

CRANFIELD UNIVERSITY

CORDELIA MATTUVARKUZHALI EZHILARASU

A FRAMEWORK FOR AEROSPACE VEHICLE REASONING
(FAVER)

SCHOOL OF AEROSPACE, TRANSPORT AND MANUFACTURING
Integrated Vehicle Health Management

DOCTOR OF PHILOSOPHY
Academic Years: 2017 - 2020

Supervisor: Prof. Ian K Jennions
Associate Supervisor: Dr. Zakwan Skaf
November 2020

CRANFIELD UNIVERSITY

SCHOOL OF AEROSPACE, TRANSPORT AND MANUFACTURING
Integrated Vehicle Health Management

DOCTOR OF PHILOSOPHY

Academic Years: 2017 - 2020

CORDELIA MATTUVARKUZHALI EZHILARASU

A FRAMEWORK FOR AEROSPACE VEHICLE REASONING
(FAVER)

Supervisor: Prof. Ian K Jennions
Associate Supervisor: Dr. Zakwan Skaf
November 2020

© Cranfield University 2020. All rights reserved. No part of this
publication may be reproduced without the written permission of the
copyright owner.

ABSTRACT

Airliners spend over 9% of their total revenue in Maintenance, Repair, and Overhaul (MRO) and working to bring down the cost and time involved. The prime focus is on unexpected downtime and extended maintenance leading to delays in the flights, which also reduces the trustworthiness of the airliners among the customers. One of the effective solutions to address this issue is Condition based Maintenance (CBM), in which the aircraft systems are monitored frequently, and maintenance plans are customized to suit the health of these systems. Integrated Vehicle Health Management (IVHM) is a capability enabling CBM by assessing the current condition of the aircraft at component/ Line Replaceable Unit/ system levels and providing diagnosis and remaining useful life calculations required for CBM. However, there is a lack of focus on vehicle level health monitoring in IVHM, which is vital to identify fault propagation between the systems, owing to their part in the complicated troubleshooting process resulting in prolonged maintenance. This research addresses this issue by proposing a Framework for Aerospace Vehicle Reasoning, shortly called FAVER. FAVER is developed to enable isolation and root cause identification of faults propagating between multiple systems at the aircraft level. This is done by involving Digital Twins (DTs) of aircraft systems in order to emulate interactions between these systems and Reasoning to assess health information to isolate cascading faults. FAVER currently uses four aircraft systems: i) the Electrical Power System, ii) the Fuel System, iii) the Engine, and iv) the Environmental Control System, to demonstrate its ability to provide high level reasoning, which can be used for troubleshooting in practice. FAVER is also demonstrated for its ability to expand, update, and scale for accommodating new aircraft systems into the framework along with its flexibility. FAVER's reasoning ability is also evaluated by testing various use cases.

Keywords:

Condition Based Maintenance; Vehicle health monitoring; Diagnosis; Reasoning; Aircraft systems; Fault propagation; Aircraft accidents

தொட்டனைத் தூறும் மணற்கேணி மாந்தர்க்குக்

கற்றனைத் தூறம் அறிவு

அதிகாரம்: கல்வி; குறள்: 396

In sandy soil, when deep you delve, you reach the springs below;

The more you learn, the freer streams of wisdom flow

Chapter: Learning; Thirukkural: 396

Dedicating this work to my husband, Vasanth, without whom this PhD journey would not have been possible.

And, of course, to our six-month-old baby, Aazhini, whose late-night insightful chats helped me with my thesis.

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my supervisor, Prof. Ian K Jennions, for being an excellent teacher, guide, and a mentor. I have learned a lot and have grown as a researcher and as a person, thanks to my journey with him. I have enjoyed our conversations immensely. I feel I used a good part of my ‘luck potion’ to have found and worked with such a talented and a wonderful person.

I would also like to thank Dr. Zakwan Skaf for approaching me to pursue PhD after going through my initial proposal, for his guidance in the initial phase of the research, and for his contribution in few of the chapters.

I would like to thank The Boeing Company for funding this project and extending support during this study.

My friends from the IVHM Centre, especially Mehmet, Shaf, Vivian, and Nithya, deserve a special thanks; they were a good support throughout my PhD journey. I enjoyed our adventures in the otherwise peaceful university campus.

Big thanks to my family, especially my mum, for being my little cheering group. I would not have been able to complete my PhD without their support and encouragement.

Last but most importantly, I would like to thank my husband, Vasanth. From supporting me financially, to listening to all my rambling thoughts, bearing with reading through all my papers’ first drafts, taking care of our baby and me, and encouraging me to bring my PhD to completion, he has turned his life upside down to help me achieve one of my dreams. In retrospection, I am lucky to have found him a decade back, for I could not think of living through the pandemic and lockdowns without him beside me.

As far as acknowledgements go, I would like to acknowledge the pandemic caused by COVID19, which made me, and everyone around me learn and reassess the value of our lives.

TABLE OF CONTENTS

ABSTRACT	i
ACKNOWLEDGEMENTS	iv
LIST OF FIGURES	ix
LIST OF TABLES	xiii
LIST OF EQUATIONS.....	xiv
LIST OF ABBREVIATIONS	xv
1 Introduction	1
1.1 Motivation behind this research	2
1.1.1 Aircraft systems interactions	2
1.1.2 Failure propagation across systems	3
1.1.3 Example of Safety assurance during Design & Development	4
1.1.4 IVHM for Condition Based Maintenance	5
1.2 Aim and Objectives.....	7
1.3 Methodology.....	8
1.3.1 Phase 1: Literature review	8
1.3.2 Phase 2: Use case conceptualization.....	11
1.3.3 Phase 3: Framework for Aerospace Vehicle Reasoning (FAVER)	12
1.3.4 Phase 4: Implementation and Testing	13
1.3.5 Phase 5: Evaluation and Conclusion	14
1.4 Structure of the thesis.....	14
1.4.1 Chapter 2: The Application of Reasoning to Aerospace Integrated Vehicle Health Management (IVHM): Challenges and Opportunities	15
1.4.2 Chapter 3: Understanding the role of a Digital Twin in Integrated Vehicle Health Management (IVHM)	16
1.4.3 Chapter 4: Progress towards a Framework for Aerospace Vehicle Reasoning (FAVER).....	16
1.4.4 Chapter 5: Developing the architecture of FAVER (a Framework for Aerospace Vehicle Reasoning) using a System-of-Systems methodology	18
1.4.5 Chapter 6: A System-Level Failure Propagation Detectability Using ANFIS for an Aircraft Electrical Power System	19
1.4.6 Chapter 7: A Generalized Methodology for the Diagnosis of Aircraft Systems.....	20
1.4.7 Chapter 8: A Framework for Aerospace Vehicle Reasoning (FAVER).....	20
1.4.8 Chapter 9: Summary & Future works.....	21
1.5 List of Publications	21
References	22
2 The Application of Reasoning to Aerospace Integrated Vehicle Health Management (IVHM): Challenges and Opportunities.....	25
2.1 Introduction	26
2.1.1 Background	26

2.1.2 Need for Vehicle Level Health Monitoring	30
2.2 Reasoning – A Literature Review	31
2.2.1 Definitions.....	33
2.2.2 Reasoning Strategies.....	34
2.2.3 Reasoning System – A generalized architecture.....	36
2.2.4 Types of reasoning systems	37
2.3 Applications of Reasoning in Aerospace IVHM	49
2.4 Reasoning for IVHM at the Vehicle level	54
2.5 Summary and Conclusions.....	57
3 Understanding the role of a Digital Twin in Integrated Vehicle Health Management (IVHM).....	77
3.1 Introduction	78
3.2 Building Blocks of a Digital Twin	80
3.2.1 Product Lifecycle	81
3.2.2 Virtual model.....	82
3.2.3 The Connecting segment	82
3.2.4 Enabling Technologies.....	83
3.3 Applications of A Digital Twin	85
3.3.1 Digital Twin in design & development.....	85
3.3.2 Digital Twin in Manufacturing.....	86
3.3.3 Digital Twin in in-service monitoring	87
3.4 Digital Twin and IVHM	88
3.5 Challenges, Gaps & Opportunities	92
3.5.1 Challenges faced by Digital Twin (DT) in IVHM	92
3.5.2 Gaps & Opportunities	93
3.6 Conclusion.....	93
References	94
4 Progress towards a Framework for Aerospace Vehicle Reasoning (FAVER).....	98
4.1 Introduction	99
4.2 FAVER Fundamentals.....	100
4.2.1 Digital Twin (DT)	100
4.2.2 Reasoning.....	101
4.2.3 The synergy between the Digital Twin and Reasoning	101
4.3 FAVER Methodology	102
4.4 Development of FAVER	105
4.5 A Thought Experiment	108
4.6 Challenges	111
4.7 Conclusion & Future Work	112
5 Developing the architecture of FAVER (a Framework for Aerospace Vehicle Reasoning) using a System-of-Systems methodology.....	115
5.1 Introduction	116
5.1.1 Gaps in the current IVHM systems	119

5.1.2 FAVER – The Proposed framework	120
5.2 FAVER Architecture with A System-of-Systems Methodology	122
5.2.1 Platform FAVER requirements	125
5.2.2 Layer, Modules & Components.....	127
5.3 FAVER Architecture & Working Mechanism for Known Faults with Use cases	137
5.4 Discussion & Future Work.....	144
5.5 Summary	147
6 A System-Level Failure Propagation Detectability Using ANFIS for an Aircraft Electrical Power System.....	153
6.1 Introduction	154
6.2 Modelling and Simulation of an Aircraft Electrical Power System (EPS).....	156
6.2.1 The EPS Digital Twin	157
6.2.2 Fault Injection in the EPS Digital Twin	160
6.3 Diagnostic Reasoning for the EPS.....	166
6.3.1 Brief Description of ANFIS	167
6.3.2 An ANFIS Monitor for the EPS	169
6.4 Results and Discussion	170
6.4.1 Method 1: The Crisp Boundary Approach	170
6.4.2 Method 2: The Fuzzy Boundary Approach	172
6.5 Conclusions	176
7 A Generalized Methodology for the Diagnosis of Aircraft Systems.....	183
7.1 Introduction	184
7.1.1 FAVER: A Framework for Aerospace Vehicle Reasoning.....	185
7.1.2 The Need for A Generalized Methodology for The Diagnostic Process	186
7.2 The OSA-CBM Methodology	187
7.2.1 Contributions in this paper.....	189
7.3 Use Cases	190
7.3.1 The Engine	190
7.3.2 The Environmental Control System (ECS)	199
7.3.3 The Fuel System.....	205
7.4 Diagnosis.....	209
7.5 Summary and Future Work.....	211
8 FAVER - A Framework for Aerospace Vehicle Reasoning	218
8.1 Introduction	219
8.2 Building the Framework	221
8.2.1 Overview.....	221
8.2.2 Previous Work and Current Contribution.....	223
8.2.3 Assumptions	226
8.2.4 Architecture.....	226
8.3 Scalability and Flexibility of FAVER.....	229
8.3.1 Scalability: Expansion of FAVER	229

8.3.2 Flexibility in FAVER.....	233
8.4 Demonstration Use cases	237
8.4.1 New system (previously unknown): Single and Interaction faults isolation	238
8.4.2 Cases of multiple faults	241
8.5 Evaluation of FAVER	244
8.5.1 Pros and Cons of using FAVER's reasoning	244
8.5.2 Use cases demonstration with Neural Networks	247
8.6 Summary and Future work	250
References	250
9 Summary & Future works	254
9.1 Addressing the research aim and objectives.....	254
9.2 Contributions to knowledge	257
9.3 Future work	259
APPENDICES	262
Appendix A.....	262
Appendix B	263
Appendix C	269

LIST OF FIGURES

Figure 1-1 Cost breakdown for the United States airlines in 2018 (adapted from [1]) ..	1
Figure 1-2 System level interactions in an aircraft	3
Figure 1-3 Failure Propagation Visualization for Rotor Burst Case [11]	4
Figure 1-4 Health Information flow in a reasoning framework at vehicle level [20]	6
Figure 1-5 Methodology flow diagram	10
Figure 2-1 Overview of Chapter 2	25
Figure 2-2 Trend of AI technologies in Aerospace IVHM versus general AI technologies	27
Figure 2-3 Applications of Reasoning in Engineering	32
Figure 2-4 The architecture of a generalized reasoning system	36
Figure 2-5 Expert System Architecture [59]	39
Figure 2-6 Procedural Reasoning System (PRS) Architecture (adapted from [10])	41
Figure 2-7 Case Based Reasoning Architecture (adapted from [63])	42
Figure 2-8 SWOT Analysis for Model based vs data driven methods	48
Figure 2-9 Information flow in reasoning at vehicle level	56
Figure 3-1 Overview of Chapter 3	77
Figure 3-2 The role of IVHM in product lifecycle [7]	79
Figure 3-3 Digital Twin Ecosystem along the product lifecycle	81
Figure 3-4 Digital Twin in an IVHM framework (adapted from [5])	88
Figure 4-1 Overview of Chapter 4	99
Figure 4-2 The schematic of FAVER	103
Figure 4-3 Stages in developing FAVER	105
Figure 4-4 An example of a system fault with macro effect	106
Figure 4-5 Multiple systems in a fuel rig - A thought experiment	108
Figure 5-1 Overview of Chapter 5	115
Figure 5-2 OSA-CBM influenced hierarchical structure of Reasoners in an IVHM system	116
Figure 5-3 The working concept of FAVER (adapted from [28])	121
Figure 5-4 The architecture of FAVER	123

Figure 5-5 Layers and modules of FAVER developed with respect to V- diagram	124
Figure 5-6 Requirements and tests for FAVER platform.....	126
Figure 5-7 Requirements captured for layers in FAVER.....	128
Figure 5-8 Interactions between the DT facilitator and the Systems DT module	129
Figure 5-9 A generalized reasoning system architecture [9]	131
Figure 5-10 Traditional relationship matrix versus the combination of vehicle level relationship matrix with fault attributes database.....	133
Figure 5-11 Schematics of the aircraft systems used for FAVER demonstration	137
Figure 5-12 Activity diagram of FAVER to isolate known faults and ambiguity.....	140
Figure 5-13 Use case scenarios for FAVER demonstration	142
Figure 5-14 Results generated by the Central Facilitator in FAVER for the use case scenarios.....	143
Figure 6-1 Overview of chapter 6	153
Figure 6-2 MATLAB-Simulink model for the aircraft Electrical Power System (EPS)	158
Figure 6-3 Fault injection in circuit breakers connecting generators to other aircraft systems.....	161
Figure 6-4 Fault injection to DC loads via IGBT switches and noise signal added to voltage measurements	162
Figure 6-5 Fault injection in AC Pump motor by reduced voltage supply (F10).....	163
Figure 6-6 EPS performance during various fault scenarios (faults F1, F2...F9).	165
Figure 6-7 Pump motor characteristics for reduced voltage (F10)	165
Figure 6-8 (a) First order Takagi-Sugeno Fuzzy Reasoning (b) ANFIS structure (adapted from [30])	168
Figure 6-9 Healthy and faulty scenarios classified based on apparent power generated using Crisp boundary approach.....	170
Figure 6-10 Healthy scenario using the Crisp boundary approach	171
Figure 6-11 Faulty scenario using the Crisp boundary approach	172
Figure 6-12 Confusion matrix for the Crisp boundary approach.....	172
Figure 6-13 (a) Fuzzy boundaries representing Healthy and Faulty scenarios based on the Power generated, (b) Confusion matrix for Method 2 ANFIS rank processing with Fuzzy boundaries.....	174

Figure 6-14 (a) Corresponding reaction in apparent power for ACMotor_Fault Injection (F6) (b) Method 2 results with the ANFIS monitor and Causal reasoning.....	175
Figure 6-15 (a) Corresponding reaction in apparent power for DC_TRU_CBtrip (F5) fault injection (b) Method 2 results with the ANFIS monitor and Causal reasoning ...	176
Figure 6-16 (a) Diagnostics for one intermittent fault (F6) (b) Diagnostics of multiple intermittent faults (F5, F8)	177
Figure 7-1 Overview of Chapter 7	183
Figure 7-2 The schematic of FAVER	186
Figure 7-3 The Open Standard Architecture for Condition Based Maintenance (OSA-CBM) framework	187
Figure 7-4 Turbofan engine block model in T-MATS (adapted from [20]).....	191
Figure 7-5 Gross thrust and fuel flow rate in Healthy and faulty scenarios	193
Figure 7-6 Top 15 features selected by mRMR algorithm for state detection and diagnosis of the Engine	197
Figure 7-7 Performance of engine features for healthy and faulty scenarios	197
Figure 7-8 B737-800 ECS PACK model emulated in SESAC [26]	200
Figure 7-9 Temperature and Pressure profiles for healthy and faulty scenarios for ECS PACK simulation in SESAC	202
Figure 7-10 mRMR algorithm for the ECS features selection	204
Figure 7-11 Healthy and faulty case profiles for ECS features selected by mRMR algorithm	204
Figure 7-12 Layout of the fuel rig setup in IVHM Centre	206
Figure 7-13 Pressure profiles, pump speeds and flows for various faulty scenarios from fuel rig experiment.....	208
Figure 7-14 Confusion Matrices for best classification method for three aircraft systems	212
Figure 8-1 Overview of Chapter 8	218
Figure 8-2 Working schematic of FAVER	223
Figure 8-3 Previous work and current contribution	223
Figure 8-4 The architecture of FAVER.....	227
Figure 8-5 Main interactions between four aircraft systems.....	231
Figure 8-6 Activity diagram for expansion of FAVER to include two new aircraft systems	235

Figure 8-7 Activity diagram for expansion of FAVER to include new faults to the systems diagnostics	236
Figure 8-8 Use case scenario of fault originating in the EPS propagating to the FS and the engine	239
Figure 8-9 Use case scenario of fault originating from the engine and cascading to the ECS	241
Figure 8-10 Different multiple fault scenarios (some resulting in ambiguity) reasoned by FAVER	243
Figure 8-11 Updated knowledge scenario in FAVER	245
Figure 8-12 Assigning priorities in FAVER's reasoning	246
Figure 8-13 Cost of replacing traditional relationship matrix in FAVER.....	247
Figure 8-14 Neural network fitting to isolate faults at the vehicle level.....	248

LIST OF TABLES

Table 1-1 List of thesis chapters aligned with research objectives	14
Table 2-1 Reasoning Strategies	34
Table 2-2 Advantages, Disadvantages, and applications of Knowledge-based reasoning systems.....	38
Table 2-3 Reasoning in Aerospace Health Monitoring	49
Table 4-1 Use case Scenarios	106
Table 4-2 List of fault modes for the thought experiment.....	109
Table 5-1 Satisfied requirements of FAVER	145
Table 6-1 Measured values during the Healthy Run	159
Table 6-2 Faults injected in the EPS Digital Twin.....	163
Table 6-3 Diagnostic matrix for faults vs. tests.....	174
Table 7-1 Accuracy of classification by machine learning algorithms for diagnosis of faults in three aircraft systems	210
Table 7-2 Summary of the use cases demonstrated using OSA-CBM methodology ..	213
Table 8-1 Use case scenarios for neural network training.....	247

LIST OF EQUATIONS

(5-1).....132

(5-2).....132

(5-3).....133

(6-1).....157

(6-2).....157

(6-3).....167

(6-4).....167

(6-5).....167

(6-6).....168

(6-7).....168

(6-8).....169

(7-1).....193

(7-2).....195

(7-3).....195

(7-4).....195

(7-5).....195

LIST OF ABBREVIATIONS

ACM	Air Cycle Machine
ADMS	Aircraft Diagnostics and Maintenance Systems
AEPHM	Aircraft Electrical Power Systems Prognostics and Health Management
AI	Artificial Intelligence
ANFIS	Adaptive Neuro-Fuzzy Inference System
APU	Auxiliary Power Unit
BIT	Built-In Test
CBM	Condition Based Maintenance
CBR	Case Based Reasoning
CDP	Customer Discharge Pressure
CHX	Condenser
CMC	Centralized Maintenance Computers
CSP	Constraint Search Problem
DPV	Direct acting Proportional Valves
DSS	Decision Support System
DT	Digital Twin
ECS	Environmental Control System
EMA	Electro-Mechanical Actuator
EPS	Electrical Power System
FAA	Federal Aviation Administration
FAVER	A Framework for Aerospace Vehicle Reasoning
FOD	Foreign Object Damage
FS	Fuel System
GDE	General Diagnostic Engine
HIL	Hardware-in-the-loop
HPC	High Pressure Compressor
HPT	High Pressure Turbine
HPWS	High Pressure Water Separator
HVAC	Heating Ventilation and Air Conditioning
IATA	International Air Transport Association

IoT	Internet of Things
IVHM	Integrated Vehicle Health Management
kNN	k-Nearest Neighbour
LDA	Linear Discriminant Analysis
LPC	Low Pressure Compressor
LPT	Low Pressure Turbine
LRU	Line Replaceable Unit
mRMR	minimum Redundancy Maximum Relevancy
MRO	Maintenance, Repair, and Overhaul
NASA	National Aeronautics Space Administration
OEM	Original Equipment Manufacturer
OSA-CBM	Open Standard Architecture for Condition Based Maintenance
PACK	Passenger air conditioner
PBAS	pressure from the bleed air system
PHM	Prognostic Health Management
PHX	Primary Heat Exchanger
PRS	Procedural Reasoning System
RBR	Rule Based Reasoning
RCS	Reaction Control System
RHX	Reheater
RUL	Remaining Useful Life
SESAC	Simscape Environmental control system Simulation under All Conditions
SHX	Secondary Heat Exchanger
SoS	System-of-Systems
SVM	Support Vector Machine
SWOT	Strength Weakness Opportunities Threats
TCV	Temperature Control Valve
TFPG	Timed Failure Propagation Graph
TFSC	Thrust Specific Fuel Consumption
T-MATS	Toolbox for Modeling and Analysis of Thermodynamic Systems
V&V	Verification & Validation

LIST OF SYMBOLS

Electrical Power System

N_s	Synchronous speed of the generator
p	Apparent power
z	ANFIS output value

Engine

P_{t_AfCDP}	Pressure at the exit of bleed air duct
P_{t_S24}	Pressure at LPC exit (station 24)
T_{t_S5}	Temperature at LPT exit (station 5)
$\dot{m}$	mass flow rate
f	gross thrust
T_t	Temperature
P_t	Pressure
$\dot{W}_f$	mass flow

Fuel rig

P	Pump
R	Reservoir
S	Shut off Valve
V	Valve
F	Filter
$FM1$	Flowmeter 1
$FM2$	Flowmeter 2
N	Nozzle
$P1, .. P6$	Pressure sensors 1,2,..6
L	Laser sensor

Environmental Control System

T_i	Inlet temperature
T_o	Outlet temperature
T_{hi}	Hot inlet temperature
T_{ho}	Hot outlet temperature

Tci	Cold inlet temperature
Tco	Cold outlet temperature
PiC	Compressor inlet pressure
ToC	Outlet temperature of the compressor
TcoRHX	Reheater cold side outlet temperature
mRMR algorithm	
MI	Mutual Information
MIQ	Mutual Information Quotient
Px	Relevance of variable x
Qx	Redundancy of variable x

1 Introduction

A well-defined maintenance program is essential for warranting the safety and reliability of an aircraft, along with increasing its availability and reducing unexpected downtime. According to an Oliver Wyman report [1], in 2018, the maintenance costs incurred by the network and value airlines in the United States accounts for 8.95% as shown in [Figure 1-1](#). Globally, International Air Transport Association (IATA) [2] reports that Maintenance, Repair, and Overhaul (MRO) contributed to 9% of total operational costs for the airliners in 2018. The airliners are looking for methods to reduce these MRO related costs, and the prime emphasis is on unexpected downtime and extended maintenance leading to delays in the flights. For example, Southwest Airlines had to cancel several flights in 2019 due to unscheduled maintenance issues, costing the airlines millions a week [3]. This also reduces the trustworthiness of the airliners among the customers.

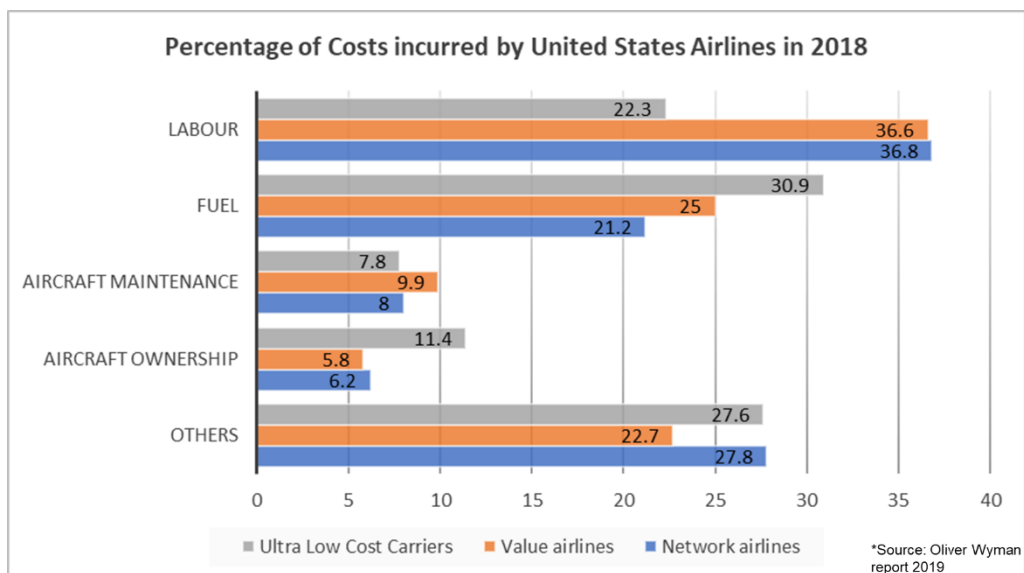


Figure 1-1 Cost breakdown for the United States airlines in 2018 (adapted from [1])

In order to save time and cost involved in MRO and to prevent unexpected downtime, maintenance programs for aircraft have evolved from scheduled maintenance and preventive maintenance to predictive maintenance and Condition Based Maintenance (CBM). The principal focus of CBM is to monitor the condition of the target systems, which could be either components or Line Replaceable Units (LRUs) in an aircraft system or the system itself, and provide maintenance on the need basis to avoid any unexpected

downtime. CBM is a real-time product performance measure, and it involves monitoring the current condition of the system to provide customized maintenance plans.

