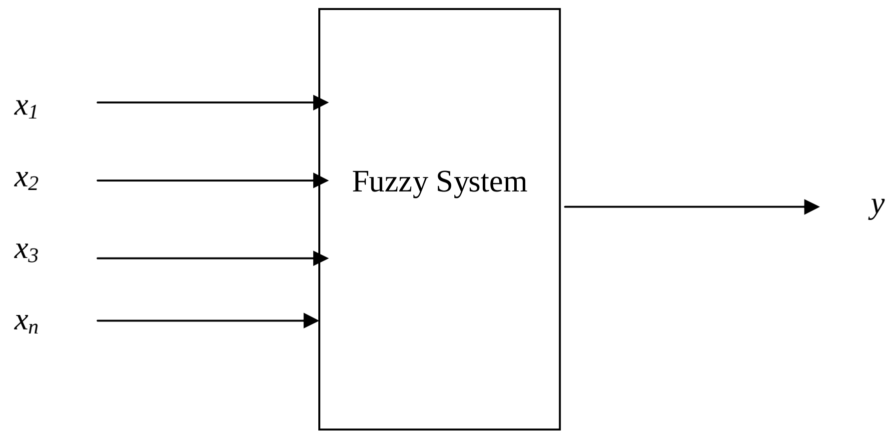


Figure 5.8: Fuzzy System

Here x_1, x_2, x_3 *and* x_n are input variables and y is output.

A more detailed fuzzy system is presented below.

The main components of the fuzzy system are Fuzzification block, Inference block and Defuzzification Block. Membership Function and Rule Base are two important intermediate processes that provide inputs for Fuzzification, Inference and Defuzzification blocks.

Now the defuzzified number for three infotainment devices can be found out:

Assume three infotainment devices as variables, TV, Radio and RSE all have a volume in the range 0 to 10. The following membership functions and rules are proposed –

$$Faulty(t) = 1 - (t/10)$$
$$NonFaulty(t) = t/10$$

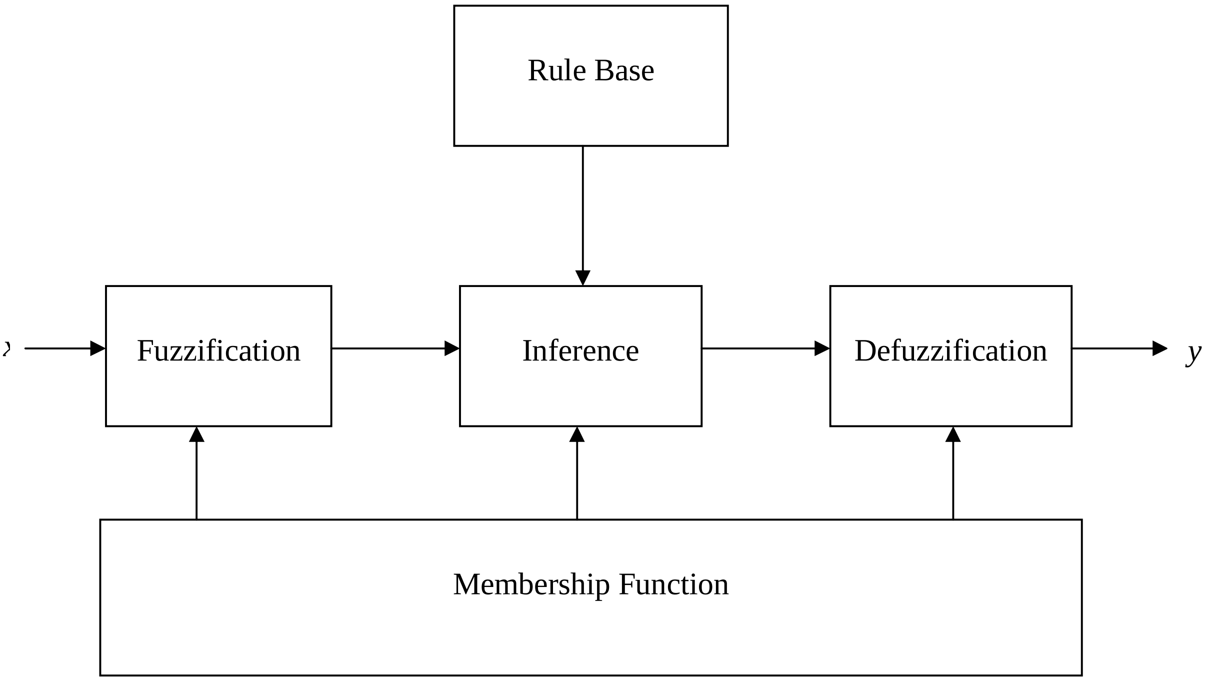


Figure 5.9: Fuzzy System Components

Rule 1: if TV is Faulty and Radio is Faulty then RSE is NonFaulty
Rule 2: if TV is Faulty and Radio is NonFaulty then RSE is Faulty
Rule 3: if TV is NonFaulty and Radio is Faulty then RSE is Faulty
Rule 4: if TV is NonFaulty and Radio is NonFaulty then RSE is NonFaulty

Now investigate Rule 1 for TV = 0.0 and Radio = 3.2. It means

$$Fauly1(TV) = 1.0$$
$$NonFaulty(TV) = 0.0$$
$$Faulty(Radio) = 0.68$$
$$NonFaulty(Radio) = 0.32$$

Now the degree of truth for a rule's premise according to MIN inferencing for different rules -

Rule 1:

$$\alpha_1(RSE) = min\{Faulty(TV), Faulty(Radio)\} = min\{1.0, 0.68\} = 0.68$$

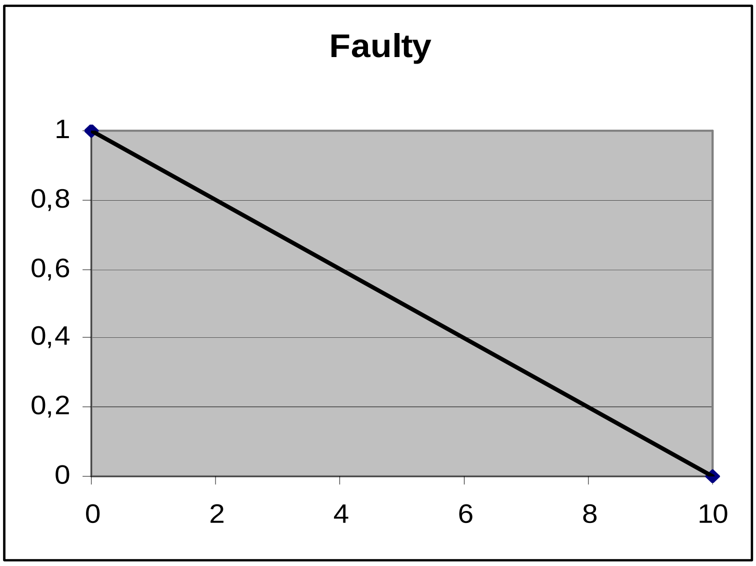


Figure 5.10: Fuzzy rule for Faulty

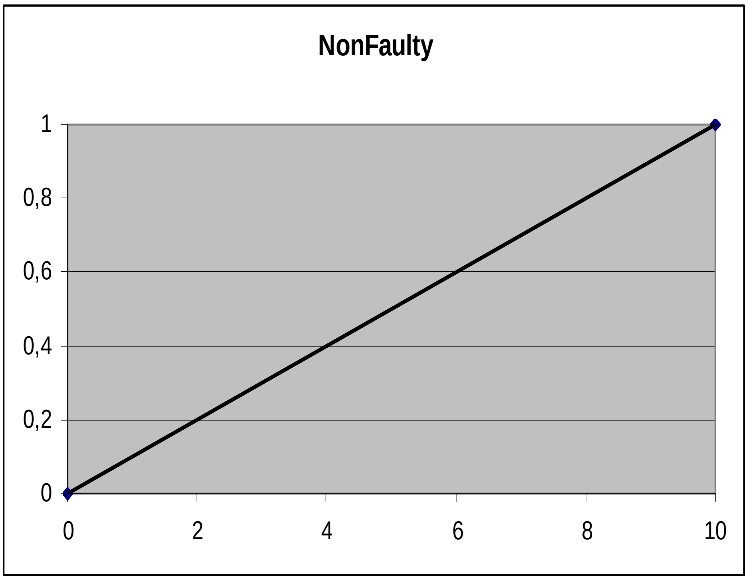


Figure 5.11: Fuzzy Rule for NonFaulty

Rule 2:

$$\alpha_2(RSE) = min\{Faulty(TV), NonFaulty(Radio)\} = min\{1.0, 0.32\} = 0.32$$

Rule 3:

$$\alpha_3\ (RSE) = min\{NonFaulty(TV), Faulty(Radio)\} = min\{0.0, 0.0.68\} = 0.0$$

Rule 4:

$$\alpha_4(RSE) = min\{NonFaulty(TV), NonFaulty(Radio)\} = min\{0.0, 0.0.32\} = 0.0$$

The above degrees of truth for a rule's premise are shown on the following figure:

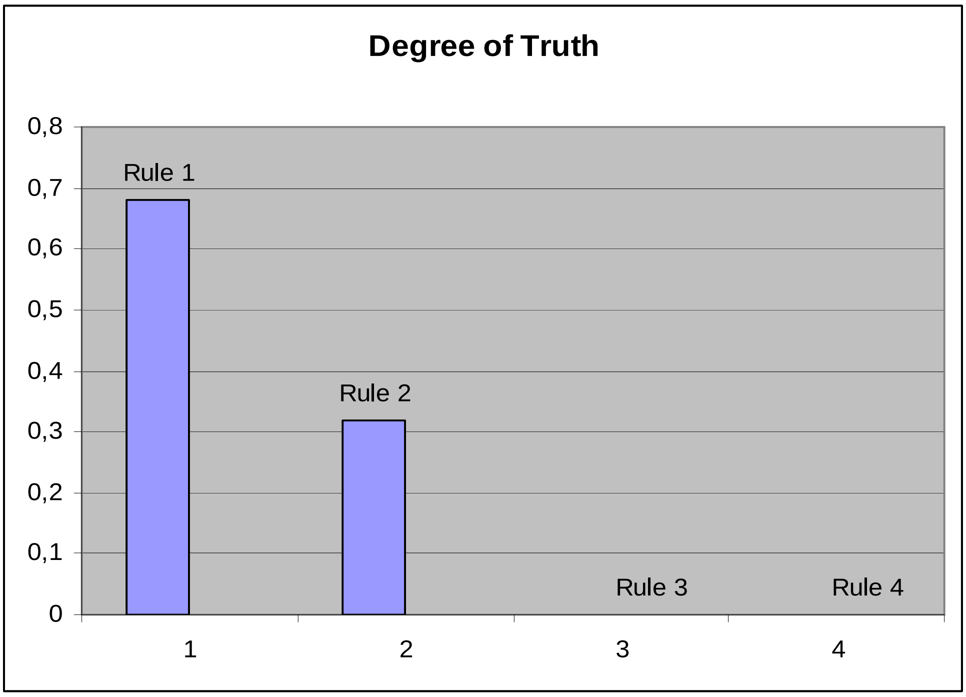


Figure 5.12: Degree of Truth for different rules

It means for Rule 1, MIN inferencing gives RSE the fuzzy subset defined by the membership function:

$$Rule1(RSE) = \{RSE/10, \text{ when } RSE <= 6.8$$

$$0.68, \quad \text{when } z >= 6.8 \}$$

PRODUCT inferencing gives RSE the fuzzy subset defined by the membership function:

$$Rule1(RSE) = \alpha_{1(RSE)} * NonFaulty(RSE)$$

$$= 0.68 * RSE/10$$

$$= 0.068 * RSE$$

Similarly,

$$Rule2(RSE) = \alpha_{2(RSE)} * Faulty(RSE)$$

= 0.32(1 – RSE/10)*

*=0.32 – 0.032*RSE*

Rule3(RSE) = $\alpha_{3(RSE)}$ **Faulty(RSE)*

*= 0.0 *(1 – RSE/10)*

=0.0

Rule4(RSE) = $\alpha_{3(RSE)}$ **NonFaulty(RSE)*

= 0.0(RSE/10)*

=0.0

SUM composition (pointwise sum over all of the fuzzy subsets assigned to the output variable by the inference rule) is

Fuzzy(RSE) = Rule1(RSE) + Rule2(RSE) + Rule3(RSE) + Rule4(RSE)

*= 0.068*RSE + 0.32 – 0.032*RSE + 0.0 + 0.0*

*= 0.32 + 0.036*RSE*

The crisp value is computed in the centroid method by calculating the centre of gravity of the membership function for the fuzzy value.

The area of the function over 0 to 10

$$\begin{aligned} A &= \int_0^{10} \{(0.32 + 0.036 \times RSE)\,dRSE\} \\ &= \{(0.32 \times RSE + 0.018 \times RSE^2)\big|_0^{10}\} \\ &= 5.0 \end{aligned}$$

The moment of the function over 0 to 10

$$\begin{aligned} M &= \int_0^{10} \{0.32 \times RSE + 0.018 \times RSE^2)\,dRSE\} \\ &= \{(0.18 \times RSE^2 + 0.012 \times RSE^3)\big|_0^{10}\} \\ &= 28.0 \end{aligned}$$

The centroid or crisp value is calculated as following

$$CrispValue(RSE) = \frac{M}{A}$$

$$= 5{:}6$$

Now consider the above fuzzy system. Assume:
n = number of inputs
m = number of membership functions
R = Maximum number of rules

$R = m^n$

If the number of input is 10 and the number of membership function is 5, then the maximum number of rules is 5^{10} which is 9,765,625. This incurs a huge effort in calculating the crisp value. This is why the rationale should be to keep the number of inputs and number of membership functions as low as possible. Restricted use of this fuzzy system will be employed for this study's infotainment system.