1.1 Motivation behind this research

Despite the dedicated maintenance practices of CBM to ensure safety, reliability, and availability of the aircraft, there are several cases of unexpected downtime due to complicated scenarios like fault propagation. In general, the cases of cascading faults are taken care of during troubleshooting activities in maintenance. However, in cases such as these accidents [4–8], where the faults propagate between the interacting aircraft systems, troubleshooting will not be straight forward, resulting in extended downtime. This problem forms the foundation of this research project.

1.1.1 Aircraft systems interactions

An aircraft is a complex machine comprising multi-physical systems, as listed by the Federal Aviation Administration (FAA) [9]. These systems interact with each other at different capacities for the safe and efficient operation of the aircraft. Some of the mainstream interactions between the aircraft systems include, but are not limited to the following types: i) mechanical and electrical interactions, ii) interactions enabled by mass transfer such as fuel and air, and iii) interactions due to data transfer such as sensor data and electronic commands [10]. These interactions between the systems can be seen in any typical aircraft; some of them are shown in [Figure 1-2](#).

- i) The main function of the Fuel System (FS) is to provide fuel to the engine and the Auxiliary Power Unit (APU).
- ii) The engine produces thrust demanded for the flight and also produces shaft speed required to drive the generators in the Electrical Power System (EPS), among other critical functions.
- iii) The APU also provides the mechanical power to drive the generators in the EPS.
- iv) The engine and the APU provide compressed bleed air to the Environmental Control System (ECS), which is then conditioned for pressurization in the aircraft cabin.
- v) The EPS provides AC and DC power required by all the other systems.

- vi) Aircraft electronics or avionics provides for Human-Machine Interface and Machine-Machine Interface for input and output, monitoring, and transferring of electronic data such as sensor data and control system commands.

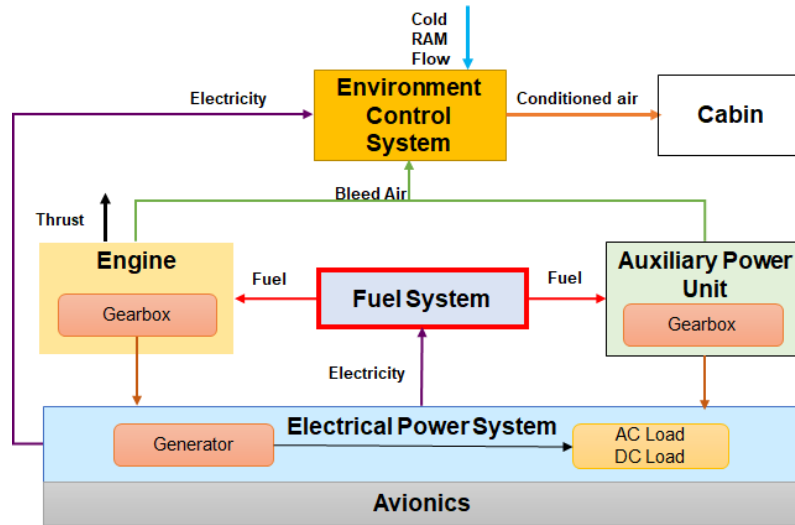


Figure 1-2 System level interactions in an aircraft

While most of the systems have interactions with few others, aircraft systems like the EPS and avionics span across the aircraft. All these systems are dependent on each other for achieving their functionality fully, as well as for the overall functioning of the aircraft. With the increasing trend of more-electric and all-electric aircraft, the modern aircraft is more integrated, making the interactions between these systems much more complex [10].

1.1.2 Failure propagation across systems

Aircraft systems are complex by themselves; they are designed and assessed to contain failures within the system in general to avoid fault propagation from one system to another during the interaction between these systems. However, with the advent of advanced technologies for high-speed data transfer, big data analysis, and the move towards lightweight and more electric concepts, the integration of major aircraft systems has become inevitable with increased interrelationship and interdependence between the systems. This has led to rigorous safety assessments during the design and development stages, followed by sophisticated maintenance programs during the operations stage, to reduce the probability of potential failures and their propagated effects.

1.1.3 Example of Safety assurance during Design & Development

SAE ARP4761 explains an approach to a complete system safety analysis required for aircraft to comply with safety requirements. This includes Common Cause Analysis, which is used to determine the potential common cause failure conditions through three elements, namely, Particular Risk Analysis, Zonal Safety Analysis, and Common Mode Analysis [11].

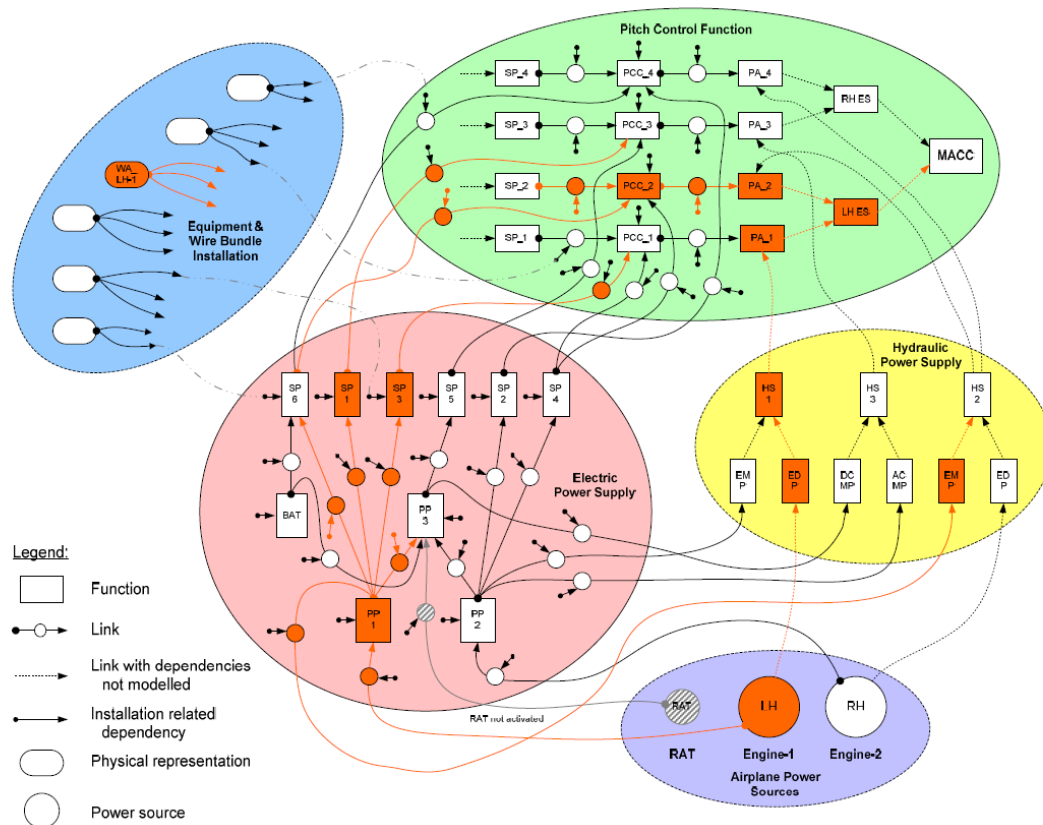


Figure 1-3 Failure Propagation Visualization for Rotor Burst Case [11]

For example, Fritz (2009) [11] uses System dependency analysis as a tool to perform the rotor burst analysis for the main engine for a redundant flight controls function (fly-by-wire pitch control) of a generic regional jet airliner. The failure propagation of a rotor burst scenario is seen in Figure 1-3, where the red/dark colored items indicate the propagation of lost/failed functions. For this analysis, the incident is assumed to start from Left Hand engine failure, which results in the loss of power in hydraulic and electric power in associated systems. As a result, in the electrical system, the engine driven generator is not powered. In the hydraulic system, the loss of hydraulic power leads to

the loss of power to an engine driven pump. Hence the electric motor pump is not able to provide the power demanded by the flight controls, finally affecting the Left Hand Elevator Surface (Figure 1-3). Wiring installation in the rotor burst zone is also considered for safety analysis. Rotor burst cases are developed from potential disc trajectories, and cases relevant to the failures assumed are simulated with their failure modes. Sensitivity analysis was carried out to capture the minimum acceptable control criterion and general requirements, using which the model was iterated during the optimization process to capture the functional margins of the considered systems [11].

Such rigorous safety analyses are essential for certification of the aircraft systems before they are used in operation. These design stage analyses are complemented by maintenance programs to ensure the safety, reliability, and availability metrics of the aircraft systems.

1.1.4 IVHM for Condition Based Maintenance

Integrated Vehicle Health Management (IVHM) systems use sophisticated technologies to acquire health data from the targeted systems, process the information to assess the current health and carry out diagnosis and prognosis, and provide inputs required by CBM for creating the maintenance plans [12]. IVHM can be integrated with the target system from its design stage itself; however, it is predominantly used as a product at the maintenance stage [13]. In recent years, health monitoring systems are used in many aircrafts, and they have reportedly contributed to huge cost savings in MRO [2]. This is possible due to accurate diagnosis and isolation of faults which help in the troubleshooting process and the improved calculation of remaining useful life of the systems, based on which maintenance activities are carried out. Even the faults propagating to nearby components or LRUs are caught during maintenance through troubleshooting with the help of built-in tests and through expert systems reference models used in IVHM systems.

1.1.4.1 Need for IVHM systems at the aircraft level

The current focus of IVHM systems involves automating the diagnostic procedures at the levels of LRUs, and for subsystems in the various aircraft systems (the engine, the ECS, etc.). Failure propagation among surrounding areas is considered in distributed diagnosis and is demonstrated in HVAC systems [14] and power distribution systems [15]. However, there are only a few studies in the IVHM systems that consider the

diagnosis of multiple systems at the aircraft level. Examples include a vehicle health monitoring system architecture patented by Honeywell [16], demonstration of the effect of a fault in a fuel-cooled oil cooler on other independent systems by Airbus Space and Defence [17], Vehicle Integrated Prognostic Reasoner [18], and Vehicle Level Reasoning System [19]. These show that the area of vehicle level health monitoring is still not sufficiently explored.

In order to trace the root cause of a cascaded fault, reasoning through the health information of multiple systems is a necessity, which has not been widely implemented. To fill this gap, this research considers developing a diagnostic reasoning framework at the vehicle level that can consider the health information from all levels, i.e., component, subsystem, and system (as seen in Figure 1-4), process the data at each level and give decisions based on all the inputs.

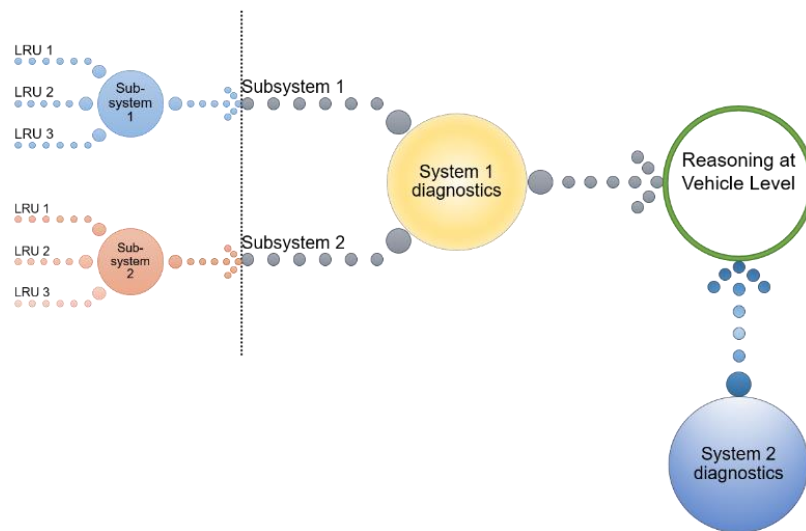


Figure 1-4 Health Information flow in a reasoning framework at vehicle level [20]

Developing a comprehensive diagnostic reasoner at the vehicle level for an aircraft within the PhD timeframe is difficult, if not impossible. Hence, rather than developing a system, this research sets up a framework that can be considered as an approach for the diagnosis of interaction faults at the aircraft level. The proposed framework (and hence the title of this thesis) is called A Framework for Aerospace Vehicle Reasoning, abbreviated as FAVER. The framework uses two different concepts, viz, i) a set of Digital Twins (DTs) representing the aircraft systems used to emulate the interactions with each

other, and ii) a reasoning mechanism to process information from multiple aircraft systems and isolate the root cause of cascaded faults.

This research project is funded by The Boeing company, which focuses on developing advanced IVHM systems like Aircraft Diagnostics and Maintenance Systems (ADMS) and integrated onboard and ground-based health reasoning systems to help predictive maintenance and CBM programs. This research aims to provide a simplified framework through FAVER, which could be used to develop IVHM systems at the aircraft level. FAVER could provide high level reasoning for troubleshooting cascading faults across systems and isolating their root causes, thus helping in preventing unexpected downtime and prolonged maintenance.

1.2 Aim and Objectives

Aim

To develop a Framework for Aerospace Vehicle Reasoning (FAVER) that incorporates a Digital Twin (DT) to demonstrate the interaction of multiple aircraft systems and a reasoning mechanism to investigate the scenarios of fault propagation between them.

Objectives

To achieve the aim of this research, a series of objectives are set out to be met. These objectives include finding existing gaps in the literature, the conceptualization of use cases, and setting up the framework for demonstration of use cases.

Objective 1:

A literature review to find the existing work and gaps in the following areas:

- a. Examples for fault propagation across aircraft systems
- b. Health monitoring techniques developed at systems and vehicle level
- c. Reasoning systems developed in academia and industries
- d. Digital Twin (DT) applications in the field of IVHM

Objective 2:

To choose multiple interacting aircraft systems from a legacy aircraft, define boundary conditions with sufficient complexity.

Objective 3:

To produce simulations of the chosen aircraft system DTs and to develop diagnostics for these systems to isolate a certain number of fault modes.

Objective 4:

To develop a reasoning module at the vehicle level that i) contains the knowledge of both the system level diagnostics and the relationship between the systems, ii) has a knowledge system that is easily scalable and updatable, and iii) possesses reasoning methodologies and algorithms to isolate single and interacting faults at the vehicle level

Objective 5:

To demonstrate the ability of the FAVER that combines the power of reasoning with the capabilities of a Digital Twin (DT) to deduce and isolate the system interaction faults and their root causes in the following cases:

- a. isolate faults that affect both local systems as well as have macro effects on the interacting systems,
- b. find ambiguity among the faults and resolve, and
- c. expand the framework to accommodate systems that are previously unknown

1.3 Methodology

The overall methodology is framed to set the context of the research, to define the aim and objectives, and to achieve these objectives within the prescribed duration. The research objectives are planned to be achieved in five phases, as shown in [Figure 1-5](#).

1.3.1 Phase 1: Literature review

This phase is to set up the context of this research and is in line with **Objective 1**. Hence, this phase focused on carrying out literature reviews in various topics to identify the research gaps. As seen in [Figure 1-5](#), the topics included aircraft accidents due to cascading faults, a survey on health monitoring systems, detailed reviews on reasoning techniques, and Digital Twin (DT) applications. As a result of literature reviews from phase 1, the following ‘**research gaps**’ were identified.

Research Gap 1: There are multiple occasions where aircraft accidents have been caused due to fault propagation between aircraft systems. This finding is significant, as it formed the backbone of the problem statement for this research.

Research Gap 2: Health monitoring systems are well-established at components/ LRUs and subsystems level, and they are now increasingly used at the systems level. However, there is not much research done with the ‘vehicle level’ as the key focus.

Research Gap 3: Most DT applications in the industry contribute to aircraft systems health management; they are not widely used for emulating interactions between aircraft systems, and they are not applied in vehicle level health monitoring.

Outcome of Phase 1: This phase helped in defining the aim of this research. As the outcome of this phase of research, one journal paper, and one conference paper were produced.

- i. Ezhilarasu, C. M., Skaf, Z., & Jennions, I. K. (2019). *The application of reasoning to aerospace Integrated Vehicle Health Management (IVHM): Challenges and opportunities*.

Progress in Aerospace Sciences, 105(January), 60–73.

<https://doi.org/10.1016/j.paerosci.2019.01.001>

- ii. Ezhilarasu, C.M., Skaf, Z, Jennions, I.K., 2019, *Understanding the Role of a Digital Twin in Integrated Vehicle Health Management (IVHM)*, IEEE SMC 2019, Bari, Italy, Oct 6-9, 2019 DOI: 10.1109/SMC.2019.8914244

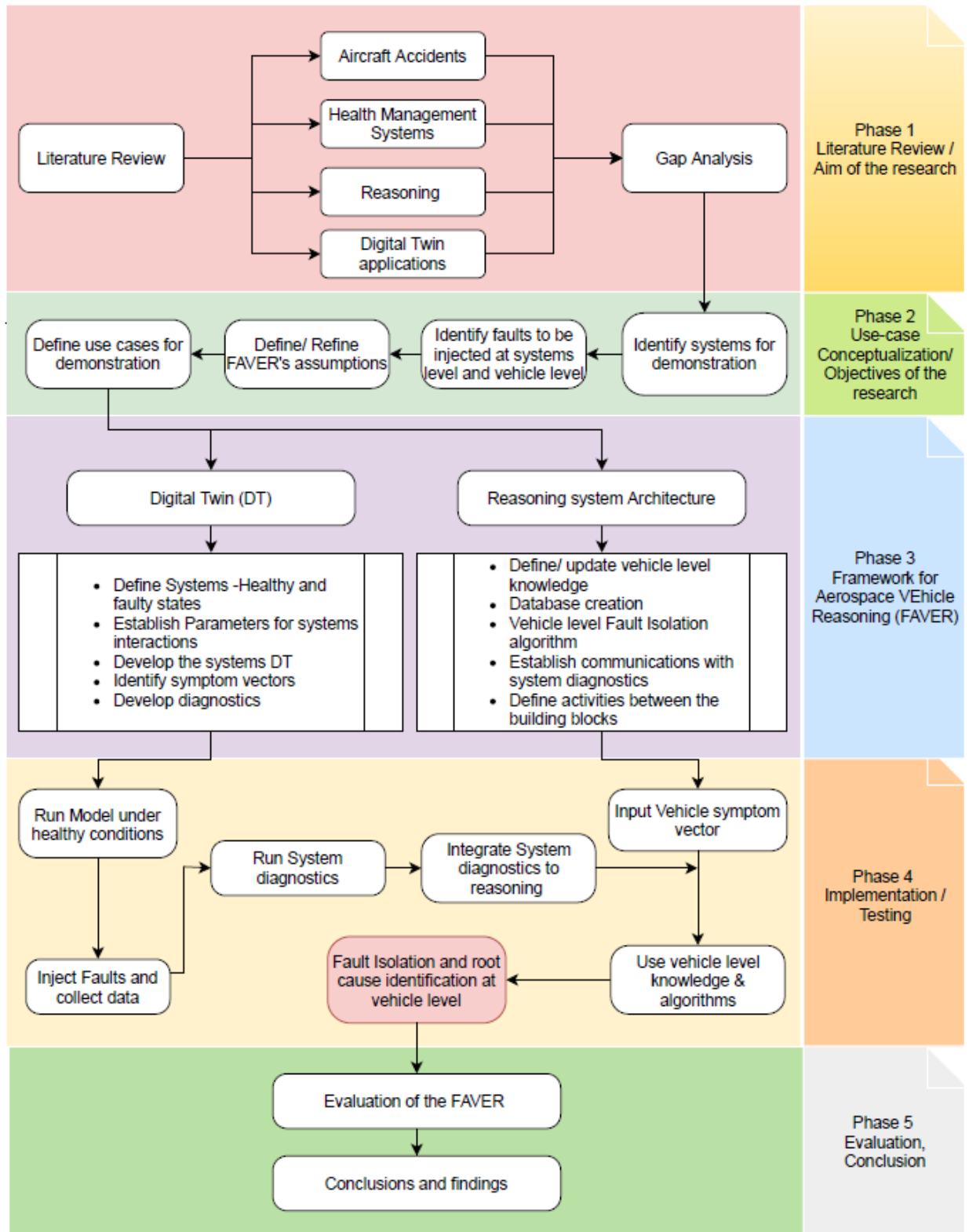


Figure 1-5 Methodology flow diagram

Apart from the above-mentioned topics, frequent literature reviews were conducted throughout the research duration in order to implement the best-known methods at every stage. The other research gaps found during this exercise are presented in their respective phases of the methodology.

1.3.2 Phase 2: Use case conceptualization

This phase focused on conceptualizing the framework, which involved choosing the systems for demonstrations, developing use cases, and defining the boundaries (as seen in [Figure 1-5](#)). This phase is significant as it is in line with **Objective 2** and helped **define the rest of the objectives of this research**, to achieve the aim of demonstrating isolation of cascading faults and their root causes at the vehicle level within the limited duration.

Outcome of Phase 2: A **thought experiment** was conducted during this phase, which helped in framing the assumptions to be held for developing FAVER at the initial stage. These assumptions were made to help kickstart the demonstration of the framework in a reasonable time period, and they are (will be) removed or conservatively updated as the research progresses. As a result of this phase, four interacting aircraft systems were chosen for the demonstration of FAVER. These systems are

- i) The Electrical Power System (EPS)
- ii) The Fuel System (FS)
- iii) The Engine
- iv) The Environmental Control System (ECS)

These four systems were chosen carefully based on the availability of resources for their respective DTs. The use cases were developed based on the possible interactions between these systems as well as the faults which can be simulated for propagation between them. As the outcome of this phase of research, one extended abstract and one conference paper were produced.

- i. Ezhilarasu, C. M. (2018). *An Integrated Reasoning Framework for Vehicle Level Diagnosis of Aircraft Subsystem Faults*. Annual Conference of Prognostics and Health Management Society.
Retrieved from <https://www.phmpapers.org/index.php/phmconf/article/view/707>

- ii. Ezhilarasu, C. M., Skaf, Z., & Jennions, I. K. (2019). *Progress towards a Framework for Aerospace Vehicle Reasoning (FAVER)*. Proceedings of the Annual Conference of the PHM Society, 11(1), 1–9.

Retrieved from <https://doi.org/https://doi.org/10.36001/phmconf.2019.v11i1.887>

1.3.3 Phase 3: Framework for Aerospace Vehicle Reasoning (FAVER)

This phase involved developing the building blocks of FAVER in order to meet **Objectives 3 and 4**. The key outcome of this phase is the **architecture of FAVER**. Here, FAVER is treated as a platform for setting it up for demonstration. A System-of-systems methodology is followed to capture the requirements for developing the architecture of FAVER, and a ‘V’ diagram is developed to systemically build the modules required for FAVER. Accordingly, as observed from [Figure 1-5](#), the Digital Twins (DTs) of the chosen aircraft systems are aggregated by developing a simulation model from scratch, using the open source as well as copyrighted simulation models, and running an experimental rig to include hardware-in-the-loop. This is followed by building stand-alone diagnostics for each of these systems. In parallel, the blocks required for the reasoning system are developed, including the vehicle level knowledge and reasoning algorithms. While developing the DTs and the system level diagnostics for the aircraft systems, the following research gaps were found and addressed.

Research Gap 4: Most diagnostics developed for the aircraft Electrical Power Systems (EPS) focused on its components and LRUs like wires, relays, generators, and induction motors. Researches were also conducted to analyze the performance of aircraft systems linked to the EPS, such as the fuel pump and the engine. However, there is a lack of research work focusing on the isolation of faults in the EPS that could have cascading effects on the connected aircraft systems.

Research Gap 5: Across the literature, diagnostics developed for aircraft systems follow different sets of instructions. For a vehicle level reasoner such as FAVER, building a system level diagnostic for each system using different steps as found in the researches, is time-consuming. Even the established methodologies in the literature are experimented exclusively on only one of the aircraft systems. There is no standardized methodology that is demonstrated with multiple aircraft systems.

Outcome of Phase 3: As a result of phase 3, the architecture of FAVER is defined, and its building blocks are constructed. The DTs for all four aircraft systems are set up, along with their diagnostics. The reasoning system is also implemented as a part of FAVER's architecture, including vehicle level knowledge and reasoning algorithms.

Research gap 4 is addressed by developing an EPS DT along with its interaction network extending to the components and LRUs from other aircraft systems like the engine, the FS, and the ECS. Faults are injected to observe both micro effects (i.e., within the systems) and macro effects on the interaction network. An Adaptive Neuro-Fuzzy Inference System (ANFIS), combined with causal reasoning, was implemented for the diagnosis of faults in the EPS, including the isolation of faults that could affect the connected systems.

Research gap 5 is addressed in this research by adopting the Open Standard Architecture for Condition Based Maintenance (OSA-CBM) standard as the methodology for developing system level diagnostics. For demonstration purposes, three aircraft systems, viz., the engine, the Fuel System (FS), and the Environmental Control System (ECS) are used for developing diagnostics using the OSA-CBM methodology.

As a result of this phase, the paper on the EPS simulation and diagnostics is published in a journal, the paper on OSA-CBM methodology for system diagnostics is under review with a journal, and the paper on architecture of FAVER is under revision for another journal.

- i. Ezhilarasu CM., Jennions IK. *A System-Level Failure Propagation Detectability Using ANFIS for an Aircraft Electrical Power System*. Applied Sciences. 2020; 10(8): 2854. Available at: <https://doi.org/10.3390/app10082854>
- ii. *A Generalized Methodology for Diagnosis of Multiple Aircraft Systems* – In revision, with IEEE Access
- iii. *Developing the architecture of FAVER (a Framework for Aerospace Vehicle Reasoning) using System-of-Systems methodology* – In revision, with IJPHM

1.3.4 Phase 4: Implementation and Testing

This phase covers **objective 5**, where the blocks built for FAVER are put together for demonstration. A couple of accident scenarios are adapted for developing use cases for

interaction faults involving multiple aircraft systems. The developed reasoning technique for FAVER is tested through several complicated scenarios to test its ability to isolate cascading faults, solving ambiguities, and identifying the root causes. In this phase, FAVER is also tested for its ability to accommodate new (or previously unknown) systems and update its existing knowledge in the framework. Phases 3 and 4 are repeated to create and add new aircraft systems to the framework.

1.3.5 Phase 5: Evaluation and Conclusion

In this phase of research, FAVER is evaluated for its proof of concept, along with a comparison study for its reasoning abilities. Research areas to be focused in the future, as an extension of the current research is also explored in this phase.

1.4 Structure of the thesis

This thesis is written in a ‘Paper-based’ format, consisting of nine chapters. Except for chapter 8 (and the introduction and conclusion chapters), all the other chapters are peer-reviewed. Chapters 2, 3, 4, 6, and 7 are reformatted versions of published papers (both journal and conference papers) with only a few changes to the main text, to reflect the updated journey of this research. Chapters 5 and 8 are papers currently formatted for submission to a journal. All these chapters are outcomes of different phases of research methodology, as mentioned in section 1.3, and are arranged cohesively to align with the research objectives, as listed in [Table 1-1](#).

Table 1-1 List of thesis chapters aligned with research objectives

Chapter	Outcomes of Research methodology phase	Objectives achieved	Publication status
Chapter 2	Phase 1	Objective 1	Published in journal
Chapter 3	Phase 1	Objective 1	Presented in conference
Chapter 4	Phase 2	Objective 2	Presented in conference
Chapter 5	Phase 3	Objective 4	In preparation for submission
Chapter 6	Phase 3	Objective 3	Published in journal
Chapter 7	Phase 3	Objective 3	Published in journal

Chapter 8	Phase 4, Phase 5	Objective 5	In preparation for submission
------------------	------------------	-------------	-------------------------------

The following subsections provide a concise review of the content and outcomes of each chapter in this thesis.

1.4.1 Chapter 2: The Application of Reasoning to Aerospace Integrated Vehicle Health Management (IVHM): Challenges and Opportunities

This chapter is an outcome of **Phase 1**, and it aligns with **Objective 1**. It reviews various reasoning methods applied in the field of aerospace for IVHM, as well as explores its relationship with respect to their timeline in the field of artificial intelligence. This chapter presents the basics of reasoning, drawing examples from different fields ranging from the gaming industry to healthcare and education. It discusses, in detail, the different reasoning techniques, methodologies, and systems, along with their pros and cons. It also reviews different algorithms used by reasoning for the purpose of diagnosis and prognosis, which are the cornerstones of IVHM. This chapter also presents various reasoners developed in aerospace IVHM by the industry as well as academia.

Outcomes of this chapter:

- i) This chapter identifies a significant gap in the current IVHM systems: the lack of research and application of IVHM systems at the vehicle level (**Research gap 2** from section 1.3). This, in turn, serves as the underlying problem statement on which this research is built.
- ii) A generalized reasoning system derived in this chapter by comparing different reasoning systems is used to develop the reasoning architecture in FAVER at the vehicle level.
- iii) The reasoning techniques surveyed are applied in the system level diagnostics as well as in the vehicle level reasoning algorithm.

This chapter is a reformatted version of the following publication:

Ezhilarasu, C. M., Skaf, Z., & Jennions, I. K. (2019). *The application of reasoning to aerospace Integrated Vehicle Health Management (IVHM): Challenges and opportunities*. Progress in Aerospace Sciences, 105(January), 60–73.
<https://doi.org/10.1016/j.paerosci.2019.01.001>

1.4.2 Chapter 3: Understanding the role of a Digital Twin in Integrated Vehicle Health Management (IVHM)

This chapter is also an outcome of **Phase 1**, aligning with **Objective 1**. This chapter studies the structure of a Digital Twin (DT), aggregates its various forms along the life cycle of a product, and discusses the building blocks and enabling technologies of a DT. This chapter reviews the applications of a DT in various industries and academia and focuses especially, in the field of IVHM.

Outcome of this chapter:

This chapter derives the role of a DT in an OSA-CBM influenced IVHM system, which is adapted in this research for the DT layer in FAVER.

This chapter is a reformatted version of the following conference paper:

Ezhilarasu, C.M., Skaf, Z, Jennions, I.K., 2019, *Understanding the Role of a Digital Twin in Integrated Vehicle Health Management (IVHM)*, IEEE SMC 2019, Bari, Italy, Oct 6-9, 2019 DOI: 10.1109/SMC.2019.8914244

1.4.3 Chapter 4: Progress towards a Framework for Aerospace Vehicle Reasoning (FAVER)

This chapter is a result of **Phase 2**, which focuses on the use case conceptualization for the framework. The framework FAVER is proposed in this chapter, along with the supporting background details. This chapter presents the preliminary objectives of the framework and the methodology to be followed to demonstrate FAVER. This is followed by developing example use cases and deriving assumptions through a thought experiment. Challenges in implementing FAVER is also discussed in this chapter and a suitable solution, provided.

Outcomes of this chapter:

The following are the outcomes of this chapter.

- i) The backstory of FAVER along with preliminary objectives
- ii) Methodology to be followed for implementing and demonstrating FAVER
- iii) Possible combinations of use cases
- iv) Assumptions arrived at with the help of a thought experiment

This chapter is a reformatted version of the following conference paper:

Ezhilarasu, C. M., Skaf, Z., & Jennions, I. K. (2019). *Progress towards a Framework for Aerospace Vehicle Reasoning (FAVER)*. Proceedings of the Annual Conference of the PHM Society, 11(1), 1–9. Retrieved from <https://doi.org/https://doi.org/10.36001/phmconf.2019.v11i1.887>

Note:

The objectives and methodology of FAVER described in this chapter are only preliminary. They are later finetuned as the research progressed. Some of the assumptions made in this chapter were updated too. Chapter 4 has not been edited from its published version in order to cover the exact path followed in this research. The difference between the concepts provided in chapter 4 and the rest of this research are listed below, with the justifications on why these concepts were not used.

- i) ***FAVER should identify the faults that were previously unknown:*** In the preliminary phase of the conceptualization of FAVER, it was decided that FAVER should be able to identify the faults that are not known to its reasoning knowledge. This point is discussed in chapter 4 in detail. In order to demonstrate this, the EPS DT was assumed to be not instrumented for diagnostics. This assumption, however, was found to be a contradiction, as FAVER's vehicle level reasoning is built upon system level diagnostics. Further, in practice, any aircraft system with sensor data would have some sort of diagnostics in place. Hence, assuming that the EPS will not be instrumented for diagnostics at first, only to be built at the later stage when FAVER throws an error for not having enough information about the EPS, proved to be a futile exercise. Therefore, the assumption made in this chapter that *the EPS will not be instrumented for diagnostics* was removed for the next chapter of this thesis. Also, instead of focusing on making FAVER identify a fault that was previously unknown, the research progressed in making FAVER expandable to accommodate an aircraft system that was previously unknown to the framework, thus making FAVER expandable and scalable.
- ii) ***Case Based Reasoning (CBR) system for FAVER's vehicle level reasoning:*** In chapter 4, the vehicle level knowledge of FAVER was assumed to be made of a CBR system and hence details were given on how the symptom vectors (a set of

parameters representing the health of an aircraft system) will be populated to create the database, against which the input symptom vector will be compared. This method was dropped at the later stage in favour of an expert system at the vehicle level. This is because a CBR system is only as good as its data repository, and for this research, creating such a repository meant access requirements to several datasets, along with the time involved in data processing and other supporting efforts. On the other hand, the novel expert system developed for FAVER has the advantage of being easily manageable and scalable. This expert system can be straightforwardly updated with additional knowledge, without much changes to the existing structure. Hence, from chapter 5 onwards, the vehicle level knowledge of FAVER represents the expert system rather than a CBR system.

1.4.4 Chapter 5: Developing the architecture of FAVER (a Framework for Aerospace Vehicle Reasoning) using a System-of-Systems methodology

This chapter is a result of **Phase 3**, aligning with **Objective 4**. In this chapter, the architecture of FAVER is developed using a System-of-Systems methodology; the different requirements are captured using the Holistic Requirements model, and a tailored ‘V’ process model is developed for this methodology. This chapter also discusses how the challenges like scalability are handled, with the help of modularity as well as a novel representation of the expert system’s knowledge, as against the traditional relationship matrix. This chapter presents the activities followed by FAVER’s architecture and demonstrates its function with use case scenarios. The use cases demonstration includes results from simulation models and diagnostic results which are described in Chapters 6 and 7.

Outcomes of this chapter:

The following are the outcomes of this chapter.

- i) Definition and implementation of the building blocks of FAVER
- ii) Expert system with fault attributes database for reasoning’s knowledge
- iii) Description of activities between the modules of FAVER

- iv) Demonstration of isolation of cascading faults and their root causes with the EPS and the FS, solving ambiguity in the process

This chapter is a reformatted version of the following paper, which is **in preparation for journal submission**.

1.4.5 Chapter 6: A System-Level Failure Propagation Detectability Using ANFIS for an Aircraft Electrical Power System

This chapter aligns with **Objective 3**, and it is an outcome of *Phase 3*. It demonstrates the development of Electrical Power System (EPS) Digital Twin (DT) focusing mainly on its connections with other aircraft systems in its interaction network, like the engine, the Fuel System (FS), and the Environmental Control System (ECS), through LRUs like valves and pumps. This is followed by simulations of both healthy and faulty scenarios using the developed DT. An ANFIS monitor is developed to detect the presence of any fault (continuous or intermittent). The results of ANFIS is processed in two approaches, and causal reasoning is combined with the approach that provides better classification results. Later, the ANFIS monitor is tested for its ability to detect faults and isolate their root causes. This EPS DT is used for simulating fault scenarios that affect the systems in its interaction network, and the results from the ANFIS monitor are used for vehicle level reasoning in FAVER.

Outcomes of this chapter:

The following are the outcomes of this chapter.

- v) The EPS DT with connections established with the engine, the FS, and the ECS
- vi) System level diagnostics using ANFIS and causal reasoning

This chapter is a reformatted version of the following journal paper:

Ezhilarasu CM., Jennions IK. *A System-Level Failure Propagation Detectability Using ANFIS for an Aircraft Electrical Power System*. Applied Sciences. 2020; 10(8): 2854. Available at: <https://doi.org/10.3390/app10082854>

1.4.6 Chapter 7: A Generalized Methodology for the Diagnosis of Aircraft Systems

Similar to chapter 6, this chapter is also an outcome of **Phase 3**, aligning with **Objective 3**. This chapter demonstrates the development of diagnosis for three aircraft systems: the engine, the ECS, and the FS, through a streamlined methodology based upon OSA-CBM. This paper applies mRMR (minimum redundancy maximum relevance) algorithm for the feature selection process, and the resulting symptom vectors are used in the diagnostic functions for fault isolation. The diagnostic results from each system are used for vehicle level reasoning in FAVER.

Outcomes of this chapter:

The following are the outcomes of this chapter.

- i) The engine DT with connections to the FS, the EPS, and the ECS
- ii) The ECS DT with connections to the engine and the EPS
- iii) The FS DT with connections to the engine and the EPS
- iv) System level diagnostics for all three aircraft systems, with higher accuracy of classification.

This chapter is a reformatted version of the following journal paper:

C. M. Ezhilarasu, Z. Skaf and I. K. Jennions, "*A Generalised Methodology for the Diagnosis of Aircraft Systems*," in IEEE Access, vol. 9, pp. 11437-11454, 2021, doi: 10.1109/ACCESS.2021.3050877.

1.4.7 Chapter 8: A Framework for Aerospace Vehicle Reasoning (FAVER)

This chapter pulls all the previous chapters together, for implementing the framework in full scale. This is an outcome of **Phase 4 and Phase 5** and aligns with **Objective 5**. This chapter demonstrates FAVER's important properties, such as flexibility and scalability. A couple of aircraft accident scenarios are adapted for the demonstration of isolating cascading faults and their root causes. Several other complicated use cases are also demonstrated to showcase FAVER's ability to resolve ambiguities. Also, in this chapter, FAVER is evaluated for its reasoning ability as well as its proof of concept.

Outcomes of this chapter:

The following are the outcomes of this chapter.

- i) Flexibility and scalability demonstrations of FAVER, including the expansion of knowledge.
- ii) Demonstration of FAVER's ability to isolate interacting faults, their root causes, and possible cascading effects
- iii) Demonstration of FAVER's ability to resolve ambiguous scenarios
- iv) Evaluation of FAVER

This chapter is **in preparation** for submission to a journal.

1.4.8 Chapter 9: Summary & Future works

This chapter summarizes the thesis, highlights the overall contributions of this research, and discusses the future direction of FAVER.

1.5 List of Publications

Journal Papers:

- i. Ezhilarasu, C. M., Skaf, Z., & Jennions, I. K. (2019). *The application of reasoning to aerospace Integrated Vehicle Health Management (IVHM): Challenges and opportunities*. Progress in Aerospace Sciences, 105(January), 60–73.
<https://doi.org/10.1016/j.paerosci.2019.01.001>
- ii. Ezhilarasu CM., Jennions IK. *A System-Level Failure Propagation Detectability Using ANFIS for an Aircraft Electrical Power System*. Applied Sciences. 2020; 10(8): 2854. Available at: <https://doi.org/10.3390/app10082854>
- iii. C. M. Ezhilarasu, Z. Skaf and I. K. Jennions, "A Generalised Methodology for the Diagnosis of Aircraft Systems," in IEEE Access, vol. 9, pp. 11437-11454, 2021, doi: 10.1109/ACCESS.2021.3050877.

Conference Papers:

- i. Ezhilarasu, C.M., Skaf, Z, Jennions, I.K., 2019, *Understanding the Role of a Digital Twin in Integrated Vehicle Health Management (IVHM)*, IEEE SMC 2019, Bari, Italy, Oct 6-9, 2019 DOI: 10.1109/SMC.2019.8914244

- ii. Ezhilarasu, C. M., Skaf, Z., & Jennions, I. K. (2019). *Progress towards a Framework for Aerospace Vehicle Reasoning (FAVER)*. Proceedings of the Annual Conference of the PHM Society, 11(1), 1–9. Retrieved from <https://doi.org/https://doi.org/10.36001/phmconf.2019.v11i1.887>

Paper in preparation:

- i. Developing the architecture of FAVER (a Framework for Aerospace Vehicle Reasoning) using System-of-Systems methodology
- ii. A Framework for Aerospace Vehicle Reasoning (FAVER)

References

1. Stalnaker T., Alport G., Buchanan A., Taylor A. Airline Economic Analysis. 2019. Available at: https://www.oliverwyman.com/content/dam/oliverwyman/v2/publications/2019/apr/APRIL262019_Airline_Economic_Analysis_2018-2019vFweb.pdf
2. IATA's Maintenance Cost Technical Group. Airline Maintenance Cost: Executive Commentary Edition 2019 (FY2018 data). International Air Transport Association. 2019. Available at: <https://www.iata.org/contentassets/bf8ca67c8bcd4358b3d004b0d6d0916f/mctg-fy2018-report-public.pdf>
3. Joseph L. Southwest's maintenance problems are costing it millions a week, CEO says. CNBC. March 2019; Available at: <https://www.cnbc.com/2019/03/05/southwests-maintenance-problems-costing-it-millions-a-week-ceo-says.html>
4. Conradi K. AAIB Annual Safety Report 2015. 2015. Available at: https://assets.publishing.service.gov.uk/media/5656b9e2ed915d0367000008/Annual_Safety_Report_2015.pdf
5. South African Civil Aviation Authority. Aircraft Incident Report. 2008. Available at: <https://www.skybrary.aero/bookshelf/books/827.pdf>
6. Sleight PA., Carter RDG. AIRCRAFT ACCIDENT REPORT 1/2010. 2014. Available at: DOI:<https://doi.org/EW/C2008/01/01>

7. Air Accidents Investigation Branch. AAIB Bulletin: 4/2014 9M-MPL EW/C2012/08/04. 2014. Available at: <https://www.gov.uk/aaib-reports/boeing-747-4h6-9m-mpl-17-august-2012>
8. Air Accidents Investigation Branch. Aircraft Accident Report No: 4/2009. Available at: https://assets.publishing.service.gov.uk/media/552659a7e5274a141800000d/Summary_AAR_4-2009_Airbus_A319-111__G-EZAC_09-09.pdf
9. Mike Monroney Aeronautical Center. Federal Aviation Administration Joint Aircraft System/Component Code and Table Definitions. Federal Aviation Administration. 2008. Available at: https://av-info.faa.gov/sdrx/documents/JASC_Code.pdf (Accessed: 12 September 2020)
10. Seabridge A. Integration and Complexity. Encyclopedia of Aerospace Engineering. Chichester, UK: John Wiley & Sons, Ltd; 2010. Available at: DOI:10.1002/9780470686652.eae481
11. Fritz K. System Dependency Analysis Supporting Common Cause Analyses for Complex Aircraft Systems. SAE International Journal of Aerospace. 2009; 2(1): 28–39.
12. Jennions IK. Introduction. In: Jennions IK (ed.) Integrated Vehicle Health Management: Perspectives on an Emerging Field. SAE International; 2011. pp. 1–8. Available at: DOI:10.4271/R-05
13. Jennions IK., Esperon-Miguez M., Niculita O. Integrating IVHM and asset design. International Journal of Prognostics and Health Management. 2016; 7(2): 1–16.
14. Papadopoulos PM., Reppa V., Polycarpou MM., Panayiotou CG. Distributed Diagnosis of Actuator and Sensor Faults in HVAC Systems. IFAC-PapersOnLine. Elsevier B.V.; July 2017; 50(1): 4209–4215. Available at: DOI:10.1016/j.ifacol.2017.08.816
15. Daigle M., Bregon A., Roychoudhury I. Qualitative Event-based Diagnosis with Possible Conflicts Applied to Spacecraft Power Distribution Systems*. IFAC Proceedings Volumes. IFAC; January 2012; 45(20): 265–270. Available at:

DOI:10.3182/20120829-3-MX-2028.00084

16. Nwadiogbu EO., Mylaraswamy D., Menon S., Voges HC., Hadden G. US008346429B2: Vehicle Health Monitoring System Architecture for Diagnostics and Prognostics Disclosure. United States Patent; 2013. Available at: DOI:10.1016/j.(73)
17. López BS., Siddiolo AM., Adhikari PP., Buderath M. A Bayesian paradigm for aircraft operational capability assessment and improved fault diagnostics. European Conference of the Prognostics and Health Management Society. 2016. pp. 1–14.
18. Bharadwaj, Raj; Mylaraswamy, Dinkar; Cornhill, Dennis; Biswas G., Koutsoukos, Xenofon; Mack D. Vehicle Integrated Prognostic Reasoner (VIPR) Final Report. NASA Langley Research Centre; 2013. Available at: <https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20130011181.pdf>
19. Khan F., Eker OF., Sreenuch T., Tsourdos A. Health Index and Behaviour Based Vehicle Level Reasoning System. Conference on Prognostics and Health Management (PHM). IEEE; 2015. Available at: DOI:10.1109/ICPHM.2015.7245042
20. Ezhilarasu CM., Skaf Z., Jennions IK. The application of reasoning to aerospace Integrated Vehicle Health Management (IVHM): Challenges and opportunities. Progress in Aerospace Sciences. Elsevier Ltd; 2019; 105(January): 60–73. Available at: DOI:10.1016/j.paerosci.2019.01.001

2 The Application of Reasoning to Aerospace Integrated Vehicle Health Management (IVHM): Challenges and Opportunities

This paper aims to discuss the importance and the necessity of reasoning applications in the field of Aerospace Integrated Vehicle Health Management (IVHM). A fully functional IVHM system is required to optimize Condition Based Maintenance (CBM), avoid unplanned maintenance activities and reduce the costs inflicted thereupon. This IVHM system should be able to utilize the information from multiple subsystems of the vehicle to assess the health of those subsystems, their effect on the other systems, and on the vehicle as a whole. Such a system can only be realized when the supporting technologies like sensor technology, control and systems engineering, communications technology and Artificial Intelligence (AI) are equally advanced. This paper focuses on the field of AI, especially reasoning technology and explores how it has helped the growth of IVHM in the past. The paper reviews various reasoning strategies, different reasoning systems, their architectures, components and finally their numerous applications. The paper discusses the shortcomings found in the IVHM field, particularly in the area of vehicle level health monitoring and how reasoning can be applied to address some of them. It also highlights the challenges faced when the reasoning system is developed to monitor the health at the vehicle level and how a few of these challenges can be mitigated.

2.1	Introduction
2.2	Reasoning - A literature review
2.3	Applications of reasoning in aerospace IVHM
2.4	Reasoning for IVHM at the vehicle level
2.5	Summary and conclusions

Figure 2-1 Overview of Chapter 2

2.1 Introduction

A study in 2018 by International Air Transport Association (IATA)'s Maintenance Cost Task Force documents that the aerospace industry spent \$76 billion for Maintenance Repair and Overhaul (MRO) of commercial aircraft in the financial year 2017 and this is expected to go up to \$118 billion by 2027 [1]. It is a well-known fact that the MRO costs make up to 9% of the overall operation costs of an airline. Owing to the increased autonomy and the resulting complexity in aircraft systems, industries are investing heavily in health management systems to improve their Condition-based Maintenance (CBM) programs. This would help in increasing the availability and dispatch reliability of the aircraft, reduce the MRO cost significantly and avoid any unplanned downtime or accidents. Integrated Vehicle Health Management (IVHM) is a technology that offers a paradigm shift in support of CBM. IVHM was initially introduced by the National Aeronautics Space Administration (NASA) in 1992, as a technology to collect data, diagnose, predict, mitigate the faults, and support the operational decisions and post-operational maintenance activities of space vehicles [2]. Ever since, IVHM has been expanded to other vehicles like aircraft, ships, and automobiles. The current version of IVHM encompasses many roles throughout the entire lifecycle of a vehicle as a product and process, from the beginning idea or business proposition, going through design, development, testing, analysis, until the long last stage of after-sales service [3]. IVHM aims to ensure that the host system functions as it is intended without failure, thus increasing the availability of the systems and reducing the cost and time involved due to unplanned maintenance activities. To achieve this, IVHM aids the use of data from the host systems, not only for the purpose of diagnosis and prognosis that help CBM but also for optimizing the troubleshooting activities. IVHM makes use of emerging technologies in the fields of sensor technology, systems and control engineering, communications technology and Artificial Intelligence [2]. This paper focuses on the growth and the challenges in aerospace IVHM with respect to the field of AI, particularly the technology of reasoning.

2.1.1 Background

This section presents the growth of aerospace health monitoring which includes both spacecraft and aircraft, with respect to AI technologies. [Figure 2-2](#) shows the timeline

comparison of widely used technologies in the field of AI since the 1980s versus the AI technologies used in aerospace IVHM in that particular timeframe. In Figure 2-2, the technologies in the general AI field are mentioned only for comparison purposes and are not discussed in this paper; further reading can be found in these references [4,5]. This section discusses only the technologies used in the aerospace IVHM field with respect to AI.