Many authors have proved that the crisp value of fuzzy system is useful for feedback in a control system. The input is modified according to the feedback. Thus crisp value is meaningful for fuzzy control system. Our aim is to visualize a fault picture based on existing expert knowledge. Thus it is easy to propose solutions to fix a fault. It is redundant for our work to use Defuzzification block in Fuzzy system. Inferencing is used to find out Defuzzification value. Inferencing block can also be eliminated from the system to be developed. The modified fuzzy system turns into the following form –

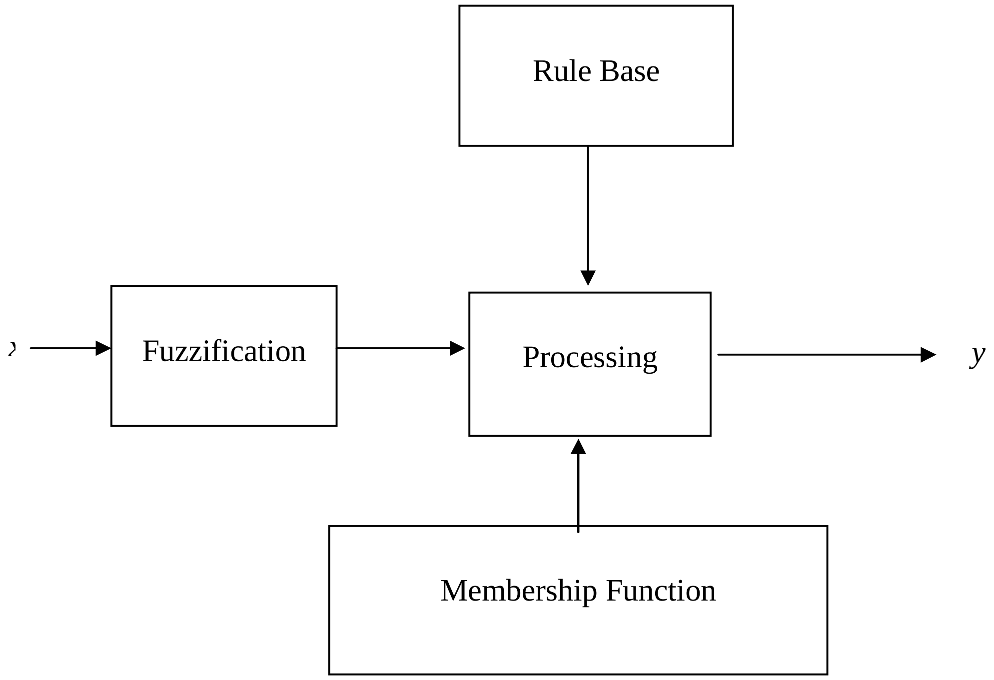


Figure 5.13: Fuzzy System for Automotive Infotainment

Chapter 6

System Design and System Model

In this chapter the system model is designed. Detail descriptions of different phases, methodologies used and techniques of the model are also outlined.

6.1 System Model

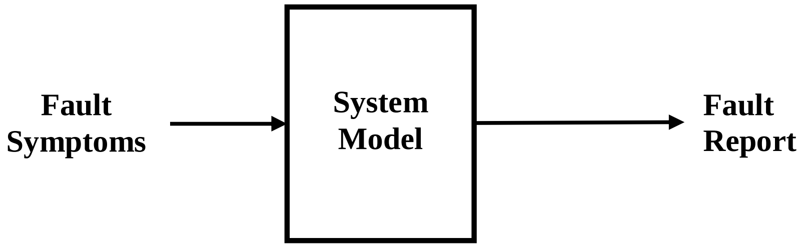


Figure 6.1: System Model

The system receives fault symptoms as input. It processes the symptoms. Then a fault report containing expected features is generated. The input into the system is the fault symptoms and the output is the fault report. The fault report prints the nature of the fault and probable solutions. Now the question is which requirements of the system should be developed.

6.2 System Design

In this section this system is described - the assumptions, system parameters, inputs, outputs.

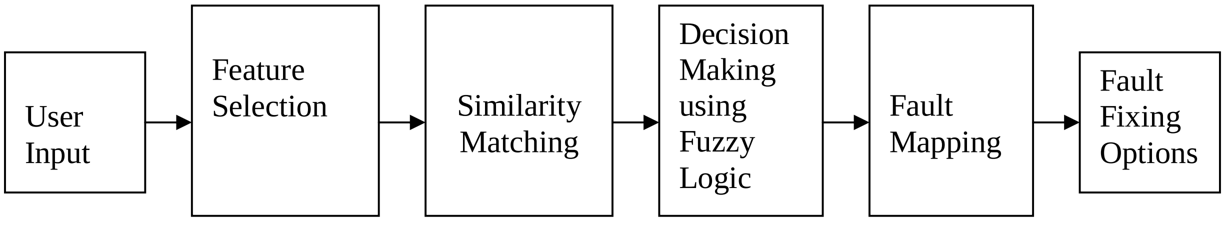


Figure 6.2: System Model for Fault Diagnosis

The new system model in this study consists of feature selection, similarity matching, decision making and fault mapping blocks. User provided fault symptoms are processed in the above blocks. The output of the system is fault fixing suggestions. A set of options will be generated to fix the fault. The figure 6.2 depicts the schematic view of the ISFAD.

Each of the blocks of ISFAD processes user provided fault characteristics. The functionalities, techniques and methodologies are described in the following parts.

A user at first describes the fault symptoms with natural language. The symptoms may contain meaningless words for determining fault matches. Meaningless words and features are filtered out from user provided fault characteristics. In this phase stop wording is used to eliminate the unnecessary words. A database of stop words for automotive multimedia and infotainment system has already been developed. If the fault symptom contains any stop word, the stop word is eliminated from further process. Then the word stemming technique is applied to treat words with the same root as the same feature. The output of word stemming is a set of selected features. Thus a set of features with high information content is extracted from user input. Figure 6.3 describes the feature selection phase.

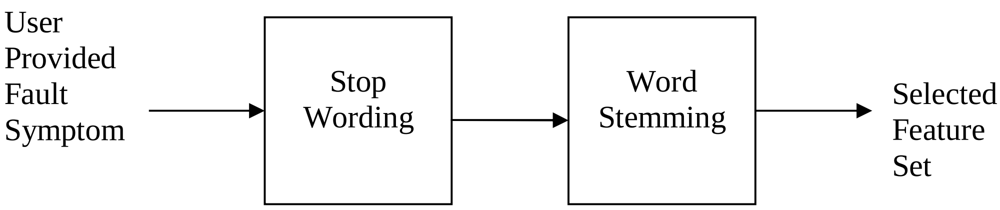


Figure 6.3: Feature Selection Phase

In Appendix 1, a list of stop words has been built up, which are closely related to multimedia and infotainment system in automotive. Appendix 2 lists stemming words.

Selected features are then processed in the similarity matching phase. Cosine algorithm is used to find the similar faults which are already stored in the system. A measure of similarity is also generated with the matched faults. The similarity scale is chosen from 0% to 100%. The outputs of this phase are similar faults and degree of similarity. Figure 6.4 depicts the basic steps of similarity matching phase.

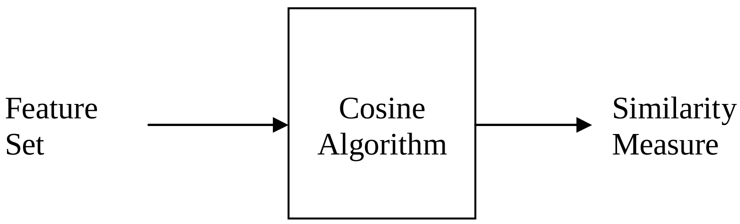


Figure 6.4: Similarity Matching Phase

The output of the similarity matching phase is provided to the Fuzzy Logic based decision making system. Fuzzy rules are built to classify the faults in three fuzzy sets of Low, Medium and High. Membership functions for the three fuzzy sets are derived mathematically. Then the grade of membership of these sets is determined based on the similarity measure. The grade of membership is the decision on fault proximity with the stored faults. This process is described in Figure 6.5.

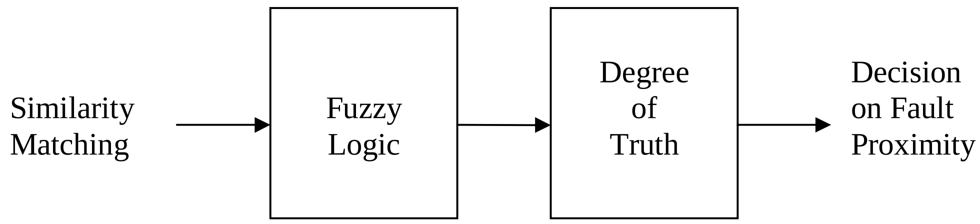


Figure 6.5: Fuzzy System

Based on the grade of membership for the fuzzy sets of low, medium and high, solution possibilities are generated. To find out the fixing possibilities, the existing fault causes corresponding to the faults stored in database are used. For every possible fault solution, a degree of confidence is provided based on fuzzy rules. These processes are depicted in figure 6.6.

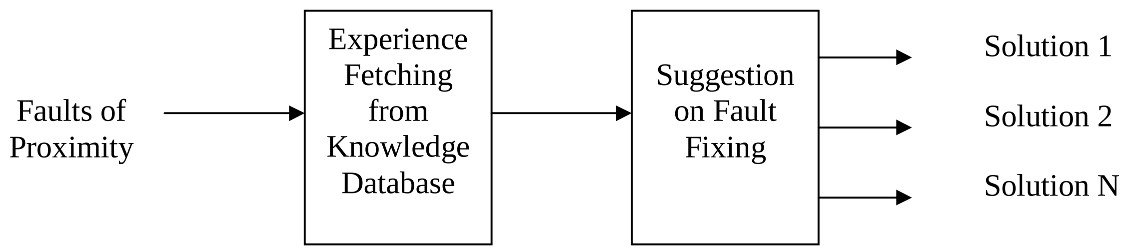


Figure 6.6: Fault Mapping and Probable Solution Phase

6.3 Design Parameters

From the above system design, it is understood that for taking decisions on fault proximity and degree of confidence for fixing suggestions it is necessary to build a mathematical model.

From the similarity measure phase, some numerical values are obtained. This value is converted into a percentage. It means the output of the phase is a numerical value from zero to 100. Based on this similarity value, a mathematical model for fuzzy system has to be developed.

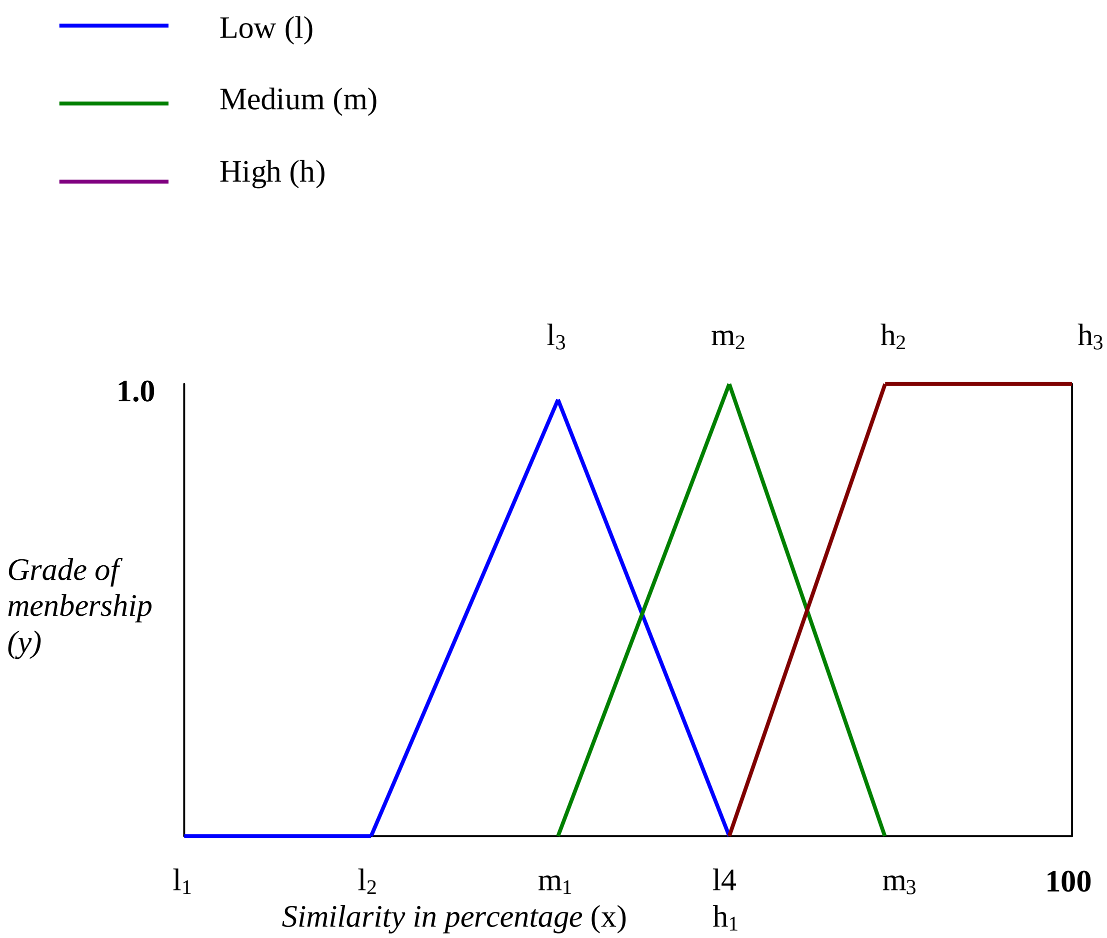


Figure 6.7: Membership function of fuzzy sets

Now equations for fuzzy sets are constructed. Fuzzy set low is represented by the following equations:

$$y = 0 \text{ for } x = 0 \text{ to } l_2 \ldots (i)$$

$$y = \frac{1}{l_3 - l_2} x - \frac{l_2}{l_3 - l_2} \text{ for } x = l_2 \text{ to } l_3 \ldots (ii)$$

$$y = \frac{l_3}{l_4 - l_3} \times x + 1 \text{ for } x = l_3 \text{ to } l_4 \ldots (iii)$$

For all the other values of *l*, $y = 0$.

Fuzzy set medium is represented by the following equations

$$y = \frac{1}{m_2 - m_1} x - \frac{m_1}{m_2 - m_1} \text{ for } x = m_1 \text{ to } m_2 \ldots (iv)$$

$$y = \frac{1}{m_2 \text{à} m_3} x \text{ à } \frac{m_3}{m_2 \text{à} m_3} \quad \text{for} \quad x = m_2 \text{ to } m_3 ::: (v)$$

For all the other values of *m, y = 0.*

Fuzzy set high is represented by the following equations

$$y = \frac{1}{h_2 \text{à} h_1} x \text{ à } \frac{h_1}{h_2 \text{à} h_1} \quad \text{for} \quad x = h_1 \text{ to } h_2 ::: (vi)$$

$$y = 1 \text{ for } x = h_2 \text{ to } h_3 ::: (vii)$$

For all the other values of *h, y = 0.*

According to the above equations, it can be said that l_2, l_3, l_4, m_1, m_2, m_3, h_1, h_2, h_3 are design parameters. Decisions on fault solutions are dependent on these parameters.

6.4 Assumptions

Consider that the peak point of fuzzy set low corresponds to the start point of fuzzy set medium and the peak point of fuzzy set medium corresponds to the start point of fuzzy set high. The start point of high and the end point of low are the same point. The following relations can be established -

$$l_3 = m_1 ::: (viii)$$

$$l_4 = m_2 = h_1 ::: (ix)$$

$$m_3 = h_2 ::: (x)$$

The two sides of low triangle and two sides of mid triangle are kept equal. It means the distances from points (l_2, 0) to (l_3, 1) and from points (l_3, 1) to (l_4, 0) are equal. Similarly, the distances from points (m_1, 0) to (m_2, 1) and from points (m_2, 1) to (m_3, 0) are equal.

To further simplify the simulations, the homogeneous distribution of low, medium and high membership functions are assumed. As a result, the following relations are established -

$$m_1 \text{ à } l_2 = h_1 \text{ à } m_1 = m_3 \text{ à } h_2 ::: (xi)$$

The following relations also hold valid for this system -

The degree of membership of low and medium coincide (cross-pint) at mid value, which is 0.5.
The degree of membership of medium and high coincide (cross-point) at mid value, which is 0.5

From the figure of membership function of fuzzy set, it is obvious that the degrees of membership of all the fuzzy sets are within the range of 0 and 1 inclusive.

6.5 Level of Confidence

Based on the degree of membership for fuzzy sets, the level of confidence for the suggestion of how to fix a fault is determined. The confidence level (CL) is classified in six categories – CL0, CL1, CL2, CL3, CL4 and CL5. CL0 indicates the lowest level of confidence (no suggestions available) and CF5 indicates the highest level of confidence on fault fixing suggestions.

Confidence levels CL0, CL1, CL2, CL3, CL4 and CL5 are defined with the following conditions:

A fault fixing suggestion is said to be in class CL0 when the following criteria are met:

Degree of Membership for Low is less than 1
And the Degree of Membership for Medium 0
And the Degree of Membership for High is 0

A fault fixing suggestion is said to be in class CL1 when the following criteria are met

Degree of Membership for Low is less than 1 and greater than 0.5
And the Degree of Membership for Medium is greater than 0 and less than 0.5
And the Degree of Membership for High is 0

A fault fixing suggestion is said to be in class CF2 when the following criteria are met

Degree of Membership for Low is less than or equal to 0.5
And the Degree of Membership for Medium is greater than or equal to 0.5
And the Degree of Membership for High is 0

A fault fixing suggestion is said to be in class CL3 when the following criteria are met

Degree of Membership for Low 0
And the Degree of Membership for Medium is greater than or equal to 0.5 and less than 1
And the Degree of Membership for High is greater than 0 and less than or equal to 0.5

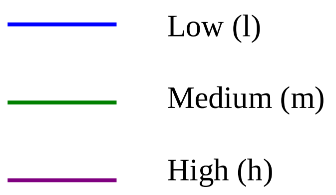
A fault fixing suggestion is said to be in class CL4 when the following criteria are met

Degree of Membership for Low is 0
And the Degree of Membership for Medium is less than 0.5
And the Degree of Membership for High is greater than 0.5 and less than 1.0

A fault fixing suggestion is said to be in class CL5 when the following criteria are met

Degree of Membership for Low is equal to 0
And the Degree of Membership for Medium is 0
And the Degree of Membership for High is equal to 1

The following figure clarifies the concepts discussed above -



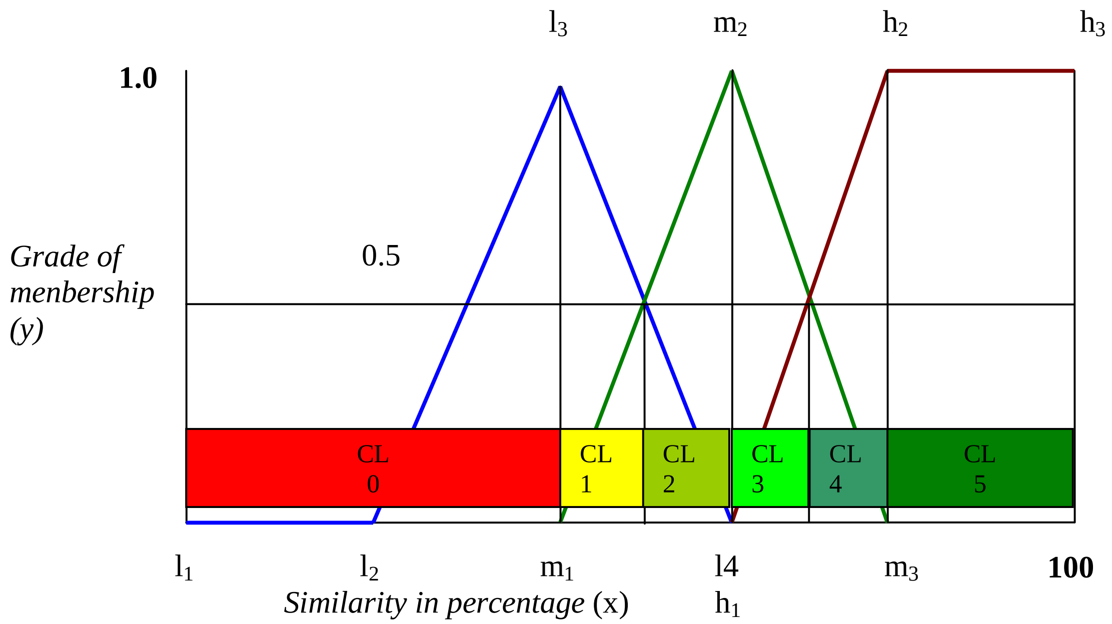


Figure 6.8: Classification on level of confidence

6.6 Tuning Parameters

In this system it is possible to choose all of the parameters of l_1, l_2, l_3, l_4, m_1, m_2, m_3, h_1, h_2, h_3 as tune parameters for simulations using the system. Based on the above assumptions, l_2 and h_2 are chosen as tuning parameters.

Chapter 7

Simulation and Result

This chapter describes the simulation setup, simulation input, processing methodologies and the output. The result is assessed and a decision is made about the performance and novelty of this study's system.

Firstly $l_2 = 20$ and $h_2 = 80$ are selected. This is simulation set 1. In simulation set 2, $l_2 = 10$ and $h_2 = 0$ are chosen. In simulation set 3, $l_2 = 0$ and $h_2 = 70$ are chosen.

7.1 Fault Database

A fault database based on a real infotainment system in a real car has been created. The faults were recorded during normal driving of a car. The infotainment system consists of Head Unit (HU), TVTuner, DABTuner (Digital Radio), Rear Seat Entertainment (RSE) and an Amplifier. HU is the main controller. The other devices are slaves. RSE consists of a CD/DVD player, Auxiliary input and a Head Phone Amplifier. To monitor the real communications among the devices an Optolyzer was used as a spy. The bus traffic indicates the commands and responses for each device. The log is used to find out the real cause of the fault.

The faults were recorded according to Fault ID. Then the description of the fault was written with as many keywords as possible. Then, after analysis, the log causes of the faults were noted. This primary database will be used as the base of this system.

FaultID	Fault Description	Fault cause
1	GroupAddress is not stored after light off in Radio	Radio NetworkManagement
2	Light not switched on within wake up time at RSE	RSE NetworkManagement
3	RSE sleep current out of range	RSE Hardware
4	Device information can not be read from DABTuner	DABTuner NetBlock
5	Critical unlock recognition out of range for TVTuner	TVTuner NetworkManagement
6	MOST: Light not switched off at HU after shutdown	HU NetworkManagement
7	Medium-Insert Event does not work at RSE	RSE MultiMediaPlayer Deckstatus

8	Displays are not switched off after shutdown at RSE	HU NetworkManagement
9	MOST: TVTuner does not switch off in high Temperature	TVTuner NetworkManagement
10	Wrong HighLevel Retry for play at HU	HU NetworkManagement
11	RSE: Volume of HeadPhoneAmplifier can not be changed	RSE HeadPhoneAmplifier MixerLevelExt/Voulme
12	MOST: Automatic update of FunctionBlocks in system state Config NotOK from RSE	RSE NetworkManagement
13	MOST: RSE does not react for any command such as play, stop etc	RSE MultiMediaPlayer NeworkManagement/Deckstatus
14	MultiMediaPlayer makes internal Reset after Notification.Set.All command from the HU	RSE MultiMediaPlayer Notification Function
15	RSE makes light off during soft reset	RSE Function Block Diagnosis/NetworkManagement
16	HeadPhoneAmplifier does not mute Audio after an MPR-	RSE HeadPhoneAmplifier Allocate/ResourceManagement
17	RSE does not update its display when DVD is inserted or ejected during play	RSE MultiMediaPlayer DeckStatus Load/Unload
18	HU does not enter error in memory for under voltage	HU Function Block Diagnosis
19	Color of TVTuner can not be changed	TVTuner Function Color
20	MOST: The display does not show tuned program name for TVTuner	TVTuner Function ProgramName
21	MOST: Headphoneamplifier does not notify changed mute status	HeadPhoneAmplifier Function Mute
22	TVTuner displays wrong country code	FBlock TVTuner Function CountryCode
23	Country name is showed wrongly in display	TVTuner Function CountryName
24	MultiMediaPlayer does not show any sub title	RSE MultiMediaPlayer Function SubTitle OnOff
25	RSE does not show any file position from currently playing video	RSE MultiMediaPlayer Function FilePosition
26	Audio goes to backward by Forward button for Audio CD	RSE MultiMediaPlayer Function FastFwSpeed

27	Movies are not repeated in RSE for DVD	RSE MultiMediaPlayer Function Repeat
28	Time position of MultiMediaPlayer stays fixed always	RSE MultiMediaPlayer Function CurrentTimePosition
29	Head phone audio is not muted by mute	RSE HeadPhoneAmplifier Function Mute
30	TVTuner does not mute volume after shut down	HU Function DeAllocate
31	No audio hearable from TVTuner	TVTuner Function Allocate
32	No program list after AutoStore in TVTuner	TVTuner Background Scan
33	Currently available stations are not shown for DABTuner	DABTuner Function Tune
34	TVTuner can not tune to the specified channel	TVTuner Antenna Disconnect
35	Current channel number is not changed for TVTuner	TVTuner Function CurrentChannel
36	Media volume goes down for volume up command in RSE	HeadPhoneAmplifier Function MixerLevelExt with Operation Type Increment
37	Loading status is not shown during CD insert in RSE	RSE MultiMediaPlayer Function LoadingStatus
38	No directory path for currently playing audio is shown for RSE	RSE MultiMediaPlayer Function CurrentDirectoryPath
39	Track position is not displayed at RSE for CD	RSE MultiMediaPlayer Function TrackPosition
40	Display video format does not change for RSE	RSE MultiMediaPlayer Function VideoFormat OperationType SetGet
41	DABTuner does not play any service during snow fall	DABTuner NetworkManegement Temperature
42	TVTuner hangs if the Auto key is inserted and removed within 5 seconds	HU Unstable Light Lock
43	RSE plays fast forward for the command of pause	RSE MultiMediaPlayer Function Deckstatus
44	Movie is not stopped by stop command in RSE	RSE MultiMediaPlayer Function DeckStatus
45	CD does not start auto play	RSE MultiMediaPlayer DeckStatus

46	TVTuner makes high sound	Amplifier Function Mute
47	No audio is playing if there exists an input at AuxiliaryInput at RSE	HU Function Allocation/Deallocation
48	Infotainment system makes CD sound on/off when the battery supply gets lower	RSE UnverVoltage Hysterisis
49	CD does not get out by eject button	RSE Hardware
50	Display does not show program list of DABTuner	HU HMI Function
51	Display remains dark. Error at the Display	Display
52	HU does not send message to radio	HU HMI Function
53	HU MMI sends wrong message to audio	HU HMI Function
54	TVTuner has no sound	Amplifier MixerLevelExt
55	RSE is not muted during incoming telephone call	HU ResourceManagemnt

7.2 Simulation Results

In this chapter the simulation results are presented. Under Fault symptom field the fault characteristics provided by a user are described. Under field Database Fault ID the ID number of the fault which is stored in our knowledge database is output. The Similarity field indicates the percentage matching of the user provided fault with the knowledge database. The degree of Membership field indicates the Fuzzy degree of membership for Fuzzy sets Low (L), Medium (M) and High (H). Under field CL (Confidence Level), the level of confidence for probable solutions is mentioned. The Solution field indicates the probable solutions for fixing a fault.