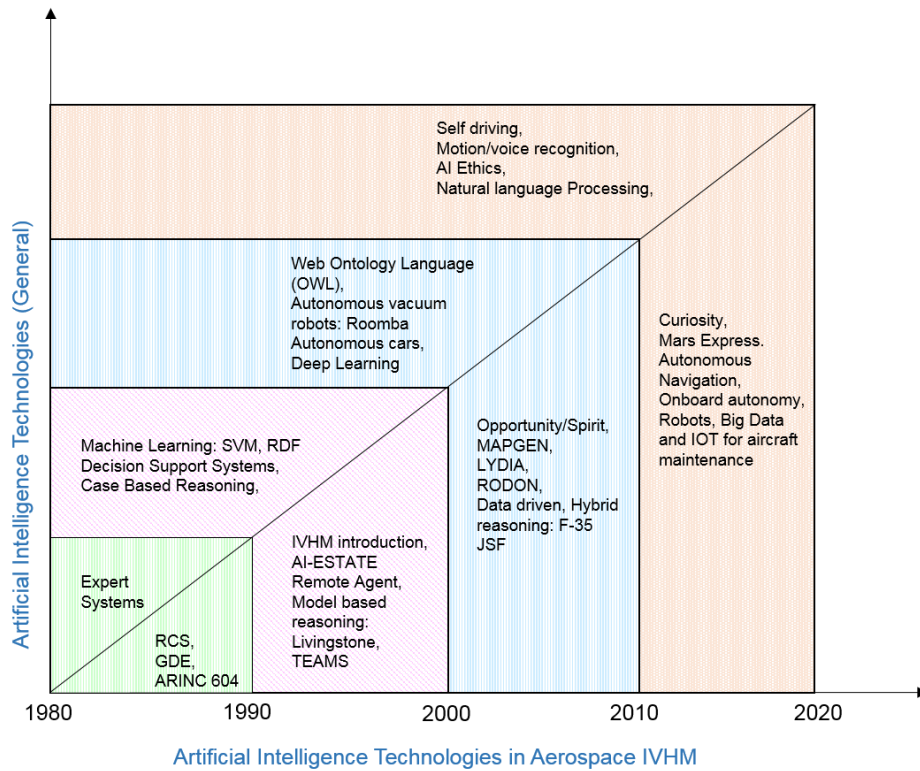


Figure 2-2 Trend of AI technologies in Aerospace IVHM versus general AI technologies

Spacecraft health monitoring: While the goal of IVHM with respect to the spacecraft health monitoring is to improve the reliability of space transportation [6], it also aims to play an advisory role for providing reliable contingency management [7]. The spacecraft health monitoring system has to ensure that the mission objectives are met with respect to the constraints and the spacecraft is protected from failures that could lead to the loss of operation [8]. To solve this issue, the spacecraft health monitoring systems have used a variety of AI technologies. In fact, the trend of technologies used for the Spacecraft health monitoring systems, show a correlation with the trends in the technologies used in the field of AI, wherein, those AI techniques that are introduced and matured over a few years are implemented in the aerospace IVHM field in the next few years. The examples

of this correlation can be seen in [Figure 2-2](#). It shows that considering a timeline starting from the 1980s, the rule-based expert system was the widely used AI technology. During this time period, technologies like the expert system for scheduling, NAVEX [9], and the Procedural Reasoning System (PRS) for health monitoring Reaction Control System (RCS) [10] were developed in aerospace. Moving to the 1990s, after the ‘AI Winter’ when the expert system was reaching its bottom, the focus turned to developing other technologies like Decision Support Systems (DSS) and improving other related fields like machine learning. While AI techniques like neural nets and evolutionary algorithms are in use for a long time, the other machine learning algorithms like Random Decision Forests (RDF) and Support Vector Machines (SVM) were introduced only in the mid-90s [11,12]. Model-based reasoning was being widely used in the ‘90s in the aerospace industry, influenced by the General Diagnostic Engine (GDE) that was presented by de Kleer in 1987 [13]. In 1999, Remote Agent, developed by NASA, was the first fully autonomous AI system used in a spacecraft which included a model-based reasoning system (Livingstone) and a DSS [14]. Similarly, TEAMS is a model-based diagnostic tool that was used for ground-based diagnostics of power distribution systems in International Space Station [15]. In the 2000s, when the AI field started focusing on robotics, speech recognition, deep learning, and autonomous vehicles, the aerospace IVHM focused mainly on model-based reasoning engines. As a result, RODON, LYDIA, and several other reasoning engines were developed in the 2000s. The aerospace field also focused on sending rovers to the planet Mars, leading to launch of the Mars Exploration Rovers Spirit and Opportunity in 2003. These rovers had a DSS, namely, MAPGEN (Mixed Initiative Activity Planning Generator) that carried out automated constraint-based planning, scheduling and temporal reasoning for the rovers [16]. In this decade of 2010s, there is a further increase in the development and applications of space robots, especially the Mars Exploration rovers like Curiosity, and Mars Express. The focus in the space transportation shifted to autonomous navigation and onboard autonomy via the Fault Detection Isolation and Recovery systems [8], whereas the focus in AI field is on motion recognition, natural language processing as well as self-driving vehicles.

Aircraft Health monitoring: In the aircraft industries, the main goal of IVHM is to reduce MRO costs and increase the availability of aircraft by enabling integrated health monitoring and supporting CBM. As for the timeline, since the operation of aircraft and

its maintenance go hand in hand, the evolution of aircraft design and operations naturally led to the periodically updated maintenance programs. Besides, the development in the field of aircraft maintenance is in line with the stabilized AI technologies at that period, similar to the spacecraft ([Figure 2-2](#)). The earlier form of troubleshooting involved testing the circuit for continuity in the mechanical-analog devices of aircraft systems, and in the 1980's the technologies moved toward digital systems leading to electronic testing of Built-In Test (BIT) circuits to detect the faulty Line Replaceable Units (LRU) [17]. The first formal standard for health monitoring, ARINC-604, 'Guidance for Design and Use of Built-in Test Equipment', was formulated in 1988 [17] and the guidelines for using matured AI techniques in the diagnostics systems was provided in 1995 by IEEE standard 1232 AI-ESTATE (Artificial Intelligence- Exchange and Service Ties to All Test Environments) [18]. Moving to the early 90s Centralized Maintenance Computers (CMC) were developed for B747 aircraft to diagnose the health of multiple LRUs. These CMCs were initially implementing AI technologies like rule-based expert systems and then evolved to apply model-based failure propagation techniques in B777s in the later 90s, similar to the spacecraft model-based reasoning techniques as shown in [Figure 2-2](#). The maintenance systems like Honeywell Prime Epic Aircraft Diagnostic Maintenance Systems (ADMS) developed in the early 2000s, used modular systems, data-driven approach for diagnosis and loadable database [17]. In this period, the maintenance systems used AI techniques like Case-Based Reasoning (CBR), and Web of Language (OWL) for troubleshooting activities with the help of historical records. Further, hybrid combinations of model-based and data-driven approaches were widely used in health management programs for F-35 Joint Strike Fighter [19]. The Maintenance System in B787 uses model-based diagnostic approaches with loadable databases to isolate faults and carry out model corrections. It uses information from various sources like control and design documents and FMEA as well [17]. The maintenance programs like AiRTHM by Airbus enable real-time monitoring by onboard maintenance systems with the help of web-based technologies and wireless communications [20]. Over the years, several standards like ARINC 624, 664, and 666 were published to help design the onboard health monitoring systems [17], whereas the PHM standards like OSA-CBM provided framework to implement a CBM system and the IEEE standards like SIMICA provide support for PHM systems to use historical data for diagnostics and prognostics [18]. It is

to be noted that the focus in the IVHM systems which were earlier at the LRU level is able to move towards the subsystems/ systems level owing to the advancement in the sensor technology, data analytics, communication protocols, and the computing power.

In the present decade, the IVHM field is taking advantage of the cloud-based computing and big data analysis for improving real-time health monitoring. Moreover, the aerospace industries like Rolls Royce are attempting to use robotics for inspection of the systems and assessment for the need for maintenance [21], in order to save time that would be otherwise spent for repair and overhaul.

2.1.2 Need for Vehicle Level Health Monitoring

In the previous section, it was discussed that the focus of IVHM systems is moving from LRU level to the Systems level due to the advancement in the supporting technologies. This is because the main goal of IVHM is to ‘integrate’ information from all the systems of a vehicle and to make a well-informed decision on the maintenance with respect to the health of the systems. According to the FAA report on General Aviation Safety 2018, 2 out of 10 leading causes of fatal accidents between 2001-2016 were due to System component failure [22]. In spite of the preventive and periodic maintenance, there are several occasions where the system still fails on its own, or due to its interaction with the other systems, leading to extended downtime and added maintenance cost, if not, accident. For example, a Boeing 777-200 ER had an engine rollback in 2008, because of which the aircraft touched down 300m short of the pavement path in the runway. The root cause, however, was found to be in the fuel oil heat exchanger, where a restriction due to the formation of ice has reduced the fuel flow to both the engines resulting in reduced pressure ratio [23]. Incidents like this emphasize the importance of having a holistic view of aircraft systems, instead of performing analysis only at the component and subsystem levels. While the state-of-the-art IVHM system ADMS in B777 can capture the cascading faults due to the interactions between the components, it uses an expert-derived fault propagation model as a systems reference model [24]. Any unexpected fault propagation could not be captured by this system and a complete IVHM system could not be fully realized. There is still a lack of health monitoring systems that could function at the vehicle level to detect and isolate faults that cascade between the systems before it is too late. In order to identify the root cause of a fault that has affected another system and

predict its cascading effect, the health monitoring system needs to reason through data from multiple systems in a vehicle, consider the causal relationships of the systems, assign priorities, and resolve conflicts. In short, the IVHM system requires an intelligent reasoning system that could analyse and make decisions regarding any system fault, its root cause and the effect at the vehicle level.

With this purpose in mind, the paper explores the AI reasoning technology. This section gave an overview of IVHM over the years and the need for vehicle health monitoring and Section 2.2 aims to answer the basic questions like what reasoning is, where is it applied, and how does it function. Section 2.3 summarizes the applications of reasoning in aerospace IVHM and Section 2.4 discusses the shortcomings in IVHM, the ways to mitigate them with reasoning along with the challenges and opportunities. Lastly, Section 2.5 concludes the chapter.

2.2 Reasoning – A Literature Review

In general, the term ‘reasoning’ is a derivative of human reasoning, where the methodology for problem-solving is governed by applying logic and cognition. The origin of reasoning as a study traces back to Greek Philosophy. Aristotle coined the term ‘Syllogism’ which is the study of logic that defines the rules for reasoning to operate [25]. Reasoning is applied in every existing field, to tackle everyday problems. Reasoning can be as simple as deductions like ‘ $a=b$, $b=c$ and hence $a=c$ ’, or as complicated as a combination of multiple strategies, based on the problem to be solved. To align with the scope of this paper, the literature review on reasoning restricts itself to the field of engineering. Even a short survey on the functions of reasoning in engineering results in a wider range of industrial applications.

Figure 2-3 shows a representative sample of various functions of reasoning as an AI technology and their applications in different sectors. The first layer from the centre shows the different problem fields where reasoning is applied; this includes, but not limited to, diagnosis, scheduling, reporting, decision support, troubleshooting, behavioural analysis, strategy planning, optimization and context awareness. The second layer shows their respective applications in the industries and the outer layer shows their corresponding industries. This non-exhaustive list of industries that employ reasoning encompasses aviation, medicine, gaming, robotics, traffic monitoring, design, simulation,

production, services, communication, electronics, automotive, computers, smart rooms, and smart buildings. The applications in the middle layer that are developed to solve the problems shown in the first layer from the centre in Figure 2-3, use different types of reasoning methods and systems to achieve their respective goals. For example, the tools like SpotLight Maintenance developed for troubleshooting in aircraft engines [26], the expert system DOCTOR developed for field service of air conditioners [27], the scheduling system CABIN [28], and decision support system CLAVIER developed by Lockheed Martin [29], use Case-Based Reasoning (CBR) systems. The CBR systems are also used for incident reporting for aircraft accidents [30] as well as cybersecurity threats [31]. Similarly, Rule-Based Reasoning (RBR) is used in the context-aware application for a smart room, namely, The Conference Guard [32], and in the goal-driven diagnosis application, TraumAID 2.0 [33]. The UM-PRS is a Procedural Reasoning System (PRS) that is designed for robotic reconnaissance task [34].

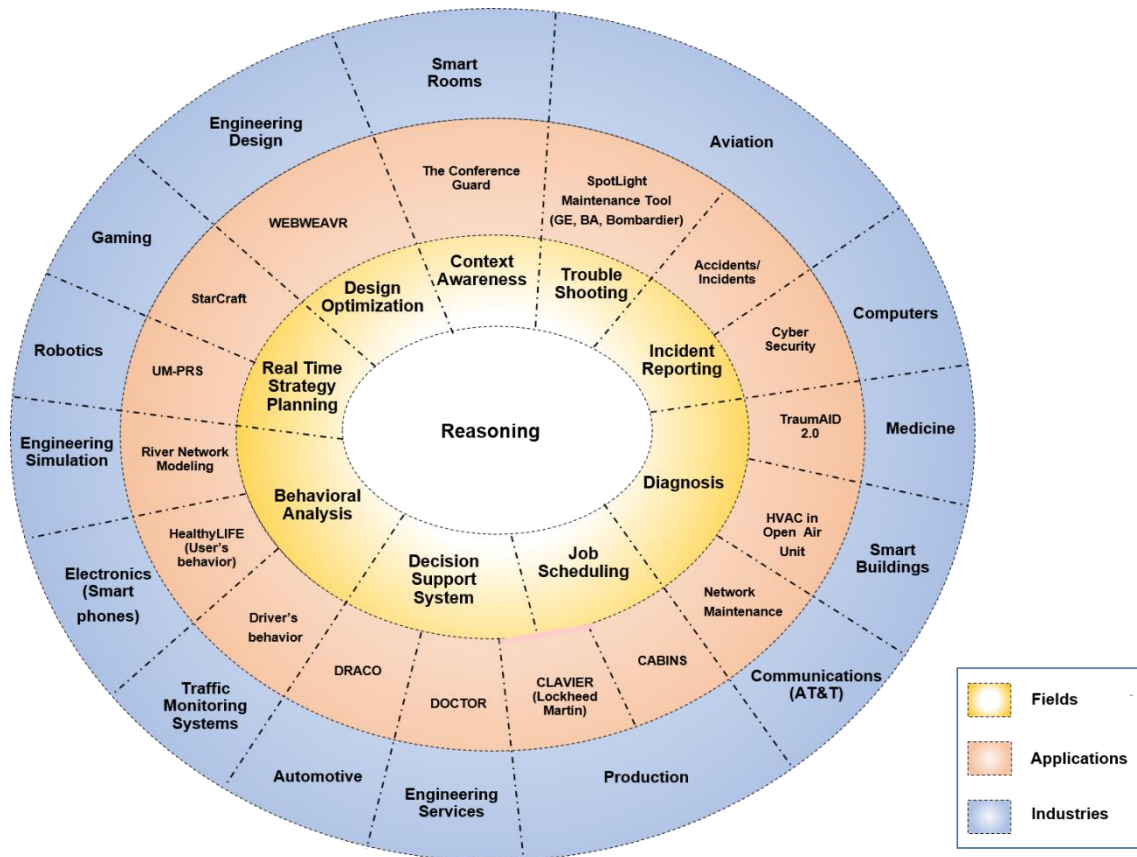


Figure 2-3 Applications of Reasoning in Engineering

Apart from CBR, RBR, and PRS, there are other reasoning methods which are used for solving various problems in the industries shown in [Figure 2-3](#). WEBWEAVER developed for design optimization is a Bayesian reasoning engine [35]. DRACO, an Intelligent Decision Support System (DSS) used a fuzzy reasoning, along with neural network models for the quality control process of automotive coating operations [36]. HEALTHYLIFE, used in smartphones for recognizing user behaviours, applies Answer set programming based Stream Reasoning along with Artificial Neural Networks [37]. A Model-Based Reasoning with mathematical modelling is used to represent a river network model for simulation of water quality control [38]. A Real-Time Strategy video game, StarCraft, extensively uses several reasoning methods including CBR, Neural Networks, and Bayesian Models for tactical decision making as well as strategic planning [39]. Furthermore, reasoning methods are used to infer the driver's behaviours during traffic in traffic monitoring systems [40]. A hybrid reasoning system with model-based and Tree Augmented Naïve Bayesian Network has been used to diagnose and find anomalies in the Open-Air Unit installed in a smart building, like a public health centre, in order to increase air quality and reduce Heat and Ventilation and Air Conditioning costs [41]. Dynamic Mental Models (DM2) used analytical model and pattern recognition rules and have been tested by applying hypothetical diagnostic reasoning on expert systems for networks maintenance at AT&T [42].

2.2.1 Definitions

It is to be noticed from [Figure 2-3](#) that, in the applications presented, reasoning is implied both as a problem-solving strategy as a part of another system, and as a system in itself. This difference can be better explained with defining the terminologies ‘reasoning strategy’, ‘reasoning system’, ‘reasoner’ and ‘reasoning’. While the term ‘reasoning strategies’ refers to the approach with which the data is assessed in a certain way, a ‘reasoning system’ refers to a software system that applies the reasoning strategies in an ‘input-process-output’ manner to achieve the specified goal. The term ‘reasoner’ is used in the literature many times referring to the reasoning system [43] and sometimes to the algorithms that carry out reasoning [44]. To avoid confusion, in this paper, ‘reasoner’ refers to the reasoning system itself. Finally, the term ‘reasoning’ itself refers to the process of analysing the given data using reasoning strategies and other algorithms in

order to achieve the goal of the reasoning system or any system that the reasoning is a part of. The next sections further explore the commonly used reasoning strategies as well as the reasoning systems in the field of aerospace IVHM.

2.2.2 Reasoning Strategies

Several reasoning strategies are applied in the general problem-solving systems. In the case of health monitoring systems, reasoning helps in achieving the goals of diagnosis and prognosis, which is to detect and isolate the fault and to estimate the Remaining Useful Life (RUL) respectively. These systems can use one or a combination of the reasoning strategies to reason through and reach their goals. [Table 2-1](#) presents the collection of different reasoning strategies commonly used in health monitoring systems. While there are several works related to health monitoring use these reasoning techniques, only one example is provided for each reasoning technique in [Table 2-1](#) to be concise. Each reasoning strategy has a different way of approaching a problem depending upon the type of data available. For example, if the given set of data is definitive but small, inductive reasoning will help in projecting or predicting the generic observations, whereas, in the case of abundant data and a certain hypothesis, deductive reasoning will help in drawing the conclusions. If the given data is insufficient or incomplete, abductive reasoning will form and test hypothesis based on the incomplete dataset, whereas, analogical reasoning will draw conclusions based on similar experiences from other cases. Likewise, depending on the type of data, if the data possess time information, temporal reasoning would be chosen for problems like timestamp comparison, whereas for data with statistical information, statistical reasoning can be used. Causal reasoning draws conclusions based on the cause-effect relationship of the data and approximate reasoning is applied in the cases that require less computational time.

Table 2-1 Reasoning Strategies

	Strategy	Features	Examples
--	----------	----------	----------

1	<i>Inductive Reasoning or induction</i>	A bottom-up approach which makes a set of generic projections from observations or data.	Inductive Monitoring System on TACSAT-3 detects anomalies by supervised learning with the help of a model constructed from nominal data [132].
2	<i>Deductive Reasoning or deduction</i>	A top-down approach wherein, a certain solution is found from the given premises by holding the hypothesis true.	Livingstone is a model-based reasoner in Remote Agent and it uses deductive reasoning. It draws conclusions based on the given commands and comparing the corresponding observations with the optimal responses [14].
3	<i>Abductive Reasoning or abduction</i>	A logical reasoning which constructs and tests a hypothesis based on the observations even if they are incomplete	As a part of the reasoning process patented by GE, the Advanced Diagnostic and Prognostic Reasoner uses abductive reasoning to construct a minimum set of fault conditions to associate with ambiguity groups [143].
4	<i>Analogical Reasoning</i>	Uses past experience to provide new conclusions by analogy	An intelligent maintenance support system for aircraft is developed using Genetic Algorithm and Case based reasoning (CBR) which retrieves the similar past cases by analogy [113].
5	<i>Temporal reasoning</i>	Helps to the reason of dynamic systems by considering time as an additional dimension	In the integrated prognostic reasoner developed for bearings, temporal reasoning is used to integrate a priori reliability information and correct the disagreement of the information source, with respect to time. [152]
6	<i>Statistical reasoning</i>	Uses statistical information of data sets.	Statistical Reasoning is used with fuzzy inference algorithm to develop a DSS [102].
7	<i>Causal reasoning</i>	Uses the relationship between the causes and their effects to draw conclusions	The data driven model developed to detect pipeline leak uses causal inference to find the pattern between the antecedents like flow difference and pressure difference to the consequent, leak size [69].
8	<i>Approximate reasoning</i>	Speeds up the process of finding a solution by sacrificing its completeness	The approximate reasoning is used with a Belief Rule Based methodology to reduce the complexity of the causal inference model for detecting pipeline leak [69]

2.2.3 Reasoning System – A generalized architecture

A reasoning system is a piece of software that implements reasoning strategies, alongside other problem-solving techniques to achieve certain objectives with the help of available knowledge. A generalized reasoning system architecture, shown in Figure 2-4, has a basic set of components which can be adapted to address particular goals based on the existing resources. The general components of a reasoning system can be grouped under three modules (Figure 2-4): i) Domain Library module, ii) Problem-Solving module and iii) Tasks module. The Domain Library module contains domain knowledge, the database of the application and the working memory. The database includes the data of the measured inputs and outputs of the system using which the reasoning will be carried out, and the domain knowledge consists of knowledge about the system represented in a certain format. The dynamic knowledge specific to the problem is stored in the working memory. In the Problem-Solving module, the components include i) an inference engine (also referred to as reasoning engine) which performs the action of reasoning based on the methodology chosen and the type of available data, and ii) a reasoner maintenance system which communicates with the inference engine to maintain consistency of the belief or truth of the system. The reasoner maintenance system can be either assumption based or justification based [45].

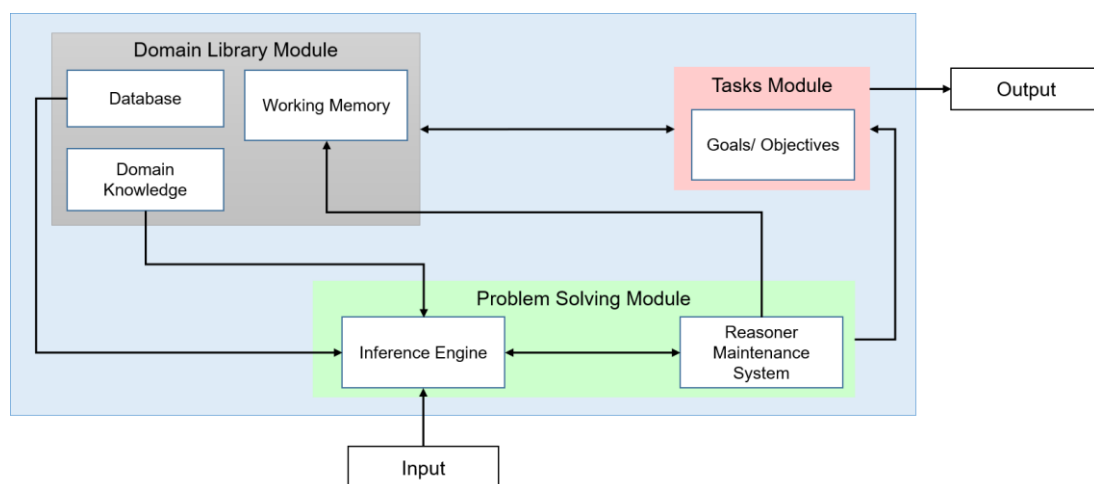


Figure 2-4 The architecture of a generalized reasoning system

The Tasks module of the reasoning system contains its objective or goal for a given problem. The reasoning system can deal with problem types like i) Constraint Search Problem (CSP), and ii) Planning and Decision Making [30]. CSP is a data-driven problem, where the problem is solved with an assumed set of variables and a defined set of constraints; variables get updated in every iteration with respect to the constraints to search for the solution in the solution space. Diagnosis and classification problems are of the CSP type, as their constraints are defined as fault symptoms, and classification group characteristics respectively. Planning and Decision Making is a goal driven problem, where the initial conditions and possible final solutions are pre-defined and the problem is driven in a certain path by applying rules to attain the defined goal.

To illustrate the functioning of a reasoning system, consider a scenario of isolating a fault in a fuel system that operates under a constant fuel flow. In this case, the goal/objective of the reasoning system is to isolate the fault. The domain library will consist of a database of the fuel system like sensor data of the parameters that are required to monitor the health of the fuel system. It will have the domain knowledge consisting of relationships between the monitored parameters and fault signatures of various faults as well. In this case, once an anomaly is fed as an input, the inference engine will search the database with the help of fault signatures to match the input anomaly with the fault signature in the database. The working memory will keep track of the transition states of the system when the problem is being solved. The reasoner maintenance system helps in keeping up with the consistency of the truth or assumptions of the system, in this case, a constant fuel flow. Hence in the case of contradicting truths, the reasoner maintenance system updates the working memory. The inference engine communicates with the reasoner maintenance system to check the assumptions and uses different reasoning strategies and algorithms appropriate for this problem and to derive the required solution. The output can either be the 'name of the fault' in case of known fault of the reasoning system already or 'Not found' in the case of a new fault.

2.2.4 Types of reasoning systems

There are different types of reasoning systems depending upon the availability of the domain knowledge, data, and the chosen problem-solving methodology. The knowledge-based systems use domain knowledge that is derived from the experts, the model-based

reasoning systems use domain knowledge that is derived from the governing equations or the behaviour of the systems and the data-driven reasoning systems are used when there is lack of domain knowledge and the reasoning is dependent mainly on the datasets. This section presents the different reasoning systems, the various types of knowledge representation and problem-solving methodologies they use. Several review papers discuss in depth about problem-solving methods used for diagnosis and prognosis [15,46–54], only a few methods are discussed in this section.

2.2.4.1 Knowledge-Based Systems

The knowledge-based systems carry out reasoning based on the existing knowledge base. This knowledge can be anything like procedural or declarative, structured or unstructured and are represented in such a way that the reasoner understands. In general, the knowledge can be represented in the form of a concept, its intent, and the context. The Concept is the basic unit of knowledge, providing the abstraction of real-world things. Concepts have an association with other concepts which give the context of the knowledge. Concepts and its associations can be represented using any of the technologies like Rules, Procedures, Frames, Nets, Models, Ontologies and Scripts, based on the intent of knowledge, which is the ability or the skill required to achieve the goal [55]. Some of the knowledge-based systems are discussed here and Table 2-2 presents the comparison of advantage and disadvantages of these systems.