Simulation set 1:

$l_2 = 20$ and $h_2 = 80$

Nr	Fault Symptom	Data base Fault ID	Simil arity (%)	Degree of Membership	CL	Solution
1	DVD auto play does not work	27 17 45	23 27 40	L(0.15), M(0.00), H(0.00) L(0.35), M(0.00), H(0.00) L(1.00), M(0.00), H(0.00)	0 0 0	No suggestion No suggestion No suggestion
2	stop	44	22	L(0.10), M(0.00), H(0.00)	0	No suggestion

	button in RSE makes play	43 13	24 35	L(0.20), M(0.00), H(0.00) L(0.75), M(0.00), H(0.00)	0 0	No suggestion No suggestion
3	TVTuner has no sound	31 46 54	38 47 100	L(0.90), M(0.00), H(0.00) L(0.65), M(0.35), H(0.00) L(0.00), M(0.00), H(1.00)	0 1 5	No suggestion Amplifier Function Mute Amplifier MixerLevelExt
4	HU does not switch off light after shutdown	08 09 06	28 28 54	L(0.40), M(0.00), H(0.00) L(0.40), M(0.00), H(0.00) L(0.30), M(0.70), H(0.00)	0 0 2	No suggestion No suggestion HU NetworkManagement
5	TVTuner sleep current too high	35 03 46	24 27 38	L(0.20), M(0.00), H(0.00) L(0.35), M(0.00), H(0.00) L(0.90), M(0.00), H(0.00)	0 0 0	No suggestion No suggestion No suggestion
6	Displays remain switched on after shutdown in RSE	51 02 08	22 29 49	L(0.10), M(0.00), H(0.00) L(0.45), M(0.00), H(0.00) L(0.55), M(0.45), H(0.00)	0 0 2	No suggestion No suggestion HU NetworkManagement
7	HU HighLeve l retry is too many for play command	43 10	20 64	L(0.00), M(0.00), H(0.00) L(0.00), M(0.80), H(0.20)	0 3	No suggestion HU NetworkManagement
8	HeadPhon eAmplifie r volume can not be increased	29 30 11	24 24 48	L(0.20), M(0.00), H(0.00) L(0.20), M(0.00), H(0.00) L(0.60), M(0.40), H(0.00)	0 0 1	No suggestion No suggestion RSE HeadPhoneAmplifier MixerLevelExt/Voulme
9	HU sends Notificati on.SetAll command but the	14	20	L(0.00), M(0.00), H(0.00)	0	No suggestion

	RSE hangs					
10	RSE turns light off for hard reset	15	49	L(0.55), M(0.45), H(0.00)	1	RSE Function Block Diagnosis/NetworkManagement
11	TVTuner color function does not work	07 34 19	24 27 57	L(0.20), M(0.00), H(0.00) L(0.35), M(0.00), H(0.00) L(0.15), M(0.85), H(0.00)	0 0 2	No suggestion No suggestion TVTuner Function Color
12	MultiMediaPlayer subtitle is wrong	24	48	L(0.60), M(0.40), H(0.00)	1	RSE MultiMediaPlayer Function SubTitleOnOff
13	Wrong time position of MultiMediaPlayer for CD	39 28	21 35	L(0.05), M(0.00), H(0.00) L(0.75), M(0.00), H(0.00)	0 0	No suggestion No suggestion
14	TVTuner audio is always muted	26 29 31	22 29 38	L(0.10), M(0.00), H(0.00) L(0.45), M(0.00), H(0.00) L(0.90), M(0.00), H(0.00)	0 0 0	No suggestion No suggestion No suggestion
15	Country code of TVTuner is not set according to the HU command	52 19 22	21 24 35	L(0.05), M(0.00), H(0.00) L(0.20), M(0.00), H(0.00) L(0.75), M(0.00), H(0.00	0 0 0	No suggestion No suggestion No suggestion
16	No update of Function Blocks in system	12	52	L(0.40), M(0.60), H(0.00)	2	RSE NetworkManagement

	state Config Ok for RSE					
17	Program name in display is wrong for TVTuner	20 22 23	32 40 40	L(0.60), M(0.00), H(0.00) L(1.00), M(0.00), H(0.00) L(1.00), M(0.00), H(0.00)	0 0 0	No suggestion No suggestion No suggestion
18	Volume of HeadPhoneAmplifier of RSE does not change according to the command	40 36 11	31 36 55	L(0.55), M(0.00), H(0.00) L(0.80), M(0.00), H(0.00) L(0.25), M(0.75), H(0.00)	0 0 2	No suggestion No suggestion RSE HeadPhoneAmplifier MixerLevelExt/Voulme
19	Video format of RSE is wrong	25 40	22 42	L(0.10), M(0.00), H(0.00) L(0.90), M(0.10), H(0.00)	0 1	No suggestion RSE MultiMediaPlayer Function VideoFormat OperationType SetGet
20	CD plays backward for pause command in RSE	45 43	21 47	L(0.05), M(0.00), H(0.00) L(0.65), M(0.35), H(0.00)	0 2	No suggestion RSE MultiMediaPlayer Function Deckstatus
21	No auto play for CD in RSE	47 45	21 83	L(0.05), M(0.00), H(0.00) L(0.00), M(0.00), H(1.00)	0 1	No suggestion RSE MultiMediaPlayer DeckStatus
22	Eject command from RSE does not out the CD	39 49	31 60	L(0.55), M(0.00), H(0.00) L(0.00), M(1.00), H(0.00)	0 2	No suggestion RSE Hardware

23	There is no update of TunedProgram if antenna is disconnected for TVTuner	31	24	L(0.20), M(0.00), H(0.00)	0	No suggestion
24	Music is not paused for pause command	43	34	L(0.70), M(0.00), H(0.00)	0	No suggestion
25	DABTuner does not play any service if the car speed is too high	41	35	L(0.75), M(0.00), H(0.00)	0	No suggestion
26	Directory path of Audio CD is not displayed	39	31	L(0.55), M(0.00), H(0.00)	0	No suggestion
27	currently available service of DDABTuner is not displayed	38	25	L(0.25), M(0.00), H(0.00)	0	No suggestion
28	change of MPR in MOST does not mute the	29 16	27 40	L(0.35), M(0.00), H(0.00) L(1.00), M(0.00), H(0.00)	0 0	No suggestion No suggestion

	audio					
29	HeadPhoneAmplifier mute does not work	21 16 29	38 42 42	L(0.90), M(0.00), H(0.00) L(0.90), M(0.10), H(0.00) L(0.90), M(0.10), H(0.00)	0 1 1	No suggestion RSE HeadPhoneAmplifier Allocate/ResourceManagement RSE HeadPhoneAmplifier Function Mute
30	Light off error is not entered by HU in error memory	18	63	L(0.00), M(0.85), H(0.15)	3	HU FBlock Diagnosis

Simulation set 2:

$l_2 = 10$ and $h_2 = 70$

Nr	Fault Symptom	Data base Fault ID	Similarity (%)	Degree of Membership	CL	Solutions
1	DVD auto play does not work	27 17 45	23 27 40	L(0.65), M(0.00), H(0.00) L(0.85), M(0.00), H(0.00) L(0.50), M(0.50), H(0.00)	0 0 1	No suggestion No suggestion RSE MultiMediaPlayer DeckStatus
2	stop button in RSE makes play	44 43 13	22 24 35	L(0.60), M(0.00), H(0.00) L(0.70), M(0.00), H(0.00) L(0.75), M(0.25), H(0.00)	0 0 1	No suggestion No suggestion RSE MultiMediaPlayer NeworkManagement/Deckstatus
3	TVTuner has no sound	31 46 54	38 47 100	L(0.60), M(0.40), H(0.00) L(0.15), M(0.85), H(0.00) L(0.00), M(0.00), H(1.00)	1 2 5	No suggestion Amplifier Function Mute Amplifier MixerLevelExt
4	HU does not switch off light	08 09 06	28 28 54	L(0.90), M(0.00), H(0.00) L(0.90), M(0.00), H(0.00) L(0.00), M(0.80), H(0.20)	0 0 3	No suggestion No suggestion HU NetworkManagement

	after shutdown					
5	TVTuner sleep current too high	35 03 46	24 27 38	L(0.70), M(0.00), H(0.00) L(0.85), M(0.00), H(0.00) L(0.60), M(0.40), H(0.00)	0 0 1	No suggestion No suggestion Amplifier Function Mute
6	Displays remain switched on after shutdown in RSE	51 02 08	22 29 49	L(0.60), M(0.00), H(0.00) L(0.95), M(0.00), H(0.00) L(0.05), M(0.95), H(0.00)	0 0 2	No suggestion No suggestion HU NetworkManagement
7	HU HighLevel retry is too many for play command	43 10	20 64	L(0.50), M(0.00), H(0.00) L(0.00), M(0.30), H(0.70)	0 4	No suggestion HU NetworkManagement
8	HeadPhoneAmplifier volume can not be increased	29 30 11	24 24 48	L(0.70), M(0.00), H(0.00) L(0.70), M(0.00), H(0.00) L(0.10), M(0.90), H(0.00)	0 0 2	No suggestion No suggestion RSE HeadPhoneAmplifier MixerLevelExt/Voulme
9	HU sends Notification.SetAll command but the RSE hangs	14	20	L(0.50), M(0.00), H(0.00)	0	No suggestion
10	RSE turns light off for hard reset	15	49	L(0.05), M(0.95), H(0.00)	2	RSE Function Block Diagnosis/NetworkManagement
11	TVTuner color function does not	07 34 19	24 27 57	L(0.70), M(0.00), H(0.00) L(0.85), M(0.00), H(0.00) L(0.00), M(0.65), H(0.35)	0 0 3	No suggestion No suggestion TVTuner Function Color

	work					
12	MultiMediaPlayer subtitle is wrong	24	48	L(0.10), M(0.90), H(0.00)	2	RSE MultiMediaPlayer Function SubTitleOnOff
13	Wrong time position of MultiMediaPlayer for CD	39 28	21 35	L(0.55), M(0.00), H(0.00) L(0.75), M(0.25), H(0.00)	0 1	No suggestion RSE MultiMediaPlayer Function CurrentTimePosition
14	TVTuner audio is always muted	26 29 31	22 29 38	L(0.60), M(0.00), H(0.00) L(0.95), M(0.00), H(0.00) L(0.60), M(0.40), H(0.00)	0 0 1	No suggestion No suggestion TVTuner Function Allocate
15	Country code of TVTuner is not set according to the HU command	52 19 22	21 24 35	L(0.55), M(0.00), H(0.00) L(0.70), M(0.00), H(0.00) L(0.75), M(0.25), H(0.00)	0 0 1	No suggestion No suggestion FBlock TVTuner Function CountryCode
16	No update of Function Blocks in system state Config Ok for RSE	12	52	L(0.00), M(0.90), H(0.10)	3	RSE NetworkManagement
17	Program name in display is wrong for TVTuner	20 22 23	32 40 40	L(0.90), M(0.10), H(0.00) L(0.50), M(0.50), H(0.00) L(0.50), M(0.50), H(0.00)	1 1 1	TVTuner Function ProgramName Block TVTuner Function CountryCode TVTuner Function CountryName

18	Volume of HeadPhoneAmplifier of RSE does not change according to the command	40 36 11	31 36 55	L(0.95), M(0.05), H(0.00) L(0.70), M(0.30), H(0.00) L(0.00), M(0.75), H(0.25)	1 1 3	RSE MultiMediaPlayer Function VideoFormat OperationType SetGet HeadPhoneAmplifier Function MixerLevelExt with Operation Type Increment RSE HeadPhoneAmplifier MixerLevelExt/Voulme
19	Video format of RSE is wrong	25 40	22 42	L(0.60), M(0.00), H(0.00) L(0.40), M(0.60), H(0.00)	0 2	No suggestion RSE MultiMediaPlayer Function VideoFormat OperationType SetGet
20	CD plays backward for pause command in RSE	45 43	21 47	L(0.55), M(0.00), H(0.00) L(0.15), M(0.85), H(0.00)	0 2	No suggestion RSE MultiMediaPlayer Function Deckstatus
21	No auto play for CD in RSE	47 45	21 83	L(0.55), M(0.00), H(0.00) L(0.00), M(0.00), H(1.00)	0 5	No suggestion RSE MultiMediaPlayer DeckStatus
22	Eject command from RSE does not out the CD	39 49	31 60	L(0.95), M(0.05), H(0.00) L(0.00), M(0.50), H(0.50)	1 3	RSE MultiMediaPlayer Function TrackPosition RSE Hardware
23	There is no update of TunedProgram if antenna is disconnected for TVTuner	31	24	L(0.70), M(0.00), H(0.00)	0	No suggestion
24	Music is	43	34	L(0.80), M(0.20), H(0.00)	1	RSE MultiMediaPlayer

	not paused for pause command					Function Deckstatus
25	DABTuner does not play any service if the car speed is too high	41	35	L(0.75), M(0.25), H(0.00)	1	DABTuner NetworkManegement Temperature
26	Directory path of Audio CD is not displayed	39	31	L(0.95), M(0.05), H(0.00)	1	RSE MultiMediaPlayer Function TrackPosition
27	currently available service of DABTuner is not displayed	38	25	L(0.75), M(0.00), H(0.00)	1	RSE MultiMediaPlayer Function CurrentDirectoryPath
28	change of MPR in MOST does not mute the audio	29 16	27 40	L(0.85), M(0.00), H(0.00) L(0.50), M(0.50), H(0.00)	0 1	No suggestion RSE HeadPhoneAmplifier Allocate/ResourceManagement
29	HeadPhoneAmplifier mute does not work	21 16 29	38 42 42	L(0.60), M(0.40), H(0.00) L(0.40), M(0.60), H(0.00) L(0.40), M(0.60), H(0.00)	1 2 2	HeadPhoneAmplifier Function Mute RSE HeadPhoneAmplifier Allocate/ResourceManagement RSE HeadPhoneAmplifier Function Mute
30	Light off	18	63	L(0.00), M(0.35), H(0.65)	4	HU Function Block

	error is not entered by HU in error memory					Diagnosis

Simulation set 3:

$l_2 = 0$ and $h_2 = 60$

Nr	Fault Symptom	Data base Fault ID	Similarity (%)	Degree of Membership	CL	Solutions
1	DVD auto play does not work	27 17 45	23 27 40	L(0.85), M(0.15), H(0.00) L(0.65), M(0.35), H(0.00) L(0.00), M(1.00), H(0.00)	1 1 2	RSE MultiMediaPlayer Function Repeat RSE MultiMediaPlayer DeckStatus Load/Unload RSE MultiMediaPlayer DeckStatus
2	stop button in RSE makes play	44 43 13	22 24 35	L(0.90), M(0.10), H(0.00) L(0.80), M(0.20), H(0.00) L(0.25), M(0.75), H(0.00)	1 1 2	RSE MultiMediaPlayer Function DeckStatus RSE MultiMediaPlayer Function Deckstatus RSE MultiMediaPlayer NeworkManagement/Deckstatus
3	TVTuner has no sound	31 46 54	38 47 100	L(0.10), M(0.90), H(0.00) L(0.00), M(0.65), H(0.35) L(0.00), M(0.00), H(1.00)	2 3 5	TVTuner Function Allocate Amplifier Function Mute Amplifier MixerLevelExt
4	HU does not switch off light after shutdown	08 09 06	28 28 54	L(0.60), M(0.40), H(0.00) L(0.60), M(0.40), H(0.00) L(0.00), M(0.30), H(0.70)	1 1 4	HU Command Missing TVTuner NetworkManagement HU NetworkManagement
5	TVTuner sleep current too high	35 03 46	24 27 38	L(0.80), M(0.20), H(0.00) L(0.65), M(0.35), H(0.00) L(0.10), M(0.90), H(0.00)	1 1 2	TVTuner Function CurrentChannel RSE Hardware Amplifier Function Mute

6	Displays remain switched on after shutdown in RSE	51 02 08	22 29 49	L(0.90), M(0.10), H(0.00) L(0.55), M(0.45), H(0.00) L(0.00), M(0.55), H(0.45)	1 1 3	Display RSE NetworkManagement HU NetworkManagement
7	HU HighLevel retry is too many for play command	43 10	20 64	L(1.00), M(0.00), H(0.00) L(0.00), M(0.00), H(1.00)	0 5	No suggestion HU NetworkManagement
8	HeadPhoneAmplifier volume can not be increased	29 30 11	24 24 48	L(0.80), M(0.20), H(0.00) L(0.80), M(0.20), H(0.00) L(0.00), M(0.60), H(0.40)	1 1 3	RSE HeadPhoneAmplifier Function Mute HU Function DeAllocate RSE HeadPhoneAmplifier MixerLevelExt/Voulme
9	HU sends Notification.SetAll command but the RSE hangs	14	20	L(1.00), M(0.00), H(0.00)	0	No suggestion
10	RSE turns light off for hard reset	15	49	L(0.00), M(0.55), H(0.45)	3	RSE Function Block Diagnosis/NetworkManagement
11	TVTuner color function does not work	07 34 19	24 27 57	L(0.80), M(0.20), H(0.00) L(0.65), M(0.35), H(0.00) L(0.00), M(0.15), H(0.85)	1 1 4	RSE MultiMediaPlayer Deckstatus TVTuner Antenna Disconnect TVTuner Function Color
12	MultiMediaPlayer subtitle is wrong	24	48	L(0.00), M(0.60), H(0.40)	2	RSE MultiMediaPlayer Function SubTitleOnOff

13	Wrong time position of MultiMediaPlayer for CD	39 28	21 35	L(0.95), M(0.05), H(0.00) L(0.25), M(0.75), H(0.00)	1 2	RSE MultiMediaPlayer Function TrackPosition RSE MultiMediaPlayer Function CurrentTimePosition
14	TVTuner audio is always muted	26 29 31	22 29 38	L(0.90), M(0.10), H(0.00) L(0.55), M(0.45), H(0.00) L(0.10), M(0.90), H(0.00)	1 1 2	RSE MultiMediaPlayer Function FastFwSpeed RSE HeadPhoneAmplifier Function Mute Parameter TVTuner Function Allocate
15	Country code of TVTuner is not set according to the HU command	52 19 22	21 24 35	L(0.95), M(0.05), H(0.00) L(0.80), M(0.20), H(0.00) L(0.25), M(0.75), H(0.00)	1 1 2	HU HMI Function TVTuner Function Color FBlock TVTuner Function CountryCode
16	No update of Function Blocks in system state Config Ok for RSE	12	52	L(0.00), M(0.40), H(0.60)	4	RSE NetworkManagement
17	Program name in display is wrong for TVTuner	20 22 23	32 40 40	L(0.40), M(0.60), H(0.00) L(0.00), M(1.00), H(0.00) L(0.00), M(1.00), H(0.00)	2 2 2	TVTuner Function ProgramName FBlock TVTuner Function CountryCode TVTuner Function CountryName
18	Volume of HeadPhoneAmplifie	40	31	L(0.45), M(0.55), H(0.00)	2	RSE MultiMediaPlayer Function VideoFormat OperationType SetGet

	r of RSE does not change according to the command	36 11	36 55	L(0.20), M(0.80), H(0.00) L(0.00), M(0.25), H(0.75)	2 4	HeadPhoneAmplifier Function MixerLevelExt with Operation Type Increment RSE HeadPhoneAmplifier MixerLevelExt/Voulme
19	Video format of RSE is wrong	25 40	22 42	L(0.90), M(0.10), H(0.00) L(0.00), M(0.90), H(0.10)	1 3	RSE MultiMediaPlayer Function FilePosition RSE MultiMediaPlayer Function VideoFormat OperationType SetGet
20	CD plays backward for pause command in RSE	45 43	21 47	L(0.95), M(0.05), H(0.00) L(0.00), M(0.65), H(0.35)	1 3	RSE MultiMediaPlayer DeckStatus RSE MultiMediaPlayer Function Deckstatus
21	No auto play for CD in RSE	47 45	21 83	L(0.95), M(0.05), H(0.00) L(0.00), M(0.00), H(1.00)	1 5	HU Function Allocation/Deallocation RSE MultiMediaPlayer DeckStatus
22	Eject command from RSE does not out the CD	39 49	31 60	L(0.45), M(0.55), H(0.00) L(0.00), M(0.00), H(1.00)	2 4	RSE MultiMediaPlayer Function TrackPosition RSE Hardware
23	There is no update of TunedProgram if antenna is disconnected for TVTuner	31	24	L(0.80), M(0.20), H(0.00)	1	TVTuner Function Allocate
24	Music is not paused for pause command	43	34	L(0.30), M(0.70), H(0.00)	2	RSE MultiMediaPlayer Function Deckstatus

25	DABTuner does not play any service if the car speed is too high	41	35	L(0.25), M(0.75), H(0.00)	2	DABTuner NetworkManegement Temperature
26	Directory path of Audio CD is not displayed	39	31	L(0.45), M(0.55), H(0.00)	2	RSE MultiMediaPlayer Function TrackPosition
27	Currently available service of DABTuner is not displayed	38	25	L(0.75), M(0.25), H(0.00)	1	RSE MultiMediaPlayer Function CurrentDirectoryPath
28	change of MPR in MOST does not mute the audio	29 16	27 40	L(0.65), M(0.35), H(0.00) L(0.00), M(1.00), H(0.00)	1 2	RSE HeadPhoneAmplifier Function Mute Parameter On/Off RSE HeadPhoneAmplifier Allocate/ResourceManagement
29	HeadPhoneAmplifier mute does not work	21 16 29	38 42 42	L(0.10), M(0.90), H(0.00) L(0.00), M(0.90), H(0.10) L(0.00), M(0.90), H(0.10)	2 3 3	HeadPhoneAmplifier Function Mute RSE HeadPhoneAmplifier Allocate/ResourceManagement RSE HeadPhoneAmplifier Function Mute
30	Light off error is not entered by HU in	18	63	L(0.00), M(0.00), H(1.00)	5	HU Function Block Diagnosis

	error memory					

7.3 Result Analysis

30 faults from a real car have been investigated. Probable solutions or fixing methods for the faults have been provided using the knowledge database. These faults were resolved by the software, hardware and system developers. The system provided solutions were compared with the real ones. Below is the summary of accuracy for the solutions.

Set 1 (l_2 = 20, h_2 = 80): correct 17, No Solutions 13
Set 2 (l_2 = 10, h_2 = 70): correct 23, incorrect 5, no solution 2
Set 3 (l_2 = 0, h_2 = 60): correct 22, incorrect 7, no solution 1

Figure 7.1 depicts the status of solutions of the input faults for l_2 = 20, h_2 = 80. Here it can be seen that only 17 suggestions were correct. It is not possible to suggest any solutions for 13 faults. There was no incorrect solution.

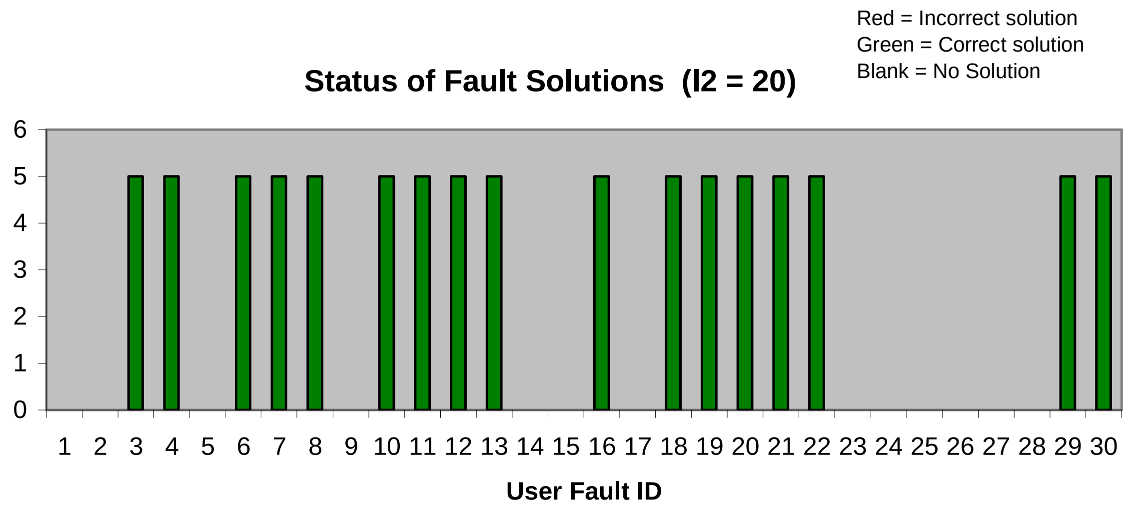


Figure 7.1: Graph on solutions of faults for l_2 = 20

Figure 7.2 depicts the status of solutions of the input faults for l_2 = 10, h_2 = 70. Here it can be seen that 23 suggestions were correct. It is not possible to suggest any solutions for only 2 faults. There are 5 incorrect solutions. It means five solutions which were suggested to fix faults proved to be wrong.

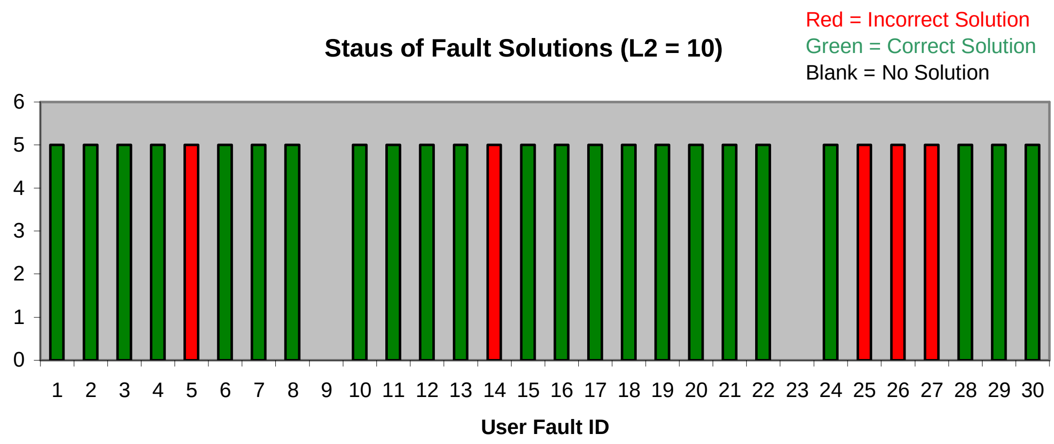


Figure 7.2: Graph on solutions of faults for $l_2 = 10$

Figure 7.3 depicts the status of solutions of the input faults for $l_2 = 0$, $h_2 = 60$. Here it can be seen that 22 suggestions were correct. It is not possible to suggest any solutions for only 1 fault. There are 7 incorrect solutions.