Table 2-2 Advantages, Disadvantages, and applications of Knowledge-based reasoning systems

Reasoning Systems	Advantages	Disadvantages	Applications
<i>Expert System</i>	Easy to implement (straight forward) Transparent	Extrapolation not possible The system needs to be updated frequently Depends on expert knowledge – brittle Computationally expensive	DENDRAL [58] MYCIN [58] CLIPS [55]
<i>Procedural Reasoning System</i>	Real Time and meta level Reasoning Simple Semantics	Indexing mechanism makes it difficult to delete or change rules Tasks have to be kept short	RCS in NASA Space shuttle [10]

	Can act upon partial plans	Needs updating if plans are changed	Robotic reconnaissance [34]
Case Based Reasoning System	<p>Simple to use</p> <p>Problems that cannot be expressed mathematically can be dealt with</p> <p>Learning capacity – knowledge evolution possible</p> <p>Flexible architecture</p>	<p>Depends only on historical information</p> <p>Time-consuming</p> <p>Retrieval process needs powerful algorithm to get exact match</p>	<p>IDS [127]</p> <p>ISAC [64]</p>

Expert System: The Expert System architecture consists of a knowledge base and an inference engine (Figure 2-5). Its knowledge is generally represented in terms of rules. Rules represent domain knowledge in an ‘if-then-else’ format and they can be written in different programming languages like C, LISP, and OWL. For example, in the CLIPS expert system used by Siemens, rules are written in OWL 2 language in the format of concept-ontology and instance-ontology [56]. In some cases, frames are also used to represent the knowledge in the expert system. Frames are used to represent stereotyped knowledge as a collection of attributes and their associated values. An example is the meteorological vehicle system, wherein the expert system for fault diagnosis is built using object-frame structure, with the frame being a collection of state-object, test-object, rule-object and repair-object [57].

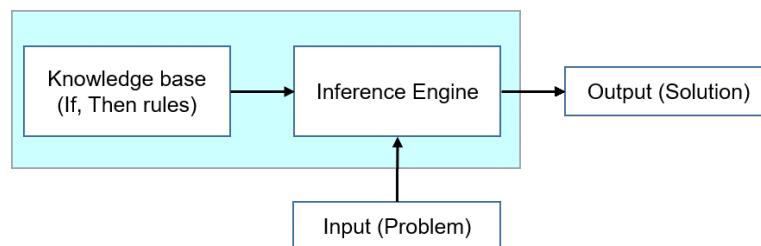


Figure 2-5 Expert System Architecture [59]

Expert systems employ Rule-Based Reasoning (RBR) methodology to solve its problem. The RBR is executed in the following two ways: i) Forward Chaining, and ii) Backward Chaining. Forward Chaining starts with the initial state of facts and applies the rules until the endpoint is reached. Backward Chaining starts from the hypothesis to prove and looks for rules that will allow the hypothesis to be proven. In other words, it starts with an effect and looks for the possible root causes that could lead to that effect. Forward Chaining is data driven whereas Backward Chaining is goal driven [58].

The Expert System is one of the initial implementations of AI software, with DENDRAL and MYCIN being two of the earliest rule-based expert systems developed for analyzing chemical components and for diagnosing infectious diseases respectively [59]. The Expert System was popular in the 1980s and it has been implemented for several health monitoring and maintenance systems. However, it could not sustain the growing demands of the field. While the expert system required straightforward implementation, its knowledge base is dependent on the experts and hence, brittle and needed frequent maintenance. It could not provide intuitive results and incur high computational costs [60]. Furthermore, the expert system could not handle the dynamics and uncertainty involved in the aerospace models [46]. Hence, the expert system was integrated along with other frameworks like in Blackboard architecture in BEST (Blackboard-based Expert System Toolkit) [61] to enhance the performance of the application. It is used along with model-based methods as well. An example is the expert system patented by GE that applies model-based reasoning for diagnosing faults in rotary machines. In this system, the model is represented by partial differential equations based on their first order; the vibrations measured were used by the rule-based expert system to identify the root cause [62].

Procedural Reasoning System (PRS): The PRS is a knowledge-based reasoning system which has its knowledge in the form of procedures called the Knowledge Areas. The PRS implements the Belief-Desire-Intention concept of modelling for real-time reasoning of dynamic systems [10]. It consists of the following modules (Figure 2-6): i) a goal or objective set, ii) a database with domain knowledge and beliefs that update themselves with new knowledge, iii) a knowledge area which is a library of procedures for actions and tests to achieve the goals, iv) an intention graph which has partially completed

procedures to run real-time, and v) an interpreter which communicates with all these modules and carries out reasoning. The interpreter receives the goal or objective for the system, chooses the correct procedure required from the knowledge area, places it on the intention graph to narrow down the set of actions, chooses the correct action based on the intention, and finally starts the procedure which will update the next goal [10]. [Figure 2-6](#) shows the general architecture of the Procedural Reasoning System [10].

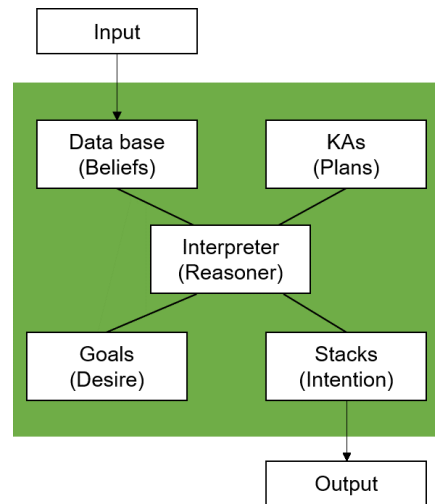


Figure 2-6 Procedural Reasoning System (PRS) Architecture (adapted from [10])

The PRS has been applied to monitor the malfunctioning of the RCS of NASA’s space shuttle and also to diagnose and control the overloading of telecommunication networks [63]. The PRS architecture is simple for execution and reduces the computational time as the procedures can skip the unnecessary steps for a particular problem, and narrow down the rules to the relevant set directly [63]. The PRS can implement real-time reasoning and handle dynamic systems, but it can handle only simple plans; any changes to the existing plans and procedures will be time-consuming and tedious.

Case-Based Reasoning (CBR) System: The CBR has its knowledge derived from the historical cases. It has a simple framework consisting of four phases: i) retrieve, ii) reuse, iii) revise, and iv) retain ([Figure 2-7](#)). In the retrieval phase, knowledge in the database (case repository) in the form of previous experiences and histories (cases) related to the application are searched for. These old cases are then retrieved based on their index and interpreted for the current problem. In the reuse phase, the old cases are adapted to the present situation in order to find the solution. Evaluation of the new cases is carried out

and solutions suggested in the revise phase and the new cases are then added to the case repository for future learning, as a part of the retain phase [64].

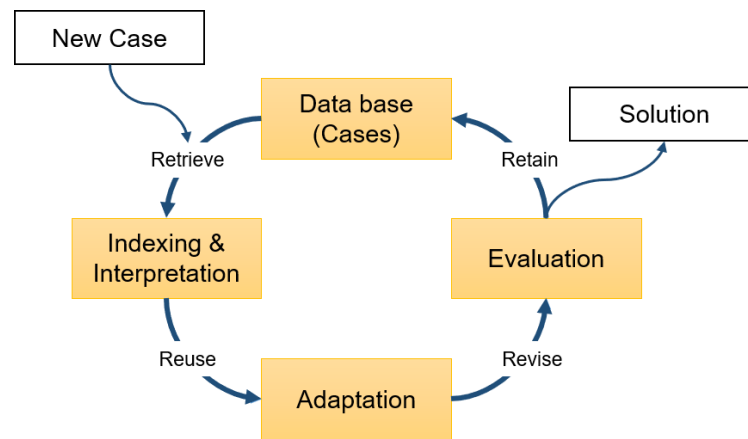


Figure 2-7 Case Based Reasoning Architecture (adapted from [63])

One of the examples of CBR application is Intelligent Systems for Aircraft Conflict Resolution (ISAC) [65] which was developed to help the decision-making process of aircraft controllers to resolve the conflicts between aircraft. CBR is one of the most commonly used reasoning systems as its architecture has the capability of accommodating any advanced algorithms, mainly text processing techniques. Moreover, CBR does not require prior knowledge about the system as it solely depends on past experience. Since the CBR involves learning as a part of its methodology enabling knowledge evolution, the CBR can evolve and become expert in the domain, thus having the potential of becoming the future expert systems [66]. This is an advantage over the other frameworks since the other reasoning systems require maintenance time for updating the knowledge base. Nonetheless, the CBR system is computationally demanding when compared with the PRS and simple rule-based Expert Systems.

The knowledge from the historical records or cases can be represented in a variety of graphical representations. One such method, Semantic nets, are used for representing the relationship between the concepts that are usually described with verbal information. Knowledge is extracted by mining text with a defined set of rules and finding relationships between the phrases, and it is represented in accessible formats like tables or matrices. Semantic nets are used in CBR in cases like searching the maintenance repair records to find similar cases [67]. They can also be used for text mining the datasets to find anomalies [68]. Dependency Matrix or D-Matrix [69,70] is a commonly used

representation in the field of diagnosis as well. It consists of dependency/ causal models represented in the form of a matrix using the logical relationship between the elements. D-matrix is used in CBR to represent the relationship between the tests and diagnosis, where a case is a collection of test results that are mapped to the appropriate diagnosis based on its results. Thus when a new case or test result is given, the similar cases are retrieved and the corresponding diagnosis is chosen [69].

Apart from Semantic nets and D-Matrix, there are several other graphical methods that can be used for historical knowledge representation. These methods can as well be used to represent the knowledge of model-based systems and the a-priori knowledge of data-driven systems.

2.2.4.2 Model-based Reasoning Systems

Model-based reasoning systems are used when there is a rich domain knowledge about the system. Its knowledge i.e., the concepts and their relationships to the other concepts are represented based on their physics-based or functional relations [71]. For example, NASA developed a physics-based model for diagnosis of the Liquid Hydrogen Propellant Loading System, as it requires a detailed model for understanding the system under healthy and faulty conditions, enabling quick detection and recovery, in case of failures [72]. On the contrast, a fuel system was represented by a functional model in MADe, in which, the functional flow in terms of energy, material, and signal between the components. This model differentiates nominal cases from faults and helps in isolating the identified faults [73].

The theory of Model-Based Diagnosis was first formulated and applied in a General Diagnostic Engine by de Kleer [13]. This framework used the diagnostic reasoner to identify the behavioural discrepancy between the model and the observed value and it made use of the assumption-based truth maintenance system to achieve efficiency. Ever since a lot of methods have been developed for representing and solving the model-based problems efficiently. Presently, the model-based reasoning uses both qualitative and quantitative methods to solve its problems. The quantitative methods involve comparing the output predicted values generated by analytically redundant models with the measured sensor values from the systems to generate residuals. The error can be reduced by solving the algebraic or differential equations for the best set of parameters. Parity equation is a

residual generation method which consists of linear equations that link the different variables by their mathematical relationships. The deviation in the parameters can be found through the residuals and the faults can be flagged if the errors cross the set threshold [74]. These equations can also be grouped in a way that a set of parity equations could represent a diagnostic model and are solved in order to isolate the faults [75]. Developing this model is time-consuming because of the complexity involved. Kalman Filter is another approach for residual generation, which estimates the state of the system in presence of noise and is used widely in diagnostics for its ability to adaptively estimate the state parameters of the degradation model as the time evolves [76]. One drawback of this method is that it assumes the equations to be linear and noise to be of Gaussian distribution [77]. These drawbacks are mitigated by using derivatives of this method like Particle filters, Switching Kalman Filters and, Extended Kalman Filters [52,76,78].

Among the qualitative methods, Timed Failure Propagation Graph (TFPG) [79] is used for the abstract representation of the system behaviour during a failure propagation. It is a directed graph consisting of nodes representing failure modes as well as the possible deviation in the system behaviour in the form of discrepancies, and edges that connect the failure modes to their respective discrepancies. TFPG helps in tracing the root cause of a propagated fault as well as the likelihood of further failure progression and it is used widely for representing the complex dynamic systems. TFPG is applied in an embedded systems environment to isolate the faults [80] and a hybrid version of TFPG is applied in the diagnosis of switching systems [81]. Petri-Net, a bipartite graph, is another qualitative model used in model-based reasoning. It is a mathematical modelling language used to represent the relationship between the conditions and the events in a distributed system. It consists of i) nodes representing events or transitions, ii) places representing conditions and containing tokens, and iii) arcs representing the direction of the transition from a place (condition). Petri-Net is used for analysing the dynamic behaviour of the system where the firing of a token from one place to another represents the transition taking place in the system. Petri-Net is used in failure analysis in many systems [82,83] and a hybrid version of the Petri-Net, combined with fuzzy reasoning, called Fuzzy Petri-Nets is used for fault diagnosis while dealing with uncertainty [84]. The other commonly used qualitative methods are Fault Tree Analysis and Multi-flow Models. Fault Tree Analysis [85,86] expresses the logical relationship between a failure and its possible causes, but it

can be modelled only for the expected failures. Multi-Flow Model [87] represents the system and its subsystems in terms of its goals and functions and networks connecting these functional flows.

2.2.4.3 Data-driven reasoning systems

Data-driven reasoning methods extract the underlying patterns in the datasets and use them for understanding the systems' characteristics to reason through them. They are generally used in the cases which have rich datasets and have a very little or no domain knowledge itself. This section briefly discusses the most widely used data-driven methods.

Bayesian reasoning uses the estimates based on data from statistics and deals with uncertainty in the system model. The cause-effect relationship between the variables have to be known probabilistically but the entire system model knowledge is not required [88]. The Bayesian Network is a probabilistic graphical model which uses a set of nodes to represent variables of a system and directed edges, the relationships between the variables. It uses Bayes' theorem as its principle to decide the root cause based on a priori probability with respect to the observed value. Conditional probabilities are specified for every node of this network and probability estimates get updated based on every new observed value. The Bayesian Network was first introduced by Pearl in the 1980s [89] and has been widely used for a variety of intents including tacking uncertainties, within the broad field of diagnostics. For example, the Bayesian Network and its derivatives like Dynamic Bayesian Network and Tree Augmented Naïve Bayesian Network have been applied for fault diagnosis of a fuel system [90], auto-generation of the nets for diagnosis of an Electrical Power System [91], benchmarking of diagnostic systems [92], enhancing existing diagnostic models [93,94] and for change point detection of a valve failure in a fuel system in Unmanned Aerial Vehicle [95].

Fuzzy Reasoning is a data-driven method that deals with uncertainty using Fuzzy Set theory and the theory of fuzzy relations. Fuzzy logic was introduced by Zadeh (1965), in order to describe the systems that are 'too complex or too ill-defined' for mathematical analysis [96]. Fuzzy logic is a three-step process: i) Fuzzification, ii) Evaluation, and iii) Defuzzification. In the first step, the inputs representing a range of variables in overlapping regions are fuzzified by applying membership functions. These functions

overlap and hence there is a possibility of variables belonging to more than one membership. Then, a set of 'IF-THEN' rules (generally known as a deductive form) are applied to evaluate the response to each input. These sets of consequences are then evaluated to give one final output using the aggregation principle. The output is then converted to crisp quantities in the defuzzification stage [97]. Fuzzy Reasoning has been used directly for fault diagnosis in various industries [98,99] or in the derived form combined with Petri-Nets [100], besides developing Decision Support Systems (DSS) [101]. Likewise, Evidential Reasoning is another data-driven method that deals with uncertainty and imprecise information, by handling the evidence to make an assertion. It uses the Dempster-Shafer theory of beliefs which replaces probability distributions with belief functions. This theory is very useful in modelling uncertain conditions where probabilities in subjective judgments can be assigned a base degree of belief. Such degrees of belief can be combined from multiple independent evidence as well. One of the major advantages of Dempster-Shafer is that it enables reasoning with partial or conflicting pieces of evidence [102]. The lack of knowledge on assigning belief to the evidence and the computational complexity has made it a 'difficult to use' reasoning method. Dempster-Shafer reasoning has been used for fault diagnosis [103,104] and for developing the BNs for DSS [105].

Machine learning algorithms are the most commonly used data-driven methods in the current period. Albeit being in existence over several decades, they are used extensively in the recent times, owing to their capability to analyse the enormous data sets generated by the industries and are aided by the increased computing power. One of the most popular machine learning algorithms is Artificial Neural Networks (ANN). ANN is modelled after the biological neuron structure; it is used widely for pattern classification problems of diagnosis and prognosis because of its ability to infer functions from the observations. A neural network is composed of several layers (mainly the input, hidden and output layers) with interconnected nodes which have activation function and are connected to the network by the weighted edges. Back Propagation Neural Network a type of ANN, is a multi-layer feed forward network which learns by adjusting the weights and thresholds to reduce the feedback error [106]. Deep learning is a neural network with multiple hidden layers and is helpful in mining the information from big data sets in order to achieve the classification and other goals [107]. The neural networks do not require the

background of the data for analysis and use less operational time after training, but their major drawback is their need for big datasets for training the network. k-Nearest Neighbour is another non-parametric pattern recognition algorithm which clusters a sample to the k nearest subsets from the trained data set based on the majority vote from the k-subsets [108]. Similarly, the Support Vector Machine (SVM) is a classification algorithm which groups the datasets by introducing the boundaries or hyperplanes with a maximum distance of separation [109]. Principal Component Analysis (PCA) is a parametric algorithm which is used to derive the statistical model from the historical data. It reduces the dimensionality of the data by projecting the historical training data onto a lower dimensional space and uses it to study the factors that contribute to the major trends [48]. The Decision tree algorithm uses a divide and conquer approach on the datasets to classify the data space by recursive partition until there is no further splitting possible [109].

The data-driven algorithms include traditional statistical models like Weibull, Normal distributions, Hidden Markov model and Cox Proportional Hazard models as well. Weibull and normal distributions are used to fit the failure data to predict the time-to-failure [77]. Hidden Markov Model contains the transition and observation probabilities and is used for detection in the stochastic processes where the transition is not directly observed [110]. Cox Proportional Hazard model is a powerful regression model that uses hazard function or conditional failure rate function as a baseline to build predictive models for failure events based on trigger events. These trigger event can be characterized as anything like a failure or a particular behaviour and the model can be used to build a framework that connects the failure mode with its indicator [111].

Figure 2-8 shows the Strengths, Weaknesses, Opportunities, and Threats (SWOT analysis) of both model-based and data-driven methods, summarised from the reviews of above-mentioned techniques. Though the model-based method can be precise and could be extrapolated, it requires domain knowledge and can be time-consuming. Further, modelling errors and under-modelling are the downsides to this approach. As for the data-driven approach, it does not require detailed domain knowledge and modelling skills, but it does require immense data and risk misinterpretation of results as well as imprecise

outputs. Thus, the choice of problem-solving methods depends on the factors like available knowledge, data, time, and skill.

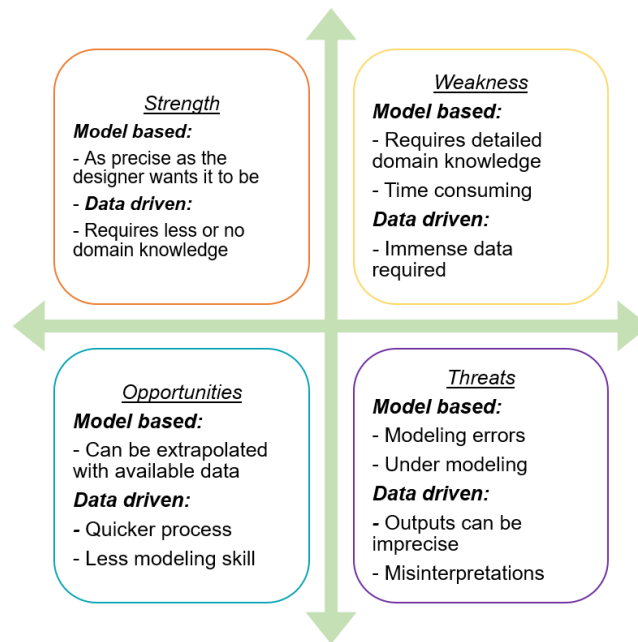


Figure 2-8 SWOT Analysis for Model based vs data driven methods

2.2.4.4 Hybrid reasoning Systems

These systems use a combination of reasoning techniques, applied in order to compensate for the disadvantages the algorithms possess. There are several studies that carried out reasoning by combining different methods. For example, a combination of Artificial Neural network and Fuzzy Inference System (ANFIS) is one of the most used hybrid methods in diagnosis applications [58]. Fuzzy algorithm is used with Bayesian Networks as well to model the gray-scale health as component degradation for a lithium-ion battery [112]. Another such combination is the Genetic Algorithm that is used with the CBR to enhance the retrieval ability of the CBR mechanism for fault diagnosis of electronic ballasts in aircraft [113]. Apart from the combination of multiple data-driven methods, in many cases, data-driven methods are combined with model-based reasoning as well. One example is Abductive Diagnosis through Adaptation of Past Episodes for Reuse (ADAPtER) [114], where a CBR system is integrated with model-based reasoning system in a master-slave architecture, using two knowledge bases, i) solved diagnostics problems in the case memory, and ii) a behavioural model in place of the domain knowledge. Another such combination can be found in the hybrid prognostic model, that was

developed using a physics-based model and similarity-based data-driven approach in order to carry out short-term predictions as well as RUL estimations in the cases of a clogged filter and fatigue crack propagation [115].

2.3 Applications of Reasoning in Aerospace IVHM

As the previous section discussed the different types of techniques used in the reasoning systems and the various problems that could be solved by reasoning, this section discusses how the reasoning systems are applied practically in aerospace IVHM. It is already known that the aim of IVHM is to collect and utilize as much useful information as possible to understand the behaviour of the systems, to diagnose and predict their health in order to aid CBM. Hence, in aerospace applications, reasoning systems are implemented majorly with the goal of real-time health monitoring, diagnostics, and prognostics as well as the troubleshooting activities. A select few examples of applications of reasoning in these areas are discussed below. [Table 2-3](#) summarizes the various reasoners developed by different industries for health monitoring in aerospace IVHM.

Table 2-3 Reasoning in Aerospace Health Monitoring

Reasoner	Organization	Used in / Experimented on
NASA		
SHINE	JPL	Deep Space Missions [117]
Livingstone	NASA AMES Research Centre	Deep Space Missions [14]
BEAM	NASA AMES Research Centre	Propulsion IVHM Test Experiment [119]
R2U2 Framework	NASA AMES Research Centre	Electric Unmanned Aerial Vehicle health management program [120]
MAPR	NASA AMES Research Centre, Ridgetop Group, Inc	Anomaly detection and fault classification in the Electro Mechanical Actuator (EMA) [70]
Aerospace OEMs		
AGET	Pratt & Whitney	Aircraft Engines [122]
IOT based ontological reasoning	Siemens	Gas turbines [123]
CONSOLIDATE	Smith Industries Aerospace and Defense, Inc	Aircraft engines [124]

Meta-diagnostic reasoner	Airbus	Centralized Maintenance Systems' knowledge database [125]
Smart TPS	Boeing	Navy/ Airforce aircrafts [126]
IDS	For Air Canada	Generic Maintenance Management System [127]
COTS Packages		
RODON	Uptime Solutions AB	Adapt [13]
eXpress	DSI International Inc	Multi-disciplinary applications [129]
TEAMS	Qualtech Systems INC	EMA [130], Naval Shipboard Systems [131], TacSat3 [132]
MADe	PHM Technology	Multi-disciplinary applications [133]
ReasonPro	Impact Technologies	Avionics [134]
Health Management Programs		
Bayesian network top level reasoner	Impact Technologies	Avionics Vehicle Health Management [135].
Ground based reasoner	Boeing	Integrated Diagnostics System [136], Aircraft Electrical Power Systems Prognostics and Health Management (AEPHM) program [137]
Model-based TRANSCEND + Data driven reasoning	NASA	PHM for EMA [138]
Academia		
FACT	Vanderbilt University, and Budapest University of Technology and Economics	Fault diagnosis in aircraft fuel system [139]
LYDIA	Delft University	Implemented in TELEMOS, HMS-RSTS, and Harbor Cranes [140].
PHM reasoner	Cranfield University, IVHM Center	EPGS Failure prediction [141]
HyDE	University of California, NASA Ames Research center	Propulsion system in ALDER, EPS in ISS [121]

By NASA: NASA is a pioneer in the field of aerospace IVHM. Over the years, it has developed several reasoning systems for monitoring the health of its space shuttles, satellites, and aircraft. Space Health Inference Engine (SHINE) is a reusable expert system used for real-time and non-real-time health monitoring and diagnosis. It consists

of a blackboard structure, knowledge base, and database; it has high computational power and speed that is required for real-time monitoring of deep space missions like Voyager, Galileo, and Cassini [116]. The first fully autonomous AI system in a spacecraft, Remote Agent, has a planning and scheduling AI module, a multi-threaded executive module and its centrepiece is a model-based reasoning software, Livingstone [14]. Livingstone is developed to monitor the overall behaviour of a complex system. Livingstone consists of qualitative models for both nominal and off-nominal behaviours and is capable of reasoning through system-wide interactions, thus enabling the inference of the effect of any failure over the complex system [6]. Similarly, Beacon-based Exception Analysis for Multi-missions (BEAM) [117], also developed by NASA consists of several signal processing modules, which was integrated with Livingstone, where it functioned as a virtual sensor to resolve the ambiguities for multiple fault scenarios in Propulsion IVHM Test Experiment [118]. In this, BEAM was used to detect and isolate a local fault, whereas Livingstone used the BEAM's output as evidence to reason through the entire system, thus complementing each other. NASA developed the Realizable Responsive Unobtrusive Unit (R2U2) framework, in order to improve the diagnostic capability in an electric Unmanned Aerial Vehicle health management program. The R2U2 framework was adapted to use the prognostic information derived from a Bayesian Network to improve diagnostic accuracy [119]. Another reasoner, namely the Model-based Avionic Prognostic Reasoner (MAPR) [71] is capable of real-time monitoring, processing as well as reconfiguring the model based on the system's health and it was used for anomaly detection and fault classification in the Electro Mechanical Actuator (EMA).

By Aerospace OEMs: The Original Equipment Manufacturers (OEMs) of aircraft industry have developed their own reasoning systems and the combination of strategies to aid their maintenance programs. For example, Automated Ground Engines Test Set (AGETS) Model-Based Reasoner developed by Pratt & Whitney uses an AI tool called Qualitative Reasoning System to run diagnostics tests and isolate failures using a troubleshooting approach involving F100-P100/200 gas turbine engines [120]. Siemens uses the Internet of Things (IoT) architecture and ontological reasoning to carry out diagnosis and prognosis of turbines [121]. Smith Industries Aerospace and Defence, Inc. developed a diagnostic reasoning system called CONSOLIDATE, which uses an Object Oriented Database and hypothetical reasoning to diagnose the faults in aircraft systems

like engines [43]. Airbus developed a meta-diagnosis methodology to reason through and isolate faults in their Centralized Maintenance System's knowledge database [122]. Boeing developed a Smart Test Program Set (TPS) software suite with AI Estate architecture that was utilized to communicate with multiple system reasoners and to provide suggestions for aircraft in the Navy and the Airforce. The diagnostic reasoners used were developed in MATLAB and were capable of using the BIT data, historical diagnostic and maintenance records and other related information to provide suggestions on entry points and call outs in the test programs to the engineers via the Smart TPS suite [123]. The Integrated Diagnostic System (IDS) developed as a generic maintenance management system for Air Canada, uses CBR to retrieve and classify the warning and failure messages from aircraft with case bases, as the original messages cannot be tracked with simple string matches. Furthermore, a case-base for diagnostics is developed which serves as a corporate memory that stores historical information of an aircraft fleet, thus helping the repair and the maintenance processes across the fleet [124].

COTS Packages: Numerous commercial reasoning packages are used by the industries for both health monitoring as well as maintenance. RODON is a commercial model-based diagnostic reasoning system from Uptime Solutions AB. It generates hypotheses based on contradictions between the simulated and observed systems. A declarative equation-based language called Rodelica is used to capture the model's knowledge. RODON has been tested on the ADAPT system [125] and is capable of generating decision trees which can be used for troubleshooting of the system [126]. DSI International Inc. developed the eXpress Embedded Reasoner, which is capable of being embedded into the HM system to receive systems data and provide a diagnosis of the system including root cause of failures, instructions for repair, as well as suggestions on further tests to be carried out [127]. Another commercial package, Testability Maintenance and Engineering Systems (TEAMS) is developed by Qualtech Systems Inc. (QSI). It is a domain-neutral Decision Support Software Suite that has been applied for integrated diagnosis and prognosis of various systems like EMA [128], Naval Shipboard Systems [129], and Tactical Satellite 3 (TacSat 3) spacecraft [130]. Similarly, PHM Technology developed the Maintenance awareness Design Environment (MADe) which is an integrated model-based design, analysis and DSS software suite [131]. MADe is capable of functional modelling, failure reporting as well as health-related analysis. ReasonPro is another COTS package,

developed by Impact Technologies has automated many reasoning processes, including the Built-In-Test (BIT) reasoning that uses system level knowledge and LRU level data. ReasonPro provides interactive diagnostics solutions to the maintainers, based on the information and codes collected from the aircraft and it further helps fault isolation through in ambiguity reduction and evidence-driven diagnostics [132].

Health Management Programs: Several health management programs have implemented reasoning as a part of the monitoring process as well. Impact Technologies developed a Bayesian Network top level reasoner for Avionics Vehicle Health Management [133]. The Integrated Diagnostics System [134] and the Aircraft Electrical Power Systems Prognostics and Health Management (AEPHM) program [135], both developed by Boeing, use a Ground-Based Reasoner component to optimize maintenance of hydraulic subsystem in legacy aircraft and to improve the mission readiness of Electrical Power Systems in military aircraft respectively. A Prognostics Health Management (PHM) program for EMA uses model-based TRANSCEND approach as well as a data-driven reasoning for diagnosis of the faults in the EMA [136].

Academic Work: Some of the reasoners developed by the academia are presented here. Fault-Adaptive Control Technology (FACT) was developed by Vanderbilt University and Budapest University of Technology and Economics, with support from NASA and Boeing. It uses Temporal Causal Graph at a lower level and TFPG at system high level to describe the effect of the faults. FACT was tested on an aircraft fuel system, it was able to identify the fault injected, as well as reconfigure the control system to maintain continued operation [137]. Similarly, a modelling language, Language for sYstem DIagnosis (LYDIA), developed by Delft University was implemented in TELE-operations and Model-based Supervision of instruments for planetary exploration (TELEMOS), a model-based diagnostic reasoning system. It has been demonstrated on several other projects like Health Management System for Reusable Space Transportation System (HMS-RSTS), and Harbour Cranes [138]. Another model-based prognostic reasoner was developed in Cranfield University for an Electrical Power Generation System, based on power management design. In this work, a hybrid mathematical model was developed to study structural and parametric faults and a framework with PHM reasoner was developed to predict the failures accurately along with the warning of

secondary damages that might affect aircraft operations [139]. Furthermore, the Hybrid Diagnostic Engine (HyDE) reasoning system was developed by the University of California along with NASA Ames Research Centre. It is used for diagnosis of faults in several components of the propulsion system in the Autonomous Lander Demonstration Project (ALDER), the Electrical Power System of the International Space Station (ISS), and the landing gear of an aircraft [140].

2.4 Reasoning for IVHM at the Vehicle level

Opportunities: The previous section discussed how reasoning is used for health monitoring by different sectors within the aerospace industry. However, among the health monitoring systems cited, the major focus was on the component or the systems level, but not at the vehicle level. This is an important shortcoming, considering that one of the main objectives of IVHM is to assess and predict the health of the system and its effect at the vehicle level and avoid unexpected downtime, thereby supporting CBM.

There are only a handful of studies that consider the notion of vehicle level as the primary driver. For example, Vehicle Integrated Prognostic Reasoner developed by a team from Honeywell, Vanderbilt University, and NASA Langley Research Centre integrates information from various aircraft subsystems, namely APU, Avionics, Flaps, and LRUs, to generate multiple evidence and simulate evidence and false alarms for other subsystems [141]. Dinkar et al [142] patented a Vehicle Level Reasoner Engine that consists of an Advanced Diagnostic and Prognostic Reasoner, and a System Fault Model which has fault models from components level, group area level, and subsystems level, and a set of diagnostic and prognostic algorithms, in order to provide diagnostic and prognostic conclusions, based on the observed failure modes. An evolvable tri-reasoner conceptual framework [143] was developed by Boeing, along with Impact Technologies, NASA and Vanderbilt University which consisted of reasoners for anomaly, diagnosis, and prognosis that provided health information of the vehicle which was used by the Reasoner Integration Manager as evidence to prioritize the faults based on their interaction with the integrated model. Khan et al [144] developed a Vehicle Level Reasoning System framework, that takes the health status of several subsystems like engine, fuel system and electrical system into account, and weighted health indices were calculated based on behavioural evidence and diagnosed faults. Schoeller et al, [54]

proposed a reasoning structure that had hierarchical layers of reasoning where, the health monitoring techniques are focused on component or LRU level, sub-system level, and system level. Failure mechanisms of multiple sub-systems of the main system like the engine, actuators and drive trains of an aircraft are monitored and overall functionality of the vehicle (aircraft) during simultaneous degradation of all the monitored failure mechanisms is calculated. Thanigaivelu et al [145] developed a methodology for the BIT Effectivity analysis, which is based on Failure Mode and Effect Summary, to calculate the probability of BIT fault detection and isolation in a control system considering its effect at the LRU, system and at the aircraft level.

Most work presented shows only the effect of a single component's or a sub-system's failure at the vehicle level. The effects of these failures on the other subsystems, i.e. the effect due to their interactions and their root causes, at the vehicle level are not widely explored. Only a very few works have so far considered the effect of subsystem interactions. For example, Lopez et al. [146] demonstrated the effect of a fault in fuel-cooled oil cooler on independent systems using a modular framework. However, the finding did not describe the effect at the vehicle level and focused mainly on the cross-system diagnostics. Similarly, Nwadiogbu et al. [147] from Honeywell International Inc. patented a vehicle health monitoring system architecture in which the DSS would choose to ignore the fault issues reported by the Environmental Control system when there is a fault in the engine. The drawback of this decision is that it runs the risk of ignoring the faults that originate in the Environmental Control System that is independent of the faults in the engine. Another such example is the Vehicle Health Monitoring algorithms developed for diagnosing in the flight control system of the space shuttle [148]. The multivariate algorithm is used with the parametric models to detect the fault onset by combining the prediction models of three subsystems: Guidance, Navigation and Control System, Propulsion system and Gimbal actuation and study the effect of four faults on these three systems. This study suggests that a vehicle level reasoner is required to find the root cause of the faults by combining the symptoms from the other systems as well as BIT data.

The number of examples in this section and their findings are evidence to the point that the area of diagnosis and prognosis of interacting faults requires considerable

exploration. On another note, it is worth noticing that even among these few numbers of works on vehicle level frameworks, reasoning seems to play a key role in assessing the effect of the system's fault in the most. Hence, it is safe to assume that reasoning, both as a system as well as a strategy, is an inevitable requirement for any work related to vehicle level health management. Any research in the detection and prediction of interacting faults is, thus, a great opportunity to harness the full potential of reasoning. At the vehicle level, reasoning could be used to make inferences with the information available from multiple systems, ranking priorities and solving conflicts. It could make use of the connectivity between the components and the systems, consider the health information from all the levels, i.e. at the component/ subsystem/ system levels (as shown in [Figure 2-9](#)), process the data at each stage, and give decisions on the root cause of the fault and their cascading effects [88].

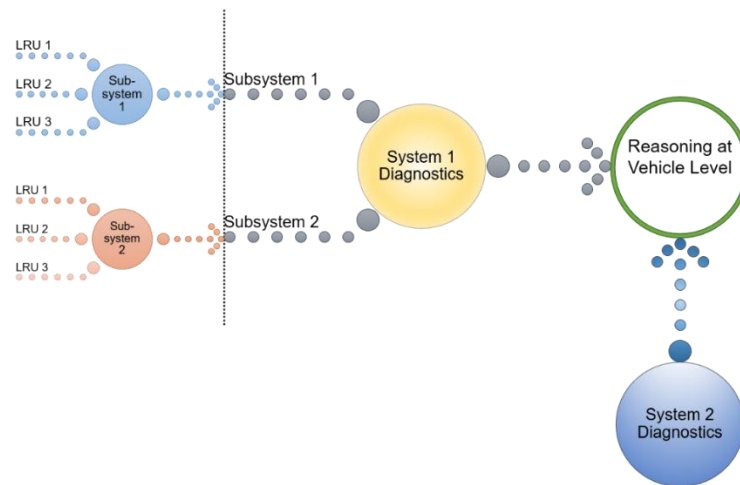


Figure 2-9 Information flow in reasoning at vehicle level

Challenges: The implementation of reasoning at the vehicle level is a challenging task and would face a lot of hurdles. One of the major reservations faced by IVHM is access to data. In general, the systems are manufactured by different suppliers, and the OEMs assemble these individual systems onto a common platform. Hence, there is a problem of intellectual property which prevents the suppliers from sharing data to the OEMs or the airlines. The other challenge is that these systems have their own methods of data processing and monitoring which may not be compatible with other systems' methods [149]. Hence, the integration of information from these systems at the vehicle level

requires a framework that could accommodate the processing of different types of data sources.

A reasoning system for vehicle level health management would require a robust framework that could clearly distinguish between the different layers of the vehicle and accommodate different data from multiple systems. Since the database is mainly dependent on sensor data, suitable algorithms have to be chosen for data processing, feature selection and extraction [44]. Correspondingly, methods to enable sensor data fusion and methods to mitigate the risk of sensor failure should be chosen as well [44]. A database management system is required, which could maintain all the different types of data, link their functionalities and make them accessible [44]. In order to mitigate the challenge of knowledge acquisition, the systems could use modular architecture to protect the IP protected and the private data. For this reason, suitable communication protocol has to be established between the database modules that possess the system health information, and the reasoning system developed at the vehicle level. Apart from this, suitable knowledge representation and problem-solving methods have to be chosen depending upon the type of interaction between the systems and the data sources. For example, if the reasoning system is being built for a vehicle that has rich data sources, the CBR methodology can be chosen to make use of the existing data as historical information. This would be an advantage to the reasoning system, in terms of developing the knowledge as well, as the CBR methodology allows learning through its revise and retain phases. Apart from these requirements, performance metrics like efficiency, false negatives, and false positives need benchmarking [44]. Lastly, developing a reasoning system for vehicle level health monitoring will require rigorous verification and validation for qualifying the system.

2.5 Summary and Conclusions

Following contributions are made in this literature review on reasoning in aerospace IVHM and its potential.

- i) A timeline comparison of the AI technologies used in the aerospace IVHM versus the AI technologies in general ([Figure 2-2](#)).

- ii) Applications of reasoning as an AI technology in the engineering field ([Figure 2-3](#))
- iii) Different reasoning strategies with examples ([Table 2-1](#)).
- iv) A generalized architecture for reasoning system ([Figure 2-4](#))
- v) Pros, cons, and applications of different knowledge-based reasoning systems ([Table 2-2](#)).
- vi) A SWOT analysis for model-based and data-driven methods ([Figure 2-8](#)).
- vii) Applications of reasoning in aerospace IVHM ([Table 2-3](#)).
- viii) Shortcomings in the existing health monitoring systems in aerospace IVHM and the opportunities for reasoning in vehicle level health management along with the potential challenges.

A fully realized IVHM system is required to face the high demand for CBM and reduce the MRO costs. Nevertheless, it is a challenging task, considering that the development of such an IVHM system depends on equally advanced supporting fields. This chapter attempts to learn the importance of the AI technology reasoning and its capability to contribute to the shortcomings in the aerospace IVHM field. To that effect, the paper reviews the reasoning system's architecture, its types, and functions along with its applications in various fields and explores the spaces where reasoning can further contribute to IVHM. It is identified from this paper that the vehicle level health monitoring system requires reasoning in any format: a strategy or a system. Likewise, the challenges involved in developing a reasoning system for vehicle level health monitoring is presented too and a few suggestions provided to mitigate some of them. The full potential of reasoning as both a strategy and a system can be harvested by combining with a suitable technology that has the capability of negating the challenges faced by reasoning. In identifying such potential technologies and making them function alongside reasoning lies the key to the future of fully functional IVHM systems.

References

1. Management OC. Preliminary Analysis of MCTF FY2017 Data Preliminary Analysis of FY2017 Data. 14th Maintenance Cost Conference. 2018. Available at:

<https://www.iata.org/whatwedo/workgroups/Documents/MCC-2018-ATL/Day1/1000-1030-prelim-analysis-of-mctf-data-iata.pdf>

2. Redding L. An Introduction to Integrated Vehicle Health Management - A Perspective from Literature. In: Jennions IK (ed.) Integrated Vehicle Health Management: Perspectives on an Emerging Field. SAE International; 2011. pp. 17–26. Available at: DOI:10.4271/R-05
3. Jennions IK. Introduction. In: Jennions IK (ed.) Integrated Vehicle Health Management: Perspectives on an Emerging Field. SAE International; 2011. pp. 1–8. Available at: DOI:10.4271/R-05
4. Wikipedia. Timeline of artificial intelligence. Available at: https://en.wikipedia.org/wiki/Timeline_of_artificial_intelligence (Accessed: 3 August 2018)
5. Wikipedia. Timeline of machine learning. Available at: https://en.wikipedia.org/wiki/Timeline_of_machine_learning (Accessed: 3 August 2018)
6. Schwabacher M., Samuels J., Brownston L. The NASA Integrated Vehicle Health Management technology experiment for X-37. Proc. SPIE 4733, Component and Systems Diagnostics, Prognostics, and Health Management II. Available at: DOI:10.1117/12.475494
7. Krichene A., Roemer M. Design Tool and Toolkits. In: Jennions IK (ed.) Integrated Vehicle Health Management: Perspectives on an Emerging Field. SAE International; 2011. pp. 77–88. Available at: DOI:10.4271/R-05
8. Tipaldi M., Bruenjes B. Spacecraft Health Monitoring and Management Systems. IEEE Metrology for Aerospace (MetroAeroSpace). IEEE; 2014. pp. 68–72. Available at: DOI:10.1109/MetroAeroSpace.2014.6865896
9. Laffey TJ., Cox PA., James L., Kao SM., Read JY. Real-Time Knowledge-Based Systems. AI Magazine. 1988; 9(1): 27–45.
10. Georgeff MP., Lansky AL. Reactive reasoning and planning. Proceedings of the sixth national conference on artificial intelligence (AAAI). 1987; : 677–682.

Available at:
<http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Reactive+reasoning+and+planning#0>

11. Tin Kam Ho. Random decision forests. Proceedings of 3rd International Conference on Document Analysis and Recognition. IEEE Comput. Soc. Press; pp. 278–282. Available at: DOI:10.1109/ICDAR.1995.598994 (Accessed: 27 November 2018)
12. Cortes C., Vapnik V. Support-Vector Networks. Machine Learning. October 1995; 20: 273–297. Available at: DOI:<https://doi.org/10.1007/BF00994018>
13. de Kleer J., Williams BC. Diagnosing multiple faults. Artificial Intelligence. 1987; 32(1): 97–130. Available at: DOI:10.1016/0004-3702(87)90063-4
14. Fudge M., Stagliano T., Tsiao S. Non-traditional flight safety systems & Integrated Vehicle Health Management systems. 2003. Available at: https://www.faa.gov/about/office_org/headquarters_offices/ast/licenses_permits/media/DO3_Report_final.pdf
15. Schwabacher M. A Survey of Data-Driven Prognostics. Infotech at Aerospace. AIAA; 2005. Available at: DOI:10.2514/6.2005-7002
16. Bresina JL., Jonsson AK., Morris PH., Rajan K. Mixed-Initiative Planning in MAPGEN: Capabilities and Shortcomings. Proceedings of the ICAPS-05 Workshop on Mixed-initiative Planning and Scheduling. 2005; : 8. Available at: [https://ti.arc.nasa.gov/m/pub-archive/960h/0960\(Bresina\).pdf](https://ti.arc.nasa.gov/m/pub-archive/960h/0960(Bresina).pdf)
17. Scandura PA., Christensen M., Lutz D., Bird G. Commercial Aviation Health Management. Systems Health Management: with Aerospace Applications. John Wiley & Sons; 2011. pp. 589–603.
18. Sheppard JW., Kaufman MA., Wilmering TJ. IEEE Standards for Prognostics and Health Management. IEEE A&E Systems Magazine. 2009; (September): 34–41. Available at: DOI:10.1109/AUTEST.2008.4662592
19. Brown ER., Worth F., Worth F., Worth F. Prognostics and Health Management A Data-Driven Approach to Supporting the F-35 Lightning II. 2007 IEEE Aerospace

- Conference. IEEE; 2007. pp. 1–12. Available at: DOI:10.1109/AERO.2007.352833
20. Airbus Customer Service. AiRTHM Airbus Real Time Health Monitoring. 2014. pp. 1–16. Available at: <https://www.iata.org/whatwedo/workgroups/Documents/MCC-2014-ATH/D2/1210-1250-realtime-health-monitoring-Airbus.pdf> (Accessed: 16 January 2018)
 21. David Reid (CNBC). Rolls-Royce is developing tiny ‘cockroach’ robots to crawl in and fix airplane engines. CNBC. 2018. Available at: <https://www.cnbc.com/2018/07/17/rolls-royce-is-developing-tiny-cockroach-robots-to-fix-airplane-engines.html> (Accessed: 26 November 2018)
 22. (Federal Aviation Administration). Fact Sheet – General Aviation Safety. 2018. Available at: https://www.faa.gov/news/fact_sheets/news_story.cfm?newsId=21274 (Accessed: 26 November 2018)
 23. Sleight PA., Carter RDG. AIRCRAFT ACCIDENT REPORT 1/2010. 2014. Available at: DOI:<https://doi.org/EW/C2008/01/01>
 24. Mack DLC., Biswas G., Koutsoukos XD., Mylaraswamy D. Using tree augmented Naïve bayes classifiers to improve engine fault models. CEUR Workshop Proceedings. 2011; 818(December): 91–98.
 25. Smith R. Aristotle’s Logic. Stanford Encyclopedia of Philosophy. 2017. Available at: <https://en.wikipedia.org/wiki/Reason> (Accessed: 21 August 2018)
 26. Gupta KM. Case-Based Troubleshooting Knowledge Management. AAAI Technical Report SS-99-04. 1999. Available at: DOI:10.1.1.582.2503
 27. Sugimatsu H., Hori S., Sawada A., Azuma K. A CBR Application: Service Productivity Improvement by Sharing Experience. IEEE Symposium on Emerging Technologies & Factory Automation. 1994; : 132–141. Available at: DOI:10.1109/ETFA.1994.402012
 28. Sycara K. Learning from failure experiences in case-based schedule repair.

- Proceedings of the Twenty-Seventh Hawaii International Conference on System Sciences HICSS. 1994. pp. 122–131. Available at: DOI:10.1109/HICSS.1994.323359
29. Hinkle D., Toomey CN. CLAVIER: Applying Case-Based Reasoning to Composite part fabrication. IAAI-94 Proceedings. 1994. pp. 55–62. Available at: <https://www.aaai.org/Papers/IAAI/1994/IAAI94-007.pdf>
 30. Johnson C. Software tools to support incident reporting in safety-critical systems. Safety Science. 2002; 40(9): 765–780. Available at: DOI:10.1016/S0925-7535(01)00085-6
 31. Woskov SM., Grimaila MR., Mills RF., Haas MW. Design considerations for a case-based reasoning engine for scenario-based cyber incident notification. Symposium on Computational Intelligence in Cyber Security (CICS). 2011. Available at: DOI:10.1109/CICYBS.2011.5949397
 32. Nicklas D., Grossmann M., Mínguez J., Wieland M. Adding high-level reasoning to efficient low-level context management: A hybrid approach. 6th Annual International Conference on Pervasive Computing and Communications, PerCom. IEEE; 2008. pp. 447–452. Available at: DOI:10.1109/PERCOM.2008.64
 33. Rymon R. Goal-Directed Diagnosis - Diagnostic Reasoning in Exploratory-Corrective Domains. IRCS Technical Reports Series. 187. 1993. Available at: https://repository.upenn.edu/ircs_reports/187/
 34. Lee J., Huber MJ., Durfee EH., Kenny PG. UM-PRS: An Implementation of the Procedural Reasoning System for Multirobot Applications. AIAA/NASA Conference on Intelligent Robots in Field, Factory, Service, and Space (CIRFFSS '94). AIAA; 1994. Available at: DOI:10.1.1.51.6247
 35. Jones J., Xiang Y., Joseph S. Bayesian Probabilistic Reasoning in Design. Conference on Communications, Computers and Signal Processing. Pacific Rim: IEEE; 1993. pp. 501–504. Available at: DOI:10.1109/PACRIM.1993.407312
 36. Lou HH., Huang YL. Hierarchical decision making for proactive quality control: System development for defect reduction in automotive coating operations.