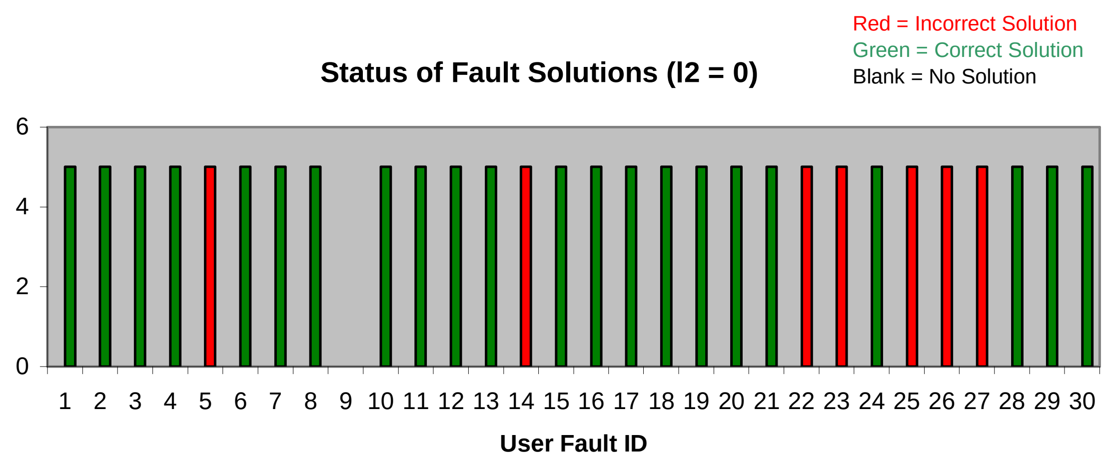


Figure 7.3: Graph on solutions of faults for $l_2 = 0$

Figure 7.4 shows the accuracy of faults for different values of l2. For l2 = 0, correct fault solutions are 22, for l2 = 10, correct faults are 23 and for l2 = 20, correct fault solutions are only 17. It shows that if l2 is chosen as very small or large, then the correct solutions of faults decrease. So l2 = 10 is an optimized value if one wants to maximize the correct solutions.

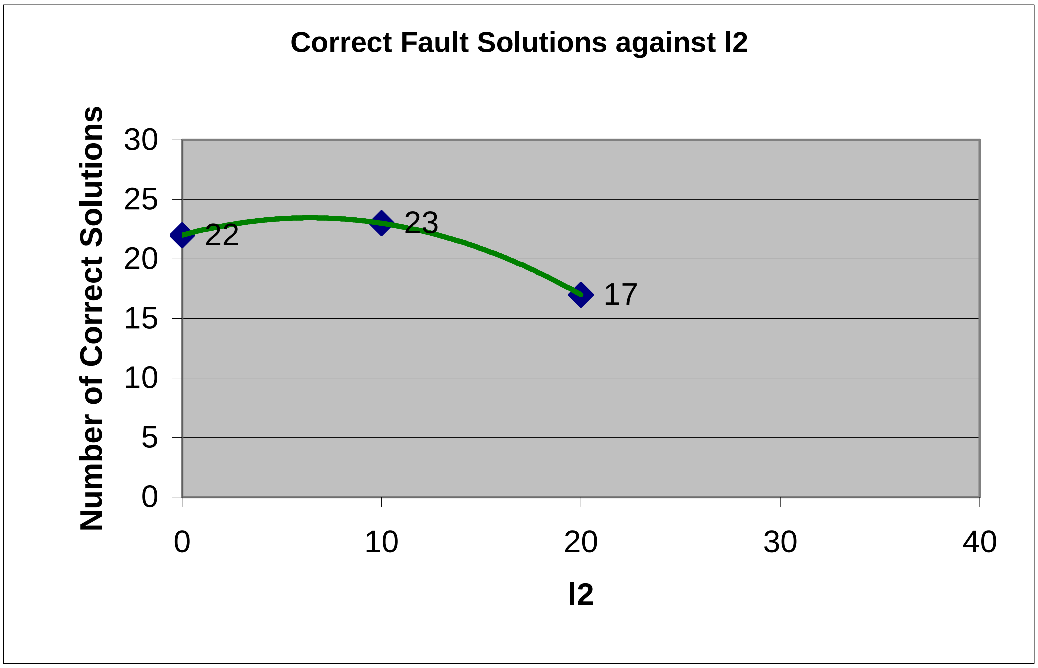


Figure 7.4: Graph for correct fault solutions against the border of Fuzzy function

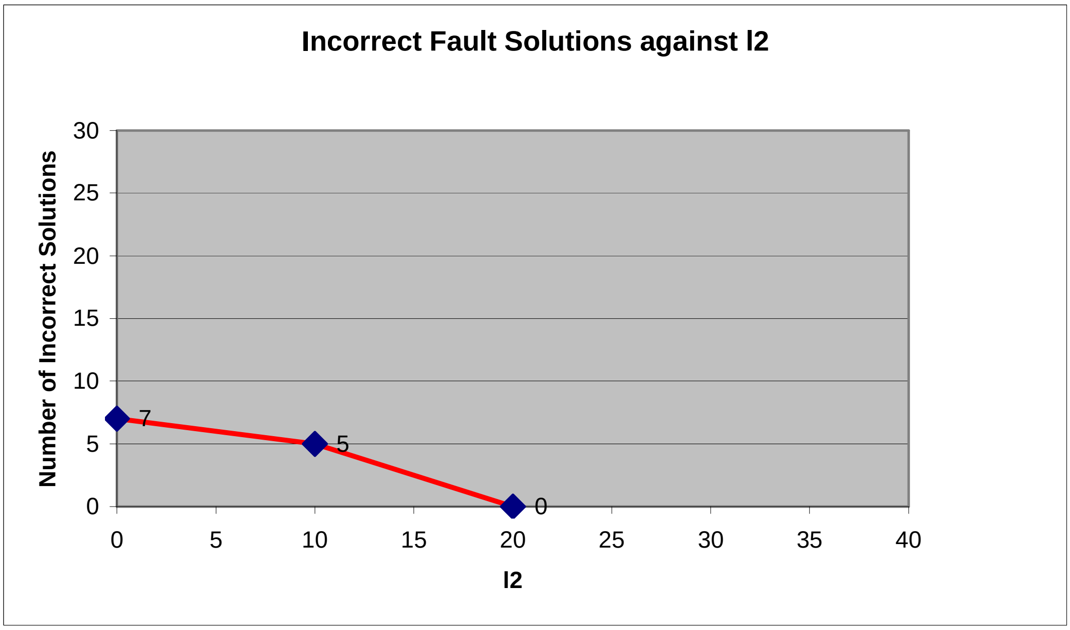


Figure 7.5: Graph for incorrect fault solutions against the border of Fuzzy function

Figure 7.5 shows the inaccuracy of faults for different values of l2. For l2 = 0, incorrect fault solutions are 7, for l2 = 10, incorrect faults are 5 and for l2 = 20, there is no incorrect solution. It shows that if l2 is chosen as very small, the incorrect solutions of faults increase. Incorrect solutions can be avoided for higher values of l2. Here in the simulation l2 = 20 is an optimized value if one wants to minimize the incorrect solutions.

Figure 7.6 shows the trends of no fault solutions for different values of l2. For l2 = 0, no fault solution is 1, for l2 = 10, no fault solutions are 2 and for l2 = 20, no fault solutions are 13. It shows that if l2 is chosen as very large, the number of no solutions for faults increases. When l2 is chosen as very low, the number of no solutions for faults decreases.

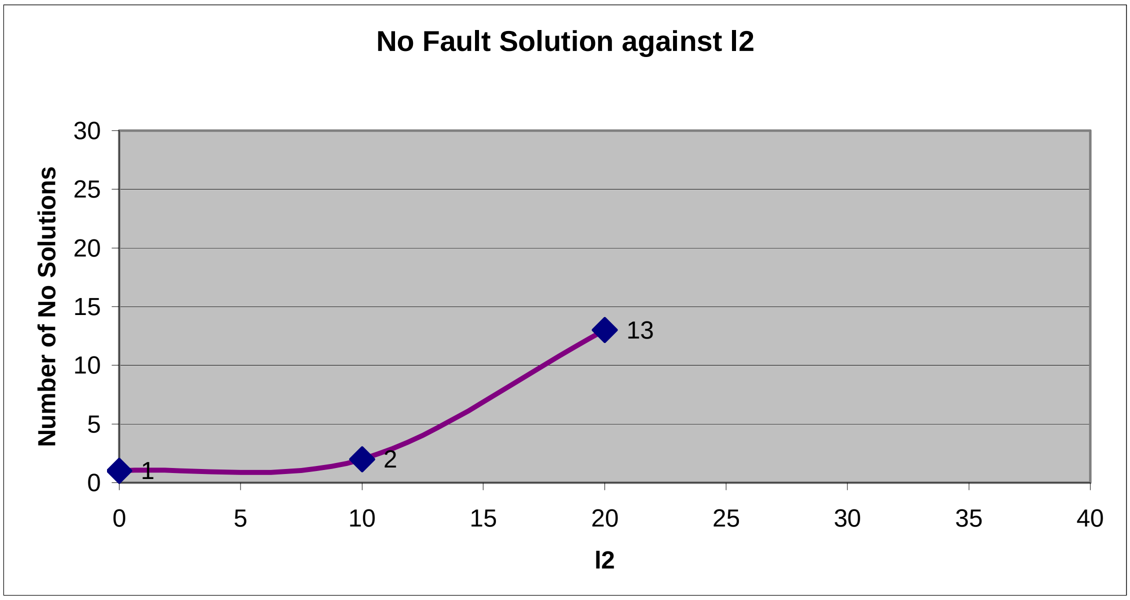


Figure 7.6: Graph for incorrect fault solutions against the border of Fuzzy function

After analyzing the simulation results, it can be concluded that if l2 is chosen as very small a large amount of correct fault solutions is produced, a very high amount of incorrect solutions is produced, but a lesser amount of no solutions is produced. If l2 is chosen as very high, a smaller amount of correct solutions is produced, a much less amount of incorrect solutions is produced, but a very high amount of no solutions is produced. Therefore l2 = 10 is chosen as an optimum tuning parameter.

Below is a table (Table 7.1) in which different confidence levels (CL) with correct and incorrect numbers of solutions are summarized for all the simulation sets.

Table 7.1:

Confidence Level	CL1	CL2	CL3	CL4	CL5
Number of correct solutions	16	20	9	5	7
Number of incorrect solutions	14	14	4	2	0

Figure 7.7 depicts the behavior of this system for correct solutions against different confidence levels. The graph shows that when the level of confidence (CL) increases, the rate of correct solutions of fault fixing becomes higher.

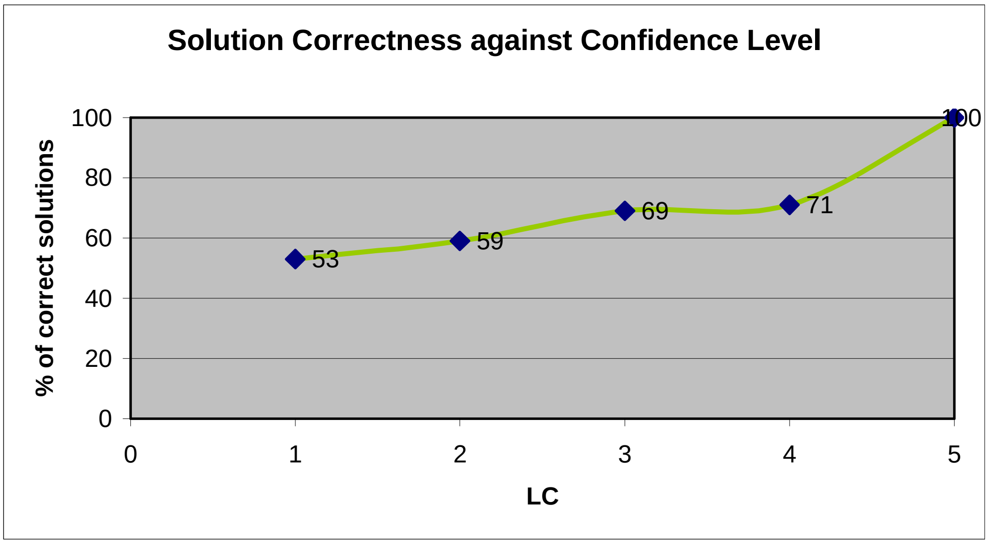


Figure 7.7: Rate of correct solutions for different confidence levels

Figure 7.8 depicts the behavior of this system for incorrect solutions against different confidence levels. The graph shows that when the level of confidence increases, the rate of incorrect solutions for fault fixing becomes lower.

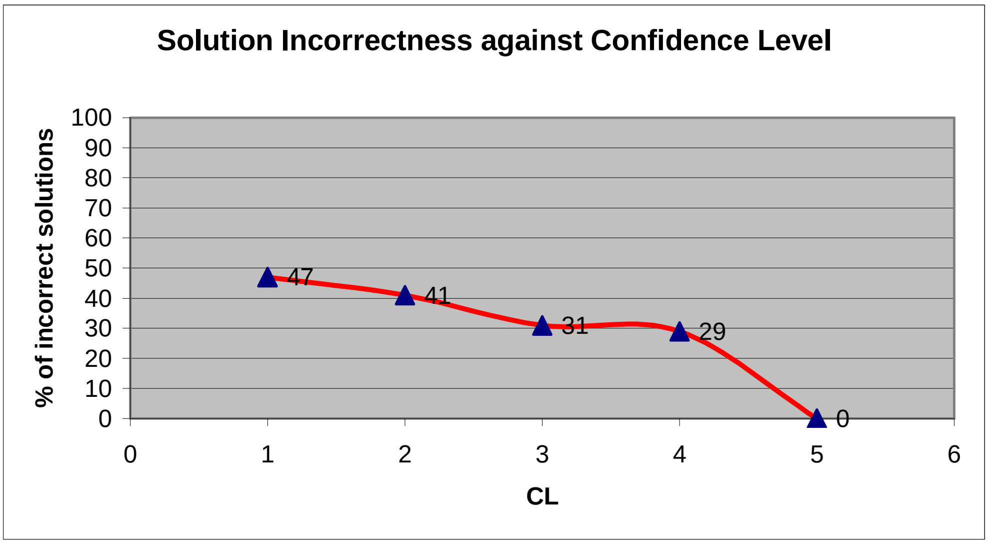


Figure 7.8: Rate of incorrect solutions for different confidence levels

The above results of the simulation prove that the higher the level of confidence, the greater the chance of obtaining a correct fault fixing solution. This validates the concept of Confidence Level.