- Engineering Applications of Artificial Intelligence. 2003; 16(3): 237–250.
Available at: DOI:10.1016/S0952-1976(03)00060-5
37. Thang M. Do SWL., Liu F. Mobile and Ubiquitous Systems : Computing, Networking, and Services. In: Zheng K, Li M, Jiang H (eds.) 9th International Conference, MobiQuitous. Springer; 2012. pp. 188–199. Available at: DOI:10.1007/978-3-642-40238-8
 38. Ma S. Automatically Modeling , Simulating and Explaining Physical Systems. 17th International Conference on Computers and Industrial Engineering. Elsevier Science Ltd; 1995. pp. 187–191.
 39. Robertson G., Watson I. A Review of Real-Time Strategy Game AI. AI Magazine. 2014; 35(4): 75–104. Available at: DOI:10.1609/aimag.v35i4.2478
 40. Rigolli M., Brady M. Towards a Behavioural Traffic Monitoring System. AAMAS '05 Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems. ACM; 2005. pp. 449–454. Available at: DOI:10.1145/1082473.1082542
 41. Khorasgani H., Biswas G. A combined model-based and data-driven approach for monitoring smart buildings. In: Zanella M, Pill I, Cimatti A (eds.) 28th International Workshop on Principles of Diagnosis (DX'17). Kalpa Publications in Computing; 2018. pp. 21–36.
 42. Lee NS. DM2: an algorithm for diagnostic reasoning that combines analytical models and experiential knowledge. International Journal of Man-Machine Studies. 1988; 28(6): 643–670. Available at: DOI:10.1016/S0020-7373(88)80066-X
 43. Reibling L a., Lafayette W. Diagnostic Reasoning Technology for the On-Board Maintenance System. Proceedings of the National Aerospace and Electronics Conference, NAECON. IEEE; 1993. pp. 930–936. Available at: DOI:10.1109/NAECON.1993.290818
 44. Vachtsevanos G., Goebel K. Basic Principles. In: Jennions IK (ed.) Integrated Vehicle Health Management: Perspectives on an Emerging Field. SAE

- International; 2011. pp. 55–65. Available at: DOI:10.4271/R-05
45. Stanjevk M., Vranos S. Using Truth Maintenance Systems - A Tutorial. IEEE Expert. 1994; 9(6): 46–56. Available at: DOI:10.1109/64.363270
 46. Marzat J., Piet-Lahanier H., Damongeot F., Walter E. Model-based fault diagnosis for aerospace systems: A survey. Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering. 2012; 226(10): 1329–1360. Available at: DOI:10.1177/0954410011421717
 47. Venkatasubramanian V., Rengaswamy R., Ka SN. A review of process fault detection and diagnosis Part II: Qualitative models and search strategies. Computers and Chemical Engineering. 2003; 27(3): 313–326. Available at: DOI:https://doi.org/10.1016/S0098-1354(02)00161-8
 48. Venkatasubramanian V., Rengaswamy R., Ka SN. A review of process fault detection and diagnosis Part III : Process history based methods. Computers and Chemical Engineering. 2003; 27(3): 327–346. Available at: DOI:https://doi.org/10.1016/S0098-1354(02)00162-X
 49. Forbus KD. Qualitative Reasoning. CRC Handbook of Computer Science. 1996; Available at: <http://www.qrg.cs.northwestern.edu/papers/Files/crc7.pdf>
 50. Patterson-Hine A., Biswas G., Aaseng G., Sriam N., Pattipati K. A review of diagnostic techniques for ISHM applications. 2005. Available at: [https://ti.arc.nasa.gov/m/pub-archive/1094h/1094 \(APH\).pdf](https://ti.arc.nasa.gov/m/pub-archive/1094h/1094%20APH.pdf)
 51. Chandola V., Banerjee A., Kumar V. Anomaly detection: A survey. ACM Computing Surveys (CSUR). 2009; 41(September): 1–58. Available at: DOI:10.1145/1541880.1541882
 52. An D., Kim NH., Choi JH. Practical options for selecting data-driven or physics-based prognostics algorithms with reviews. Reliability Engineering and System Safety. Elsevier; 2015; 133: 223–236. Available at: DOI:10.1016/j.ress.2014.09.014
 53. Tidriri K., Chatti N., Verron S., Tiplica T. Bridging data-driven and model-based approaches for process fault diagnosis and health monitoring: A review of

- researches and future challenges. *Annual Reviews in Control*. 2016; : 63–81. Available at: DOI:10.1016/j.arcontrol.2016.09.008
54. Schoeller M., Roemer M., Leonard M., Derriso M. Embedded Reasoning Supporting Aerospace IVHM. *AIAA Infotech at Aerospace Conference and Exhibit*. 2007. pp. 1–19. Available at: DOI:10.2514/6.2007-2820
 55. Ramirez C., Valdes B. A General Knowledge Representation Model of Concepts. *Advances in Knowledge Representation*. INTECH; 2012.
 56. Seitz C., Schönfelder R. Rule-based OWL reasoning for specific embedded devices. Lora, Aroyo;Chris, WeltyHarith; AlaniJamie, Taylor;Abraham, Bernstein;Lalana, KagalNatasha; NoyEva B (ed.) *The Semantic Web – ISWC 2011*. ISWC 2011. *Lecture Notes in Computer Science*. Springer, Berlin, Heidelberg; 2011. 237–252 p. Available at: DOI:10.1007/978-3-642-25093-4_16
 57. Liu B., Duan M., Zhao G. An Object Frame Knowledge Representation Approach for Fault Diagnosis Expert System. *International Conference on Future Computer Sciences and Application*. IEEE; 2011. pp. 74–77. Available at: DOI:10.1109/ICFCSA.2011.24
 58. Grosan C., Abraham A. *Intelligent Systems, A Modern Approach*. 1st edn. Kacprzyk J, Jain LC (eds.) *Intelligent Systems Reference Library*, Volume 17. Springer-Verlag Berlin Heidelberg; 2011. Available at: DOI:10.1007/978-3-642-21004-4
 59. Buchanan BG., Shortliffe EH. The Context of the MYCIN Experiments. In: Buchanan BG, Shortliffe EH (eds.) *Rule-Based Expert Systems The MYCIN Experiments of the Stanford Heuristic Programming Project MYCIN Experiments of the Stanford Heuristic Programming Project*. Reading, WA: Addison Wesley; 1984. pp. 3–19. Available at: <http://www.aaai.org/Papers/Buchanan/Buchanan03.pdf>
 60. Buchanan BG., Smith RQ. Fundamentals of expert system. *Annual review of computer science*. 1988. 23–58 p.
 61. (Mihailo Pupin Institute). *BEST: Blackboard-based Expert System Toolkit*. pp. 1–

653. Available at: <http://static.pupin.rs/2010/07/BEST.pdf>
62. Bankert RJ., Imam I., Rajiyah H. 5,633,800: Integrated Model-based Reasoning/Expert System Diagnosis for Rotating Machinery. United States Patent; 1997.
63. Ingrand FF., Rao AS., Georgeff MP. An Architecture for Real-Time Reasoning and System Control. *IEEE Expert*. 1992; 7(6): 33–44. Available at: DOI:10.1109/64.180407
64. Kolodner JL. An introduction to case-based reasoning. *Artificial Intelligence Review*. 1992; 6(1): 3–34. Available at: DOI:10.1007/BF00155578
65. Bonzano A. ISAC: a Case-Based Reasoning System for Aircraft Conflict Resolution. PhD Thesis, University of Dublin, Trinity College; 1998. Available at: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.159.2126&rep=rep1&type=pdf>
66. Althoff K-D. Case-Based Reasoning and Expert System. In: Agudo BD, Ian Watson (eds.) 20th International Conference on Case Based Reasoning ICCBR. Springer-Verlag Berlin Heidelberg; 2012. Available at: DOI:10.1007/978-3-642-32986-9
67. Saxena A., Wu B., Vachtsevanos G. A hybrid reasoning architecture for fleet vehicle maintenance - Artificial intelligence-based diagnostic approach with particular reference to dynamic case-based reasoning. *IEEE Instrumentation and Measurement Magazine*. 2006; 9(4): 29–36. Available at: DOI:10.1109/MIM.2006.1664039
68. Srivastava AN. Enabling the Discovery of Recurring Anomalies in Aerospace Problem Reports using High-Dimensional Clustering Techniques. *Aerospace Conference*. IEEE; 2006. Available at: DOI:10.1109/AERO.2006.1656136
69. Sheppard JW., Butcher SGW. A Formal Analysis of Fault Diagnosis with D - matrices. *Journal of Electronic Testing, Theory and Applications*. 2007; Available at: DOI:10.1007/s10836-006-0628-7
70. Chen Y., Chen YW., Xu X Bin., Pan CC., Yang JB., Yang GK. A data-driven approximate causal inference model using the evidential reasoning rule.

- Knowledge-Based Systems. Elsevier B.V.; 2015; 88: 264–272. Available at: DOI:10.1016/j.knosys.2015.07.026
71. Vohnout S., Kim B., Kunst N., Gleeson B., Wagoner R., Balaban E., et al. A Model-based Avionic Prognostic Reasoner (MAPR). Infotech@Aerospace 2012. Reston, Virginia: American Institute of Aeronautics and Astronautics; 2012. pp. 1–17. Available at: DOI:10.2514/6.2012-2421
 72. Daigle M., Foygel M. Model-based Diagnostics for Propellant Loading Systems. Aerospace Conference. IEEE; 2011. Available at: DOI:10.1109/AERO.2011.5747596
 73. Niculita O., Irving P., Jennions IK. Use of COTS Functional Analysis Software as an IVHM Design Tool for Detection and Isolation of UAV Fuel System Faults. Annual Conference of Prognostics and Health Management Society. 2012. pp. 1–18.
 74. Höfling T., Isermann R. Fault detection based on adaptive parity equations and single-parameter tracking. Control Engineering Practice. 1996; 4(10): 1361–1369. Available at: DOI:10.1016/0967-0661(96)00146-3
 75. Gertler JJ., Anderson KC. An Evidential Reasoning Extension to Quantitative Model-Based Failure Diagnosis. IEEE Transactions on Systems, Man and Cybernetics. 1992; 22(2): 275–289. Available at: DOI:10.1109/21.148430
 76. Chi L., Reuben K., Mba D. Diagnostics and prognostics using switching Kalman filters. Structural Health Monitoring. 2014; 13(3): 296–306. Available at: DOI:10.1177/1475921714522844
 77. Alves MRJ., Bizarria C de O., Galvão RKH. Trend analysis for prognostics and health monitoring. Brazilian Symposium on Aerospace Eng. & Applications. 2009;
 78. Yan W., Zhang B. Extended Kalman filter development in Lebesgue sampling framework with an application to Li-ion battery diagnosis and prognosis. European Conference of the Prognostics and Health Management Society. 2016. pp. 1–7. Available at: https://www.phmsociety.org/sites/phmsociety.org/files/phm_submission/2016/ph

mec_16_001.pdf

79. Bittner B., Bozzano M., Cimatti A. Automated synthesis of timed failure propagation graphs. Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI. 2016. pp. 972–978.
80. Howard L. An algorithm for diagnostic reasoning using TFPG models in embedded real-time applications. Systems Readiness Technology Conference AUTOTESTCON. IEEE; 2001. pp. 978–987. Available at: DOI:10.1109/AUTEST.2001.949477
81. Abdelwahed S., Karsai G., Biswas G. System Diagnosis using Hybrid Failure Propagation Graphs. The 15th International Workshop on Principles of Diagnosis. ISIS PUBLICATIONS; 2004. Available at: DOI:10.1.1.2.4683
82. Liu TS., Chiou SB. The application of Petri nets to failure analysis. Reliability Engineering and System Safety. 1997; 57(2): 129–142. Available at: DOI:https://doi.org/10.1016/S0951-8320(97)00030-6
83. Guia A. State Estimation and Fault Detection Using Petri Nets. Kristensen LM, Petrucci L (eds.) Applications and Theory of Petri Nets. Lecture Notes in Computer Science, vol 6709. Springer, Berlin, Heidelberg; 2011. Available at: DOI:https://doi.org/10.1007/978-3-642-21834-7_3
84. Sun J., Qin S., Song Y., Member S. Fault Diagnosis of Electric Power Systems Based on Fuzzy Petri Nets. IEEE Transactions on Power Systems. 2004; 19(4): 2053–2059. Available at: DOI:10.1109/TPWRS.2004.836256
85. Yang Q., Loparo KA. Model-based and data driven fault diagnosis methods with applications to process monitoring. Phd Thesis, Case Western Reserve University; 2004.
86. Yilin Z., Hongning L. Aircraft fault diagnosis based on case — Based reasoning and fault tree analysis integrated. International Conference on Electronics, Communications and Control (ICECC). IEEE; 2011. pp. 703–706. Available at: DOI:10.1109/ICECC.2011.6067560
87. Larsson JE. Diagnostic reasoning based on means-end models: Experiences and

- future prospects. *Knowledge-Based Systems*. Elsevier; 2002. pp. 103–110. Available at: DOI:10.1016/S0950-7051(01)00126-5
88. A.N.Srivatsava., R.W.Mah., R.Bharadwaj., D.Mylaraswamy. Vehicle-Level Reasoning Systems. In: Jennions IK (ed.) *Integrated Vehicle Health Management: The Technology*. SAE Publications; 2013. pp. 139–155. Available at: <http://books.sae.org/r-429/>
 89. Pearl J. *Bayesian Networks*. University of California, Los Angeles; 2000. Available at: http://ftp.cs.ucla.edu/pub/stat_ser/R246.pdf
 90. Lin Y., Zakwan S., Jennions I. A Bayesian Approach to Fault Identification in the Presence of Multi-component Degradation. *International Journal of Prognostics and Health Management*. 2017; 8. Available at: https://dspace.lib.cranfield.ac.uk/bitstream/handle/1826/11645/A_bayesian_approach_to_fault_identification-2017.pdf?sequence=1
 91. Mengshoel OJ., Darwiche A., Cascio K., Chavira M., Angeles L., Poll S., et al. Diagnosing Faults in Electrical Power Systems of Spacecraft and Aircraft. *Artificial Intelligence*. 2008; (650): 1699–1705. Available at: DOI:10.1.1.149.7338
 92. Kurtoglu T., Mengshoel OJ., Poll S. A Framework for systematic benchmarking of monitoring and diagnostic systems. *International Conference on Prognostics and Health Management, PHM*. 2008. Available at: DOI:10.1109/PHM.2008.4711454
 93. D.L.C. M., G. B., X.D. K., D. M., G. H. Deriving bayesian classifiers from flight data to enhance aircraft diagnosis models. *Annual Conference of the Prognostics and Health Management Society*. 2011. pp. 175–189. Available at: <http://www.scopus.com/inward/record.url?eid=2-s2.0-84920517370&partnerID=40&md5=ad0fac69dbdf8abdf471e2e41e9d4883>
 94. Mack DLC., Biswas G., Koutsoukos XD., Mylaraswamy D. Learning Bayesian Network Structures to Augment Aircraft Diagnostic Reference Models. *IEEE Transactions on Automation Science and Engineering*. 2017; 14(1): 358–369. Available at: DOI:10.1109/TASE.2016.2542186

95. Niculita O., Skaf Z., Jennions IK. The application of Bayesian change point detection in UAV fuel systems. *Procedia CIRP*. Elsevier B.V; 2014; 22(1): 115–121. Available at: DOI:10.1016/j.procir.2014.07.119
96. Vranes S., Stanojevic M., Subasic P., Vraneš S., Stanojević M., Subašić P. A Framework for Engineering Intelligent Control Systems. *Engineering Application for Artificial intelligence*. 1994; 7(4): 367–380. Available at: DOI:0952-1976(94)00027-1
97. Ross TJ. *Fuzzy Logic with Engineering Applications*. 2nd edn. John Wiley & Sons; 2005.
98. Lu Y., Chen TQ., Hamilton B. A Fuzzy Diagnostic Model and Its Application in Automotive Engineering Diagnosis. *Applied Intelligence*. 1998; 9(3): 231–243. Available at: <http://www.scopus.com/inward/record.url?eid=2-s2.0-0032206446&partnerID=tZOtx3y1>
99. Dash S., Rengaswamy R., Venkatasubramanian V. Fuzzy-logic based trend classification for fault diagnosis of chemical processes. *Chemical Engineering*. 2003; 27: 347–362. Available at: DOI:10.1016/S0098-1354(02)00214-4
100. Annis DS., Mosher DF., Roberts DD. A Fuzzy Reasoning Design for Fault Detection and Diagnosis of a Computer-Controlled System. *Engineering Application for Artificial intelligence*. 2008; 157–170(2): 339–351. Available at: DOI:10.1016/j.engappai.2007.04.007. A
101. Hamilton-wright A., Member S., Stashuk DW. Transparent Decision Support Using Statistical Reasoning and Fuzzy Inference. *IEEE Transactions on Knowledge and Data Engineering*. 2006; 18(8): 1125–1137. Available at: DOI:10.1109/TKDE.2006.132
102. Ruspini EH., Lowrance JD., Strat TM. Understanding evidential reasoning. *International Journal of Approximate Reasoning*. 1992; 6(3): 401–424. Available at: DOI:10.1016/0888-613X(92)90033-V
103. Hui KH., Lim MH., Leong MS., Al-Obaidi SM. Dempster-Shafer evidence theory for multi-bearing faults diagnosis. *Engineering Applications of Artificial*

- Intelligence. Elsevier Ltd; 2017; 57: 160–170. Available at: DOI:10.1016/j.engappai.2016.10.017
104. Verbert K., Babuška R., De Schutter B. Bayesian and Dempster–Shafer reasoning for knowledge-based fault diagnosis—A comparative study. *Engineering Applications of Artificial Intelligence*. Elsevier Ltd; 2017; 60: 136–150. Available at: DOI:10.1016/j.engappai.2017.01.011
 105. Srivastava RP. An introduction to evidential reasoning for decision making under uncertainty: Bayesian and belief function perspectives. *International Journal of Accounting Information Systems*. Elsevier Inc; 2011; 12(2): 126–135. Available at: DOI:10.1016/j.accinf.2010.12.003
 106. Zhang Z., Wang Y., Wang K. Fault diagnosis and prognosis using wavelet packet decomposition , Fourier transform and artificial neural network. *Journal of Intelligent Manufacturing*. Springer; 2013; 24: 1213–1227. Available at: DOI:10.1007/s10845-012-0657-2
 107. He M., He D., Bechhoefer E. Using Deep Learning Based Approaches for Bearing Fault Diagnosis with AE Sensors. *Annual Conference of the Prognostics and Health Management Society*. 2016. pp. 1–10.
 108. He QP., Wang J. Fault detection using the k-nearest neighbor rule for semiconductor manufacturing processes. *IEEE Transactions on Semiconductor Manufacturing*. 2007; 20(4): 345–354. Available at: DOI:10.1109/TSM.2007.907607
 109. Jung M., Niculita O., Skaf Z. Comparison of Different Classification Algorithms for Fault Detection and Fault Isolation in Complex Systems. *Procedia Manufacturing*. Elsevier B.V.; 19(2018): 111–118. Available at: DOI:10.1016/j.promfg.2018.01.016
 110. Tobon-mejia D., Medjaher K., Zerhouni N., Tripot G. Hidden Markov models for failure diagnostic and prognostic. *Prognostics and System Health Management Conference*. IEEE; 2011. pp. 1–8. Available at: DOI:10.1109/PHM.2011.5939488
 111. Li Z., Zhou S., Choubey S., Sievenpiper C. Failure event prediction using the Cox

- proportional hazard model driven by frequent failure signatures. IIE Transactions (Institute of Industrial Engineers). 2007; 39(3): 303–315. Available at: DOI:10.1080/07408170600847168
112. Ryhajlo N., Sturlaugson L., Sheppard JW. Diagnostic Bayesian networks with fuzzy evidence. AUTOTESTCON (Proceedings). 2013; : 242–249. Available at: DOI:10.1109/AUTEST.2013.6645075
 113. C C., N-H C., C-I H. Intelligent aircraft maintenance support system using genetic algorithms and case-based reasoning. International Journal of Advanced Manufacturing Technology. 2004; 24: 440–446. Available at: DOI:10.1007/s00170-003-1707-x
 114. Portinale L., Magro D., Torasso P. Multi-modal diagnosis combining case-based and model-based reasoning: A formal and experimental analysis. Artificial Intelligence. 2004; 158(2): 109–153. Available at: DOI:10.1016/j.artint.2004.05.005
 115. Eker ÖF. A Hybrid Prognostic Methodology and its Application to Well-Controlled Engineering Systems. PhD Thesis, Cranfield University. 2013. Available at: DOI:10.1017/CBO9781107415324.004
 116. James ML., Dubon LP. An autonomous diagnostic and prognostic monitoring system for NASA's Deep Space Network. Aerospace Conference Proceedings. IEEE; 2000. pp. 403–414. Available at: DOI:10.1109/AERO.2000.878248
 117. Mackey R., James M., Park H., Zak M. BEAM: technology for autonomous self-analysis. Aerospace Conference Proceedings (Cat. No.01TH8542). IEEE; 2001. Available at: DOI:10.1109/AERO.2001.931319
 118. Park HG., Cannon H., Bajwa A., Mackey R., James M., Maul W., et al. Hybrid Diagnostic System: Beacon-Based Exception Analysis for Multimissions-Livingstone Integration. 58th Meeting of the Society for Machinery Failure Prevention Technology. Pasadena, CA: JPL, NASA; 2004. Available at: <http://hdl.handle.net/2014/38232>
 119. Schumann J., Roychoudhury I., Kulkarni C. Diagnostic Reasoning using

- Prognostic Information for Unmanned Aerial Systems. Annual Conference of the Prognostics and Health Management Society. 2015. pp. 1–11.
120. Winston HA., Clark RT., Buchina G. AGETS MBR An Application of Model-Based Reasoning to Gas Turbine Diagnostics. *AI Magazine*. 1995; : 67–77. Available at: <https://pdfs.semanticscholar.org/c1a1/70c154c66fb57ee43a3963fa9d847e994e89.pdf>
 121. Mehdi G., Roshchin M., Runkler T. Internet of Turbines : An Outlook on Smart Diagnostics. Annual Conference of Prognostics and Health Management Society. 2017. pp. 1–7. Available at: https://www.phmsociety.org/sites/phmsociety.org/files/phm_submission/2017/phmc_17_043.pdf
 122. Belard N., Pencolé Y., Combacau M. A theory of meta-diagnosis: Reasoning about diagnostic systems. *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*. 2011. pp. 731–737.
 123. Gaudette T., Alwardt AL. The Boeing smart TPS application executive with MATLAB as a diagnostic reasoner. *AUTOTESTCON (Proceedings)*. IEEE; 2006. pp. 714–719. Available at: DOI:10.1109/AUTEST.2006.283754
 124. Lehane M., Halasz M., Orchard R., Wylie R., Zaluski M. Integrated Diagnostic System (IDS) for Aircraft Fleet Maintenance. *Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI)*. 1998; : 26–30.
 125. Bunus P., Isaksson O., Frey B., Munker B. RODON - A Model-Based Diagnosis Approach for the DX Diagnostic Competition. Available at: http://photon.isy.liu.se/dx09/papers/dx09_submission_dxc7.pdf
 126. Bunus P., Isaksson O., Frey B., Munker B. Model-based diagnostics techniques for avionics applications with RODON. *2nd Workshop on Aviation System Technology*. 2009. Available at: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.159.3003&rep=rep1&type=pdf>

127. (DSI international inc. . eXpress Embedded Reasoner. Available at: <https://www.dsiintl.com/products/diagnostic-reasoners/express-embedded-reasoner/> (Accessed: 10 June 2018)
128. Azam M., Ghoshal S., Deb S., Pattipati K., Haste D., Mandal S., et al. Integrated diagnostics and time to maintenance estimation for complex engineering systems. Aerospace Conference, Big SKy. IEEE; 2014. Available at: DOI:10.1109/AERO.2014.6836478
129. Mathur A., Cavanaugh KF., Pattipati KR., Willett PK., Galie TR. Reasoning and modeling systems in diagnosis and prognosis. Proceedings SPIE 4389, Component and Systems Diagnostics, Prognosis, and Health Management; Available at: DOI:10.1117/12.434239
130. Mackey R., Brownston L., Castle JP., Sweet A. Getting diagnostic reasoning off the ground: Maturing technology with TacSat-3. IEEE Intelligent Systems. 2010; : 27–35. Available at: DOI:10.1109/MIS.2010.124
131. (PHM Technology). made. 2017. Available at: <https://www.phmtechnology.com/assets/downloads/default/MADe-Overview.pdf> (Accessed: 24 June 2018)
132. Kalgren PW., Byington CS., Donovan BP. Automated Data Translation for Avionics Health Management through Intelligent Evidence Streams and Databases. The American Helicopter Society 60th Annual Forum. Baltimore, MD; 2004.
133. Byington CS., Kalgren PW., Johns R., Beers RJ. Embedded Diagnostic/Prognostic Reasoning and Information Continuity for Improved Avionics Maintenance. IEEE Systems Readiness Technology Conference. Proceedings AUTOTESTCON. 2003. pp. 518–524. Available at: DOI:10.1109/AUTEST.2003.1243595
134. Chidambaram B., Horn PL., Gilbertson DD., Pigg P., Talbot MA., Cerise KC. Recorders, Reasoners and Artificial Intelligence – Integrated Diagnostics on Military Transport Aircraft. Aerospace Conference. IEEE; 2006. pp. 1–9. Available at: DOI:10.1109/AERO.2006.1656074

135. Keller K., Amo A Del., Jordan B. Aircraft Electrical Power Systems Prognostics and Health Management (AEPHM). SAE Technical Papers Series. 2004; Available at: DOI:2004-01-3162
136. Balaban E., Saxena A., Narasimhan S., Roychoudhury I., Koopmans M., Ott C., et al. Prognostic Health-Management System Development for Electromechanical Actuators. *Journal of Aerospace Information Systems*. 2015; 12(3): 329–344. Available at: DOI:10.2514/1.I010171
137. Simon G., Karsai G., Biswas G., Abdelwahed S., Mahadevan N., Szemethy T., et al. Model-Based Fault-Adaptive Control of Complex Dynamic Systems. IMTC-2003 - Instrumentation and measurement Technology Conference. IEEE; 2003. pp. 20–22.
138. Kuijpers EA., Gelder P van., Pietersma Y. Automated on-board model-based diagnosis during planetary mission and teleoperation. 2008. Available at: <https://core.ac.uk/download/pdf/53034129.pdf>
139. Sebastian RK., Peripinayagam S., Jennions IK., Alghassi A. Prognostic reasoner based adaptive power management system for a more electric aircraft. *Proceedings of the Annual Conference of the Prognostics and Health Management Society*. 2016. pp. 1–11.
140. Narasimhan S. Automated diagnosis of physical systems. *Proceedings of ICALEPCS07*. pp. 701–705. Available at: <https://accelconf.web.cern.ch/accelconf/ica07/PAPERS/FOAA01.PDF>
141. Bharadwaj, Raj; Mylaraswamy, Dinkar; Cornhill, Dennis; Biswas G., Koutsoukos, Xenofon; Mack D. Vehicle Integrated Prognostic Reasoner (VIPR) Final Report. NASA Langley Research Centre; 2013. Available at: <https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20130011181.pdf>
142. Mylaraswamy D., Hadden GD. US 8,285,438 B2: Methods Systems and Apparatus for Analyzing Complex Systems via Prognostic Reasoning. 2012. Available at: DOI:10.1021/n10602701.
143. Atlas L., Bloor G., Brotherton T., Howard L., Jaw L., Kacprzyński G., et al. An

- evolvable tri-reasoner IVHM system. IEEE Aerospace Conference Proceedings. 2001. pp. 63023–63037. Available at: DOI:10.1109/AERO.2001.931322
144. Khan F., Eker OF., Sreenuch T., Tsourdos A. Health Index and Behaviour Based Vehicle Level Reasoning System. Conference on Prognostics and Health Management (PHM). IEEE; 2015. Available at: DOI:10.1109/ICPHM.2015.7245042
 145. Thanigaivelu V., Choksi S. On the Use of Built In Test Effectivity Analysis. SAE Technical Paper 2017-01-2051. 2017; Available at: DOI:10.4271/2017-01-2051
 146. López BS., Siddiolo AM., Adhikari PP., Buderath M. A Bayesian paradigm for aircraft operational capability assessment and improved fault diagnostics. European Conference of the Prognostics and Health Management Society. 2016. pp. 1–14.
 147. Nwadiogbu EO., Mylaraswamy D., Menon S., Voges HC., Hadden G. US008346429B2: Vehicle Health Monitoring System Architecture for Diagnostics and Prognostics Disclosure. United States Patent; 2013. Available at: DOI:10.1016/j.(73)
 148. Gorinevsky D., Samar S., Bain J. Integrated Diagnostics of Rocket Flight Control. IEEE Aerospace Conference Proceedings. IEEE; 2005.
 149. Khan F., Jennions I., Sreenuch T. Integration issues for vehicle level distributed diagnostic reasoners. SAE Technical Papers. 2013; Available at: DOI:10.4271/2013-01-2294
 150. Goebel K., Bonanni P., Eklund N. Towards an integrated reasoner for bearings prognostics. IEEE Aerospace Conference Proceedings. 2005; : 1–11. Available at: DOI:10.1109/AERO.2005.1559670

3 Understanding the role of a Digital Twin in Integrated Vehicle Health Management (IVHM)

Integrated Vehicle Health Management (IVHM) aims to support Condition-Based Maintenance (CBM) by monitoring, diagnosing, and prognosing the health of the host system. One of the technologies required by IVHM to carry out its objectives is the means to emulate the functioning of the host system, and the concept of a Digital Twin (DT) was introduced in aerospace IVHM to represent the functioning of such a complex system. This paper aims to discuss the role played by DT in the field of IVHM. A DT is the virtual representation of any physical product, that is used to project the functioning of the product at a given instance. The DT is used across the lifecycle of any product, and its output can be customized depending upon the area of application. The DT is currently popular in industry because of the technologies like sensors, cloud computing, Internet of Things, machine learning, and advanced software, which enabled its development. This paper discusses what encompasses a DT, the technologies that support the DT, its applications across industries, and its development in academia. This paper also talks about how a DT can combine with IVHM technology to assess the health of complex systems like an aircraft. Lastly, this paper presents various challenges faced by industry during the implementation of a DT and some of the possible opportunities for future growth.