7.4 System Validation

Solutions for thirty different faults have been provided. The solutions are verified against real fixing methods for the purpose of making a conclusion regarding the validity of the system results.

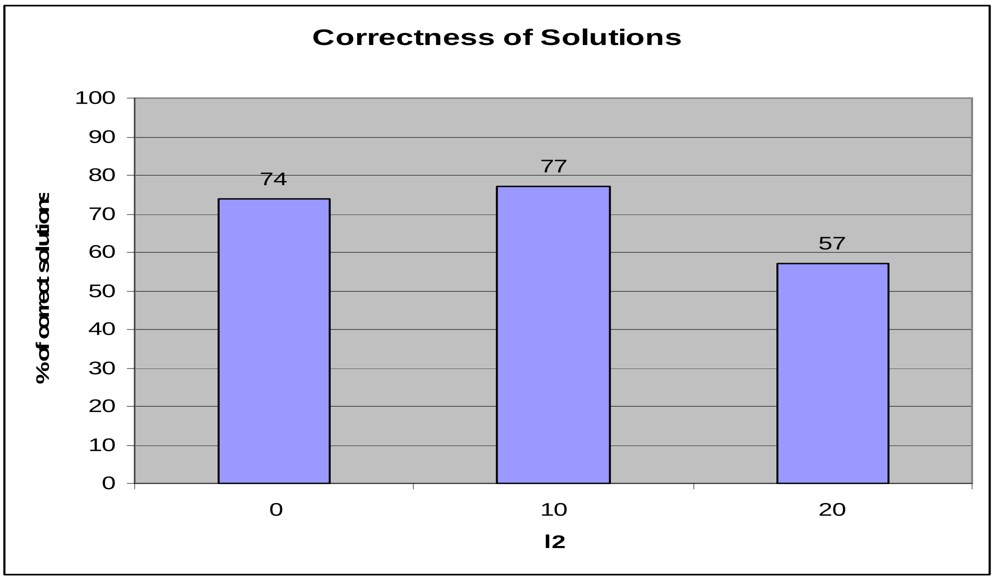


Figure 7.9: Percentage of correct solutions of faults

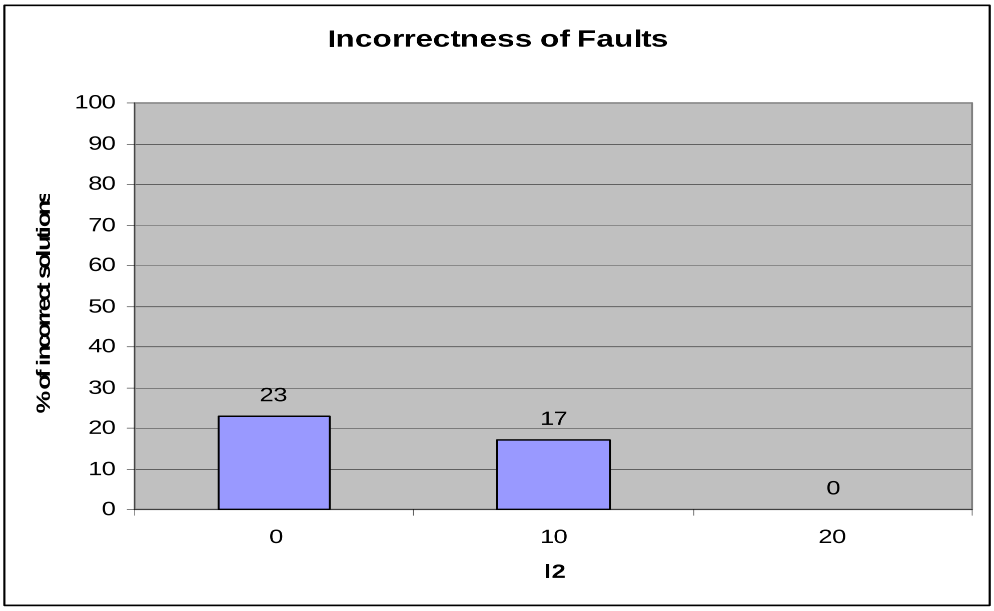


Figure 7.10: Percentage of incorrect solutions of faults

Figure 7.9 reveals that up to 74% correct solutions for faults are achievable if the tuning parameters can be optimized. Figure 7.11 shows that this system is incapable of providing up to 43% of solutions. Figures 7.10 shows that up to 23% fault solutions are wrong. These are the drawbacks of this system.

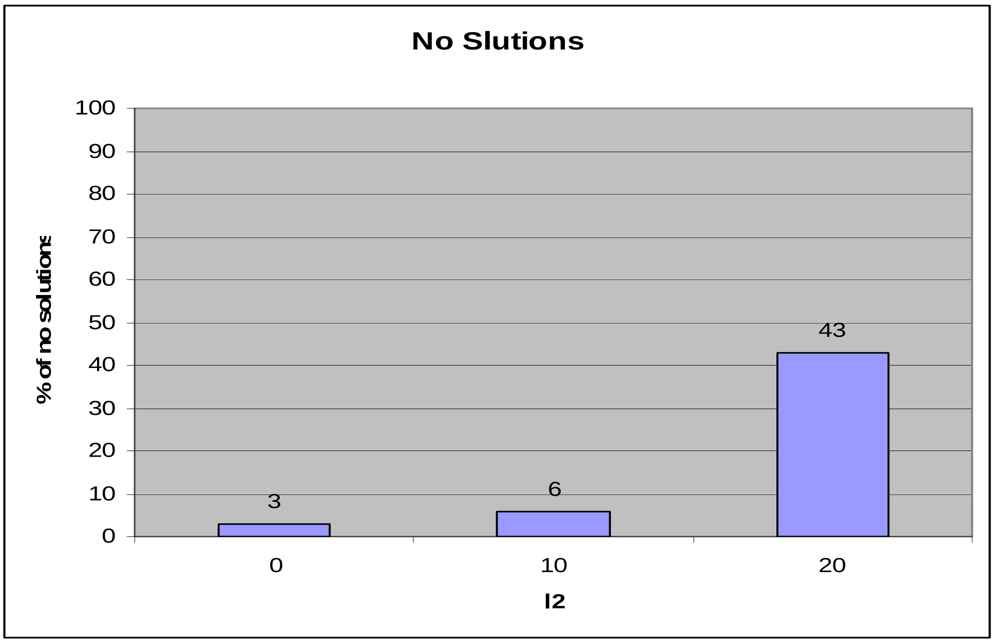


Figure 7.11: Percentage of no solutions of faults

Chapter 8

Conclusion and Future Work

In this study, a new system to diagnose a fault has been developed. The main goal was to find out the cause of a fault and provide a solution(s). Based on the user provided fault symptoms, one or more possible solutions are proposed to fix it. The methodologies of current fault classifications in different fields of applications have been researched. Then several new schemes to classify faults in different categories were proposed. An extensive review of existing literature on fault diagnosis was performed. There is no single system which can be used for newly emerging automotive infotainment and multimedia system which is based on MOST technology. New techniques for designing this system were proposed in order to fulfill the goals of this project. Pattern recognition technique was used to select features, cosine similarity method for similarity matching, fuzzy technique for decision making on a fault classification and the level of confidence concept to propose suggestions for fault fixing.

Thirty real faults from a car were collected to validate the system. The simulation result shows that this system can provide up to 77% correct solutions in the best case and up to 57% correct solutions in the worst case. For the solutions on fault fixing, confidence level has been defined. Based on the confidence level, it is possible to forecast the probability of whether a fault can really be fixed by these suggestions. In this system it has been shown that if the confidence level of a fault fixing solution is high, there is more certainty of being able to fix the fault with these suggestions.

Research shows that up to 3% of the total costs of a car development is spent on fault analyzing and fault fixing. System designers and developers of a car are the persons responsible for fixing any faults of the automotive infotainment system from the root level. A fault arises due to wrong behavior of hardware or software or both. Identifying the causes of a fault and offering solutions are the main tasks that must be carried out beforehand in order to fix it. Using this new system for fault fixing would certainly reduce many of the efforts of system developers and designers. This new system can be used by automotive manufacturers and suppliers to reduce the total development cost.

One of the main advantages of this developed system is that it takes the input from any kind of automotive user; from a driver to an expert engineer in textual format. The symptoms are described in natural language. The system extracts the necessary and important information according to its need in order to generate its outputs. It does not rely on any question/answer formats from the user.

The main limitation of this system is that the knowledge base is not mature enough. It means that this system lacks a large knowledge base for handling all the faults observed by a user. This knowledge base should be improved. In this work only three different values of the tuning parameter have been chosen. The system behavior and the output when using other values of tuning parameters should also be observed. The system is only applicable to the automotive infotainment and multimedia system. Further work should be carried out in order to apply it in other areas. This system lacks an element of self learning. A proper technique should be explored and integrated within the system for self learning.

References

[1] Robert H. Chen, Hok K. Ng, Jason L. Speyer, D. Lewis Mingori. *Testing and Evaluation of Robust Fault Detection and Identification for a Fault Tolerant Automated Highway System*. UCB-ITS-PRR-2002-24 California PATH Research Report, August 2002.

[2] P. Struss, A. Malik, M. Sachenbacher. *Case Studies in Model-Based Diagnosis and Fault Analysis of Car-Subsystems*. In proceeding ECAI, 1996.

[3] Umesh B. Nikam. *A Fault Diagnostic System for an Unmanned Autonomous Mobile Robot*. Division of Research and Advanced Studies of the University of Cincinnati, 1997.

[4] *Planning an intelligent Transport System - A Guide to System Architecture*. October 2002.

[5] Sampath Kanakaraju. *Online Fault Diagnosis System for an Autonomous Guided Vehicle*. A thesis submitted to the Division of Graduate Studies and Advanced Research of the University of Cincinnati, 2000.

[6] Christopher Tubb, Edin Omerdic. *Fault Detection and Handling*. IMPROVES Technical Report FD 001(n). 8th October 2001.

[7] Yi Lu Murphey. *Automotive Fault Diagnosis: Distributed Diagnostic Agent System*. Department of Electrical & Computer Engineering The University of Michigan-Dearborn, 2nd Annual Winter Work Shops Series - December 2-6, 2002.

[8] Karthik Balakrishnan, Vasant Honavar. *Intelligent Diagnosis Systems*. Journal of Intelligent Systems 8(3/4):pp. 239-290, 1998.

[9] Karthik Balakrishnan, Vasant Honavar. *Intelligent Diagnosis Systems*. http://www.cs.iastate.edu/~honavar/aigroup.html.

[10] Mattias Nyberg . *Framework and Method for Model Based Diagnosis with Application to an Automotive Engine. IFAC.* ECC, 1999.

[12] Marcus Boerner, Mina Zele , Rolf Isermann. *Comparison of Different Fault Detection Algorithms for Active Body Control Components: Automotive Suspension System.* Proceedings of the American Control Conference, Arlington, VA June 25-27, 2001.

[13] Marcus Börner, Rolf Isermann. *Supervision, Fault Detection, And Sensor Fault Tolerance of Passenger Cars*. 5th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes, Washington, D.C., June 9-11, 2003.

[14] Rolf Isermann, Marcus Börner. *Diagnosis Methods For Electronically Controlled Vehicles*. 3. Braunschweiger Symposium Automatisierungs- und Assistenzsysteme für Transportmittel - Möglichkeiten, Grenzen, Risiken -, Braunschweig, Deutschland, 27. und 28. Februar 2002.

[15] Peter Loborg. *Error Recovery in Automation An Overview*. AAAI 94 Spring Symposium on Detecting and Resolving Errors in Manufacturing Systems, Stanford, Ca, spring 1994.

[16] Jianhui Luo, Madhavi Namburu, Krishna Pattipati, Liu Qiao, Masayuki Kawamoto, Shunsuke Chigusa. *Model-based Prognostic Techniques*. IEEE AUTOTESTCON 2003 Conference, Disneyland Hotel, Anaheim, California, September 22-25, 2003.

[17] Marie Matoes, Philippe Robin, Sylvain Sauvage, V. Joloboff, G. Madhusudan, Younès Bennani, Philippe Dague, Luc Bourgeois. *Environment For Evolutionary Automotive Diagnosis*. Convergence 2002 Transportation Electronics Conference, Diagnostics Session, Detroit, Michigan, USA, October 21, 2002.

[18] Philip Koopman. *Fault Tolerance. Distributed Embedded Systems.* Carnegie Mellon, November 8, 2000.

[19] Kabitzsch K.; Kotte G.; Vasyutynskyy V. *Classification Method for Fault Diagnosis in Networked CIM Systems*. 10th IFAC Symposium on Information Control Problems in Manufacturing INCOM 2001, Vienna, Austria, September 2001.

[20] Volodymyr Vasyutynskyy. *Offline Diagnosis of Complex Automation Systems.* www.inf.tu-dresden.de/TU/Informatik/IRS/ Rathen2001/Folien2001_Vasyutynskyy.ppt, 2001.

[21] Wang Xuanyin Gao Lei and Tao Guoliang. *Research on Fuzzy Neural Networt Fault Diagnosis of Pneumatic System of Automatic Production Line*. Fifth International Conference on Fluid Power Transmission and Control (ICFP2001), 3-5 April, 2001.

[22] Friedemann Bitsch, *Process Model for the Development of System Requirements Specifications for Railway Systems*. Workshop on Software specification of safety relevant transportation control tasks, Braunschweig, April 2002.

[23] Daniel Louis. *Monitoring and Diagnosis of Continuous Dynamic Systems using Semi Quantitative Simulation*. Dissertation of Doctor of Philosophy, University of Texas at Austin, May 2002.

[24] R. A. Demillo and A. P. Mathur. *A grammar based fault classification scheme and its application to the classification of the errors of TEX*. Technical Report SERC-TR-165-P, Purdue University, 1995.

[25] Susanne Manz. *Online Fault Detection and Diagnosis of Complex Systems based on Hybrid Component Models*. 14th International Congress on Condition Monitoring and Diagnostic Engineering Management - COMADEM 2001.

[26] Susanne Manz. *Fuzzy Based Qualitative Models in Combination with Dynamical Models for Online Monitoring of Technical Systems*. International Conference on Computational Intelligence

for Modelling, Control and Automation - CIMCA 2001.

[27] Papadopoulos Y. *Model-based system monitoring and diagnosis of failures using statecharts and fault trees.* Int. J. of Reliability Engineering and System Safety, 81:325-341, 2003.

[28] K. Rahnamai, P. Arabshahi, T.-Y. Yan, T. Pham, and S. G. Finley. *An Intelligent Fault Detection and Isolation Architecture For Antenna Arrays*. TDA Progress Report 42-132 February 15, 1998.

[29] Chetan P. Chiba , Setumo Mohapi and Hu Hanrahan. *Fault Management Service in ATM Networks using TINA Network Resource Architecture*. SATNAC Conference, 2001**.**

[30] Buskens. R. W., ANn Bianchini, R. P., JR. *Distributed on-line Diagnosis in the Presence of Arbitrary Faults*. Dtg. Papers FTCS-21 (June), 470 479, 1993.

[31] Chen, C-H. and Honavar, V. *A Neural Network Architecture for High-Speed Database Query Processing*. *Microcomputer Applications*. vol. 15, no. 1. pp. 7-13, 1996.

[32] Siddarth Subramanian, Raymond J. Mooney. *Qualitative Multiple-Fault Diagnosis of Continuous Dynamic Systems Using Behavioral Modes.* Appears in Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI-96) pp. 965-970, Portland, OR, August 1996.

[33] Haapanen Pentti, Helminen Atte. Failure Mode Andeffects Analysisof Software-Based Automation Systems. STUK-YTO-TR 190 / August 2002.

[34] D. Y. Chen, Y. Hong, and K. S. Trivedi. *Classification of Fults, Erors and Filures in Cmmunication Sstems*. Technical report to ARO MURI project, Department of Electrical and Engineering, University of Duke, 2002.

[35] *http://www.pppl.gov/eshis/procedures/eng008.pdf.*

[36] Friedemann Bitsch. *Safety-Patterns - The Key to Formal Specification of Safety-Requirements*. SAFECOMP 2001 - Computer Safety, Reliability and Security, 20th International Conference, Budapest, Ungarn, September 2001.

[37] Friedemann Bitsch. *Way for Applicable Formal Specification of Safety Requirements by Tool-support*. FORMS 2003.

[38] O. Anton, B. Lercher, G. Reinhart. *Modelling of Faults and Fault Recovery: An Essential Aspect of Mechatronic System Design*. Annals of 2002 Int'l CIRP Design Seminar, Hong Kong, 16-18 May 2002.

[39] Puneet Kukkal, John B. Bowles, Ronald D. Bonnell. *Database Design for failure Modes and Effects Analysis*. Proceedings - Annual Reliability and Maintainability Symposium, 1993.

[40] Mark A. Thurber, Marilyn R. Miars. *Implementation of Fault Detection and Diagnosis in an Intelligent Vehicle Electrical System*. Society of Automotive Engineers, 2001-01-2790, 2001.

[41] *Heisenbugs and Bohrbugs: Why are they diferent?* www.cs.rutgers.edu/~rmartin/teaching/spring03/cs553/papers01/06.pdf, March 8, 2003.

[42] Robyn R. Lutz, Robert M. Woodhouse. Bi-directional Analysis for Certification of Safety-Critical Software. *1st International Software Assurance Certification Conference (ISACC'99)*, Washington, D. C., Feb. 28-March 2, 1999.

[43] Ruffo, G. *Learning Single and Multiple Instance Decision Tree for Computer Security Applications*. Doctoral dissertation, Department of Computer Science, University of Turin, Torino, Italy, 2000.

[44] Claudia Picardi. *Diagnosis from System Modelling to On-Board Software*. PhD Thesis, Universit degli Studi di Torino, Dipartimento di Informatica, 1998-2002.

[45] Donald P. McKay Timothy W. Finin, Anthony O'Hare. *The Intelligent Database Interface: Integrating AI and Database Systems*. Proceedings of the 8th National Conference on Artificial Intelligence (AAAI-90), 1990.

[46] Randy Goebel. *If Inductive Logic Programming Leads, Will Data Mining Follow?* Proceedings of the Japanese Society for Artificial Intelligence, Foundations of AI Special Interest Group Workshop on Inductive Logic Programming, pages 39--49, Hokkaido University, Sapporo, Japan, July 31--August 2, 1996.

[47] Annie Gal. *Cooperative Responses In Deductive Databases.* Department of Computer Science, University of Maryland, July 1988.

[48] Peter Struss, Chris Price. *Model-Based Systems in the Automotive Industry*. AI Magazine, winter, 2003.

[49]. http://en.wikipedia.org/wiki/

[50] *CANalyzer Manual*, Version 4.0. Vector Informatik GmbH. 2003.

[51] OptoLyzer4MOST. *Tool Platform for Analysis*. User Manual for Version V2.0.1. 08 July 2004.

[52] Karen Kukich. *Techniques for automatically correcting words in text*. ACM Computing Surveys, 24(4):377–439, December 1992.

[53] Dengyong Zhou,Jason Weston, Arthur Gretton, Olivier Bousquet, Bernhard Schoelkopf. *Ranking on Data Manifolds* Technical Report No. TR-113, June 2003.

[54] http://www.seattlerobotics.org/encoder/mar98/fuz/fl_part1.html

[55] Heide Brücher, Gerhard Knolmayer, Marc-André Mittermayer. *Document Classification Methods forOrganizing Explicit Knowledge.* University of Bern, Institute of Information Systems Research Group Information Engineering, Switzerland.

[56] Dumais, S. Platt, J., Heckermann, D., Sahami, M. *Inductive Learning Algorithms and Representations for Text.* Proceedings of the 7th International Conference on Information and Knowledge Management, pp. 148-155. 1998.

[57] Rachel TszWai, Lo, Ben He, Iadh Ounis. *Automatically Building a Stopword List for an Information Retrieval System.* 5th DutchBelgium Information Retrieval Workshop (DIR) Utrecht, the Netherlands. 2005.

[58] Sriharsha Veeram, Diego Sona, Paolo Avesani. *Hierarchical Dirichlet Model for Document Classification.* Proceedings of the 22 nd International Conference on Machine Learning, Bonn, Germany. 2005.

[59] *Data Mining: Concepts and Techniques – Mining Text Data*. http://cs.nyu.edu/courses/spring08/G22.3033-003/10text.ppt#1102,30,Categorization Methods

[60] Chih-Ping Wei; Yuan-Xin Dong. *A mining-based category evolution approach to managing online document categories.* System Sciences, 2001. Proceedings of the 34th Annual Hawaii International Conference on Volume , Issue , 3-6 Jan. 2001.

[61] YongSeog Kim, W. Nick Street, and Filippo Menczer. *Feature Selection in Data Mining.* University of Iowa, USA.

[62] Mark Last, Abraham Kandel, Oded Maimon, Eugene Eberbach. *Anytime Algorithm for Feature Selection.* Department of Computer Science and Engineering, University of South Florida, USA.

[63] Shmuel Brody. *Cluster-Based Pattern Recognition in Natural Language Test.* Master Thesis. August 2005.

[64] Faramarz Valafar. *Pattern Recognition Techniques in Microarray Data Analysis: A Survey*. Special issue of Annals of New York Academy of Sciences, *Techniques in Bioinformatics and Medical Informatics*. (980) 41-64, December 2002.

[65] Ahmad T. Al-Taani. *An Expert System for Car Failure Diagnosis.* Proceedings of World Academy of Science, Engineering and Technology Volume 7 August 2005.

[66] H. Kopetz. *COTS Technology and Issues Automotive.* http://www.laas.fr/IFIPWG/Workshops&Meetings/44/W2/04-Kopetz.pdf. June 2003.

[67] F. Cascio, L. Console, M. Guagliumi. *On-board diagnosis of automotive systems: from dynamic qualitative diagnosis to decision trees.* Supported by the European Commission, DG XII (project BE 95/2128).

[68] Sriram Narasimhan, Gautam Biswas. Model-based Diagnosis of Hybrid Systems. QSS, NASA Ames Research Center, Moffett Field, CA 94035 and Group Dept. of EECS & ISIS, Vanderbilt University, Nashville, TN 37235.

[69] Cheng-Ching Yu and Chyuan Lee. *Fault Diagnosis Based on Qualitative/Quantitative Process Knowledge.* Dept. of Chemical Engineering, National Taiwan Institute of Technology, Taipei, Taiwan 10772, ROC. AICHE Journal, April 1991.

[70] Venkat Venkatasubramanian a, Raghunathan Rengaswamy b, Kewen Yin c, Surya N. Kavuri. *A review of process fault detection and diagnosis Part I: Quantitative model-based methods.* Published by Elsevier Science Ltd. Computers and Chemical Engineering 27 (2003) 293-311.

[71] Philip Haves and Sat Kartar Khalsa. *Model-Based Performance Monitoring: Review of Diagnostic Methods and Chiller Case Study.* ACEEE 2000 Summer Study on Energy Efficiency in Buildings, Efficiency and Sustainability, Asilomar Conference Center, Pacific Grove, CA, August 20-25, 2000.

Appendix I

List of stop words:

a	has	someone
about	hasnt	something
above	have	sometime
across	he	sometimes
after	hence	somewhere
afterwards	her	still
again	here	such
against	hereafter	system
all	hereby	take
almost	herein	ten
alone	hereupon	than
along	hers	that
already	herself	the
also	himself	their
although	his	them
always	how	themselves
am	however	then
among	hundred	thence
amongst	i	there
amoungst	ie	thereafter
amount	if	thereby
an	in	therefore
and	inc	therein
another	indeed	thereupon
any	interest	these
anyhow	into	they
anyone	is	thick
anything	it	thin
anyway	its	third
anywhere	itself	this
are	keep	those
around	last	though
as	latter	three
at	latterly	through
back	least	throughout
be	less	thru
became	ltd	thus
because	made	to

become	may	too
becomes	me	top
becoming	meanwhile	toward
been	might	towards
beforehand	mill	twelve
behind	mine	twenty
being	more	two
beside	moreover	un
besides	most	under
between	mostly	until
beyond	move	up
bill	much	upon
both	must	us
bottom	my	very
but	myself	via
by	name	was
call	namely	we
can	neither	well
cannot	nevertheless	were
cant	him	what
co	nine	whatever
con	nobody	when
could	noone	whence
couldnt	not	whenever
cry	now	where
detail	nowhere	whereafter
do	of	whereas
does	often	whereby
done	on	wherein
down	once	whereupon
due	one	wherever
during	only	whether
each	onto	which
eg	or	while
eight	other	whither
either	others	who
eleven	otherwise	whoever
else	our	whole
elsewhere	ours	whom
etc	ourselves	whose
even	out	why
ever	over	will
every	own	with
everyone	part	within
everything	per	without
everywhere	perhaps	would
except	please	yet
few	put	you

fifteen	rather	your
fify	re	yours
fill	see	yourself
fire	seem	yourselves
first	seemed	
five	seeming	
for	seems	
former	several	
formerly	she	
forty	should	
found	show	
four	side	
from	since	
front	sincere	
further	six	
get	sixty	
give	so	
go	some	
had	somehow	

Appendix II

Word stemming List

The words inside the first bracket are mapped to the word outside the bracket. For example, words Digital Video Disc and Digital Disc are mapped to word DVD.

Play (playing, played, plays)
DVD (Digital Video Disc, Digital Disc)
CD (Compact Disc, Disc, Audio Disc, Video Disc)
Navigation (Navi)
RSE (Rear Seat Entertainemnt, Rear Entertainment)
TVTuner (TV, Television, ISDBT, DAVT)
DABTuner (DAB, Digital Radio)
Radio (AM, FM, Analog Radio)
MOST (Message Oriented System Transport)
CAN (Controller Area Network)
Forward (Fwd)
Backward (Bwd)
Current (Power)
Voltage (Low Voltage, High Voltage)
Media (Multi Media)
HeadPhoneAmplifier (Head Phone Amplifier, HPA)
Tempearture (Temp)
Diagnosis (Diag)
FunctionBlock (FBlock)
FktID (Function ID, FunctionID)
InstID (Instance ID)
OpType (OperationType, Operation Type)
Error (Fault, Faulty, Erroneous, Defect)
Video (Movie)
Audio (Music)
HU (HeadUnit, Head Unit)
Cable (Connector)
SW (Software, Soft Ware)
HW (Hard Ware, HardWare)
SWDL (Software Download, SoftwareDownload)
NWM (Network Management, Networkmanagement)
FastForward (FstFwd)
Optolyzer (Opto)
Start (Starts, Started, Starting)
Startup (Start up)
Shutdown (Shut down)

Run (Runs, Ran, Running)
Press (Pressed, Pressing, Presses)
Eject (Ejects, Ejected, Ejecting, Ejct)
Pause (Pause, Pausing, Paused)
Connect (Connection, Connecting, Connected)
Allocate (Allocating, Allocation, Allocated)
Mute (Muted, Muting)
Disconnect (Disconnecting, Disconnected, Disconnection)
RBD (Ring Break Diagnosis)
Load (Loading, Loaded)
Unload (Unloaded, Unloading)
Light (Lighting, Lights)
Wake (Wakeup, wake up)