3.1	Introduction
3.2	Building Blocks of a Digital Twin
3.3	Applications of a Digital Twin
3.4	Digital Twin and IVHM
3.5	Challenges, Gaps and Opportunities
3.6	Conclusion

Figure 3-1 Overview of Chapter 3

3.1 Introduction

The recent trend of using the concept of a Digital Twin (DT) in any industry from aerospace to agriculture, from a product-oriented environment to a process or a service-oriented environment, is due to its versatile nature and its ability to provide a solution for any given scenario. The term ‘Digital Twin’ was initially coined by Dr. Michael Grieves from the University of Michigan in 2002 to represent the digital equivalent of a physical asset [1]. Later, in 2012, NASA adapted DT as an integrated multi-physics, multi-scale probabilistic simulation model of a vehicle or a system that integrates best available physical models, sensor data from onboard health monitoring systems, fleet maintenance history and other such information to mirror the life of its physical twin [2]. Bolstered by the parallel growth in sensors and communication technologies, cyber-physical systems, computational speed, and artificial intelligence (AI), the term DT has been subsequently used by industry for several applications. A Gartner survey in 2019 mentions that around 75% of the industries that use Internet-of-Things (IoT) are either using DT already or plan to use it within the next year [3]. With the given advanced technologies, the DT is currently the key to optimize the product design process, to enable smart manufacturing with Industry 4.0, to streamline the process of Maintenance Repair and Overhaul (MRO) and to reduce the overall cost involved in the product and the process life cycles.

This paper focuses on how the DT can be combined with Integrated Vehicle Health Management (IVHM) to enable Condition-Based Maintenance (CBM) of vehicles like an aircraft. CBM is a program that provides maintenance plans based on the condition of the vehicle [4]. It is a real-time product performance measure, which has replaced the conventional time-based programs like scheduled inspections and preventive maintenance [5]. CBM enables customized maintenance planning of the targeted components and systems of a vehicle by monitoring their health condition. This results in saving time and cost involved due to unplanned downtime and increases the availability of the vehicle. A fully functional IVHM system assists in acquiring and analysing health data of a vehicle to optimize the suitable maintenance plans for CBM [6].

Integrated Vehicle Health Management (IVHM) is defined as the technology that uses the sensor data from a vehicle to carry out diagnosis and prognosis to assess and predict the health of its systems. IVHM technology was initially introduced by NASA to collect

data, diagnose, predict and mitigate faults, support operational and post-maintenance activities in space vehicles. Later, it expanded to other vehicles like aircraft and ships and evolved as a paradigm-shift to support Condition-Based Maintenance (CBM) of the vehicles. IVHM aims to ensure that the host system functions as intended, without any unexpected failure [6]. With the help of IVHM systems, the maintenance cost can be brought down significantly by predicting the system health in advance and supporting CBM.

Figure 3-2 shows the general role of IVHM technology in a product's (vehicle) lifecycle [7]. In this figure, the steps involved in IVHM are explained via the Sense-Acquire-Transfer-Analyze-Act (SATAA) cycle as shown in the right-hand side, and its role concerning the product's lifecycle is shown in the left-hand side. To monitor the health of the system, IVHM uses state-of-the-art sensors to acquire health information from the system. It also gets input from various stages of the product's lifecycle, like design-engineering, manufacturing and maintenance-logistics.

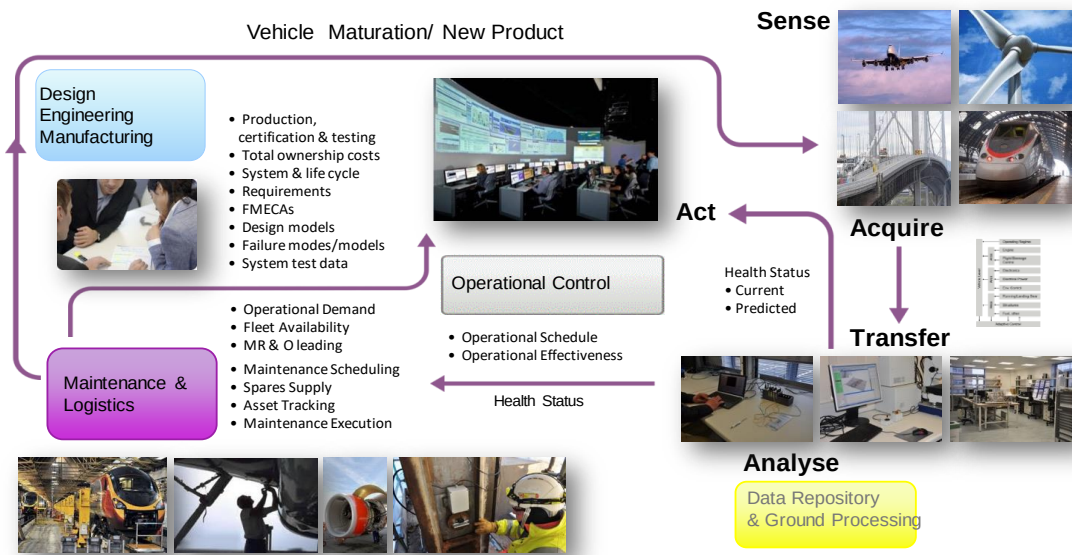


Figure 3-2 The role of IVHM in product lifecycle [7]

Once the data is acquired, they are transferred using high-end communication technologies to analyze the health information collected from the data. The IVHM system needs high computational speed for processing all the data collected and depends upon advanced machine learning algorithms and AI for diagnosing faults and predicting the

remaining useful life of the systems. This information on the system's health is provided for maintenance of the vehicle (as shown in [Figure 3-2](#)), under the CBM program.

For the diagnostics and prognostics to produce effective results, the IVHM system requires a knowledge-rich representation of the host system, that could provide all possible outcomes of multiple health scenarios concerning the host system [5]. It is at this capacity that the Digital Twin (DT) comes into the picture of IVHM. The DT is a high-fidelity simulation, that can provide the virtual representation of the physical asset at a particular instant and produce what-if simulations for a variety of scenarios depending upon the data acquired [8]. This feature of DT helps in forecasting the health of the systems that could be used by IVHM. In the last five years, many aerospace and automotive industries have started using DT at various stages of a product lifecycle.

This paper explores the field of Digital Twin (DT), its design, and its different applications in IVHM. To that effect, this section introduced the fields of DT, CBM, and IVHM. Section 3.2 presents the core elements of a DT, its functions alongside the various stages of a product's lifecycle, and its enabling technologies. Section 3.3 surveys some of the applications of DT in each stage of the product's lifecycle to understand its capabilities. Section 3.4 presents a basic IVHM system and the role of DT inscribed in it. It also talks on the developments of DT concerning the field of IVHM. It is to be noted that the majority of applications presented are from aerospace, and only a few are from other vehicles. Section 3.5 discusses various challenges faced during the implementation of DT, along with gaps and opportunities in IVHM. Section 3.6 presents the conclusion regarding the role of a DT within the field of IVHM.

3.2 Building Blocks of a Digital Twin

At its core, the DT is defined as a simulation model representing a physical asset, but unlike the usual simulation model, the DT evolves with the life of the physical asset, factoring in age-related performance and failure. It functions as a living virtual representation of the asset. It relates to the data captured from different stages of the asset's life cycle to create the contextual environment under which the asset is operating, and its output varies based on the services expected from it. Across the literature, when a DT is referred to, it could either mean the virtual representation of a product at an instant or encompass its entire life cycle [1]. Sometimes, it also refers to the entire environment

involving the physical, virtual models, data connections between them, and the nature of output [9]. Similarly, the number of building blocks of a DT also differs with the literature, ranging from three ([1], [10], [11]) to five [9], but they essentially have the same elements listed in a different order of importance.

This chapter refers to the DT as the living representation of a physical product and the Digital Twin Ecosystem [12] as the environment that includes the other elements that help in the functioning of the DT. A Digital Twin Ecosystem is thus built upon the following blocks as shown in Figure 3-3: i) the product life cycle at the top layer of Figure 3-3, ii) a virtual model or the DT at the bottom layer, and iii) a connecting segment as the middle layer. The connecting segment is characterized by data that relates the product as a physical asset to the DT as its corresponding virtual model. The transfer of data between the top and the bottom layer via its middle layer is made possible with the help of several technologies which are listed in the right-side middle layer of Figure 3-3. The following subsections discuss these central elements of a DT in detail. It is to be noted that the MRO section of Figure 3-3 is highlighted, as this chapter focuses on a DT that could be used mainly for IVHM.

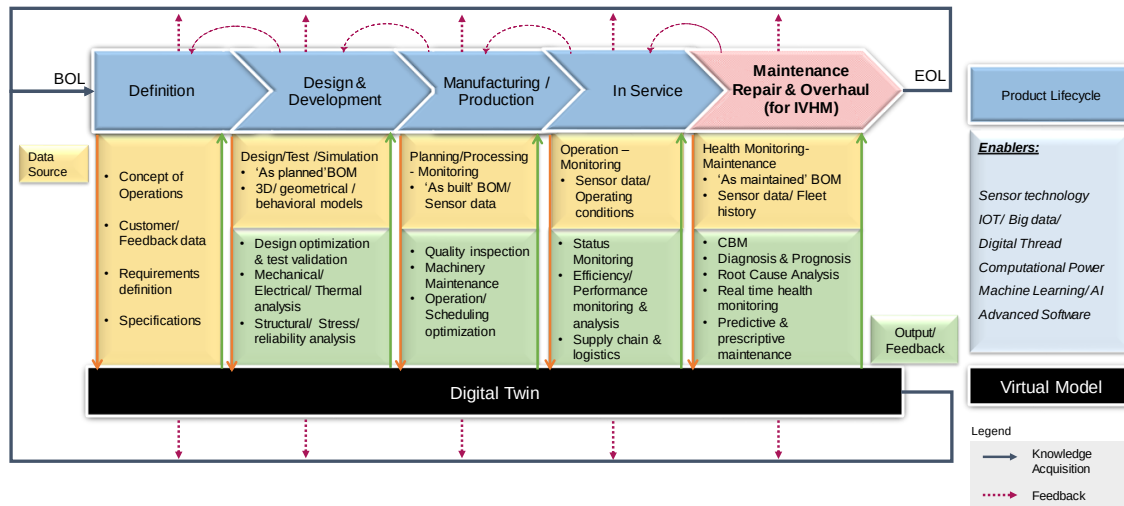


Figure 3-3 Digital Twin Ecosystem along the product lifecycle

3.2.1 Product Lifecycle

As shown in the top layer of Figure 3-3, from a product's Beginning of Life (BOL) to its End of Life (EOL), it goes through various stages during its life cycle. The product can be anything from a particular component of a system (e.g., compressor blade in an aircraft

engine) or an entire system (e.g., an aircraft engine) itself. The product information passes from its BOL to the EOL, and the data generated during the lifecycle, from concept definition, design and development, and manufacturing are used as a source of the virtual model for the DT. They also serve to validate the virtual model, which can later be considered as the reference model, against which the performance and health of the product can be measured when it is in use.

3.2.2 Virtual model

The virtual representation/ DT of a product is a highly adapted simulation model (bottom layer of [Figure 3-3](#)). The DT evolves alongside its physical asset based on the knowledge gained at each stage. It is the virtual representation of the physical asset at an instant and reflects on its operational behaviour. It utilizes various sources ranging from geometric or 3D models, thermal or CFD models, physics-based or functional models to represent the functioning of the product [13]. It also uses data like Bill of Materials (BOM) from ‘as-planned’, ‘as-built’, and ‘as-maintained’ states of the product, historical information, fleet data, and maintenance records to update the contextual knowledge of the virtual model and its environment to represent the actual working condition of the product through its lifecycle ([Figure 3-3](#)).

3.2.3 The Connecting segment

As mentioned earlier, the connecting segment (the middle layer of [Figure 3-3](#)) is characterized by the data that connects physical and virtual models and by the output/feedback from the DT. Data generated from the onboard sensors, inspections, fleet histories, and other sources during various stages of the lifecycle must be fused with the existing virtual models in order to update the DT at that stage of the product lifecycle. This data fusion within the DT ecosystem enables better model performance, adaptiveness, and robustness in the virtual models [12].

In the middle layer of [Figure 3-3](#), the data with the yellow background are input to the DT from the product and the data with the green background are the output from the DT to the product at a certain stage of its lifecycle. The output and feedback from the DT are dependent on the application. In the case of IVHM, the DT is expected to provide some of the results required to detect anomalies, health degradation, faults, and calculate the

remaining useful life of the product. These outputs also change with respect to the lifecycle of the product (Figure 3-3). For example, at the design stage, the DT is more useful in validating and optimizing the design process, whereas, in the manufacturing stage, the DT is used for smart factories, to visualize and streamline the process, to increase efficiency and to resolve conflicts on the shop floor [14]. The information from all stages of the product's lifecycle and the DT is sent as feedback (dotted lines in Figure 3-3). This feedback information is used to generate knowledge that is required for the optimization of the product as well as the process. The connecting segment, i.e., the transfer of data between the physical and the virtual model is enabled by several advanced technologies, without which, it would be impossible to implement the DT.

3.2.4 Enabling Technologies

Some of the technologies that helped the successful implementation of DT (as shown in the right-side middle layer of Figure 3-3) are as follows:

Sensors: The advancement in the field of sensor technologies is one of the primary enablers of any DT. Currently, there is a wide range of sensors at a lower price than in the past, and they produce abundant data, giving rise to Big Data. This data, if processed correctly, could provide information and the knowledge required to make the virtual model live.

Connectivity: Internet-of-Things (IoT) and cloud computing are two disruptive technologies that changed the perspective of every industry. IoT enables connectivity between various systems with an IP address, which makes it possible for data collection, and processing; it is one of the critical factors required for real-time monitoring of any process. On the other hand, cloud computing allows large-scale data processing and storage in the servers, making them accessible by the internet. It makes scalable computing available at any point, which is not possible through the local servers. Edge computing carries out computation at the end of the local network before updating to the cloud, thus reducing the burden on the server [15]. A Digital thread establishes the link between the product and the DT and is expected to create a value stream of data across its lifecycle [16].

Computational Power: The improvement in computational power and processing speed are two of the main influential factors for Big Data processing and cloud computing. Powerful processors enable real-time updates required for the DT and reduce the time taken to solve the scenarios involving complex cases.

Machine learning and Artificial Intelligence (AI): The advancement in sensors and the increased amount of data generated resulted in rekindling of machine learning techniques and AI from the past decades to complement the current technologies [6]. The input domain knowledge required by the DT, to be in line with the system's performance, can be in the form of rules, ontologies, procedures, models or even sensor data [6], depending on the compatibility with the DT model. The output is generated by the DT model with the help of reasoning systems built around it and the advanced machine learning algorithms used. With the support of high processing speeds, machine learning algorithms can provide solutions in a shorter time, thus improving the efficiency of the entire problem-solving process.

Software: The progression in the data analytics software helps to convert the results produced by the DT into user-friendly visualizations, that encourage industry to opt for the DT approach for its products. Similarly, multiple platforms have made it easier for developing various models for DT, from designing the products to monitoring the manufacturing process and managing the maintenance activities.

The abovementioned technologies have made it possible for the broad application of DT in the industry, especially for the high-value products. The synergy among these technologies has helped increase efficiency and save cost and time in every stage of a product's lifecycle. For example, considering a DT used in a manufacturing plant, the synergy among the enabling technologies will help identify the backlogs in the plant and also provide real time status warning, thus helping in increasing the efficiency and availability of the plant, reducing the maintenance downtimes and eventually save cost involved in unexpected backlogs and downtimes. More examples of the benefits of the DTs can be found in the next subsection. It is to be noted that these enabling technologies also influence IVHM as described in Section 3.1. This shows that the implementation of DT in IVHM would not require any drastic infrastructure change, rather a framework that provides state-of-the-art technologies that would benefit both the fields.

3.3 Applications of A Digital Twin

As stated before, the DT is a high-fidelity simulation model. It is capable of producing what-if scenarios for any given situation when provided with necessary data. In general, a DT serves as a virtual sensor that represents the product or the process at any instant and also as a feedback loop, providing data for further improvement. The DT is as powerful as the domain knowledge and specifications built into it, and as adaptive as the data it is supplied with. In general, the DT is adapted by the industries involved with high valued or highly customized products and processes due to the huge investment required to set up the infrastructure. With the help of enabling technologies mentioned in the previous section, industries have used DT for a variety of applications, adapted it to various scenarios. This section delivers some applications of DT developed in academia as well as by industry, based on the product's lifecycle stages, to present the versatile nature of the DT.

3.3.1 Digital Twin in design & development

In the design and development phase of a product, the DT is generally used for testing, verification and validation (V&V) purposes, as well as to structural/ thermal and other analysis on the design. For example, Rios et al. [13] along with Airbus, used Dassault Systems V6 solutions to propose a framework for developing a DT model for an aircraft during the industrial design. The framework consisted of datasets generated from product-process-resources-systems, linked with each other, to provide an avatar of a dummy aircraft DA08. Data from 'as-defined', 'as-planned', 'as-prepared', 'as-built' and 'as-maintained' were linked along with the resources like 3D and simulation models and technical publications to develop the Digital Mock-Up of an aircraft assembly. Another example of the DT used in the design and development phase is its application with the help of CAE models by Siemens. In this case, the DT was used for various functions: i) predicting aerodynamic performance, blade deflection, and cooling flow rates of gas turbine blades, ii) predicting acoustic and structural loading, optimizing noise and vibration in e-powertrain, and optimizing range and performance of electric vehicles, iii) predict fuel consumption of excavators for soil loading, and iv) enable V&V of autonomous test driving [17]. Siemens has used the DT for Model-based system testing that connects the virtual testing model with bench testing and field-testing data to enable

attribute specific evaluation throughout the development cycle. Siemens also uses the DT to support the testing process through anomaly detection process monitoring, and fault diagnosis [17].

3.3.2 Digital Twin in Manufacturing

The DT is used widely in manufacturing to monitor and optimize the production, to perform quality control on the products and predict defects in the assembly lines. Zheng et al. [18] developed a DT model of a welding production line with a 3D geometric, physical and kinematic model and connected it with the data from the production line to achieve interrelationship between the model and the real-time operation. This was used to monitor the welding production line to analyse the operation time and production cycle at each station and produce a real-time status warning. Similarly, Cai et al. [19] developed a DT of a vertical milling machine by data acquisition and information fusion techniques. They developed the DT with the characteristics of the machine using manufacturing data and sensory data from Hall-effect current and vibration sensors to monitor the operating conditions and to demonstrate the prediction capability for surface roughness model.

An industry-implemented DT use case is the 'Meggitt Modular Modifiable Manufacturing' or M4 [20] that explores the technologies of digital manufacturing, cloud computing, and IoT to address Meggitt's challenges of lack of flexibility and automation in the production lines that assemble aerospace products. The M4 framework combines the key components of the shop floor with sensors and connects them with data storage facilities, to carry out analysis required for the production line. The M4 framework was demonstrated by assembling the final stages of three critical safety-critical aerospace products, by carrying out what-if scenarios, optimizing the schedules, automating the part movements, and analysing the factory performance at the end [20]. Likewise, Boeing uses Dassault Systems' 3D EXPERIENCE platform in its 777X composite wing centre [21]. This new digital factory would help in streamlining the entire supply chain process by providing the real-time status of the factory, to optimize the plan for parts movement to match with the key performance indicators, to produce 'what-if' scenarios and compare the different methods for optimization, and to communicate with all the stakeholders about the updated plans for the movement. This will also help in loading and capacity

simulation, decision making, and scheduling for parts production, and factory's performance management [21].

3.3.3 Digital Twin in in-service monitoring

The DT can be used to monitor the in-service performance of the product, that could give feedback to both maintenance as well as design stages. For example, Wang et al. [10] developed a DT for rotating machinery with a Finite Element Analysis model, along with a model updating strategy to accommodate for sensitivity between the physical and the digital model. They demonstrated fault diagnosis of the unbalanced condition of the rotor due to the varying operating conditions and environment. An industrial application of the DT in in-service monitoring is the Abrams tank track system. The DT was created by Endurica, along with TARDEC for the T-158LL back-pad of the Abrams tank track system and implemented via Abaqus solver and Endurica TL solver [22]. This DT was used to demonstrate the virtual evaluation of complex in-service conditions like self-healing, fatigue fault modes, thermal runaway failure mode, and the operating temperature; the results were found to be close to the experimental results of the back-pad component testing [22]. On the other hand, GE has built DT of the powerplants with physics-based and data-driven models to detect faults to improve the asset failure mode management and reduce unplanned downtime [15]. It also has built thermal DTs to assess the thermal efficiency and predict emission, as well as lifing DT to optimize the mission versus maintenance reliability of the power plant. GE uses its Predix platform to run the DTs. Predix is the key platform that ingests large scale data generated, carries out data processing, uses analytical models, and provides services via DTs at high execution speed [15]. Similarly, Sandvik Coromat has used Microsoft solutions like Azure IoT suite, Corona Intelligence and Dynamic 365 to develop a scalable DT in its production flow to connect to optimize the field service provided for the customers [23].

The examples described in this section show the versatile nature of the DT and that it is used by the high-value industries throughout a product's life cycle, in a variety of applications, ranging from digital mock-up to digital factories, and from predicting failure life to evaluating performance efficiency. The applications cited in this section are mainly from design and development, manufacturing, and in-service monitoring stages of the

product life cycle. The following section discusses the applications of DT in maintenance, focusing exclusively on IVHM.

3.4 Digital Twin and IVHM

As mentioned earlier, IVHM is the paradigm shift to support CBM for vehicles. An IVHM system aims to assess the health of its host system, isolate faults, and predict useful life. The functioning of an IVHM system is influenced by the Open System Architecture for Condition Based Maintenance (OSA-CBM) framework. The OSA-CBM framework defines a series of steps (as shown in the middle layer of Figure 3-4), that could be carried out to provide a maintenance solution which is adapted by the IVHM system [5]. Figure 3-4 presents the basic process of an end-to-end IVHM system framework for an aircraft with the DT's role inscribed. The middle layer of Figure 3-4 shows the steps followed by an IVHM system as defined by OSA-CBM. It starts with collecting data from the aircraft platform regarding a concerned subject which could either be a component (e.g., compressor blade), or a subsystem (e.g., gearbox assembly), or a system (e.g., engine). It then employs several data processing and manipulating techniques to extract information. This information is later used to assess health of the subject, detect the presence of any faults, diagnose and isolate faults, and predict the failure pattern and the subject's remaining useful life. The outcome from these three stages, i.e., condition monitoring, diagnosis, and prognosis, are sent for generating advisories which are then used by CBM to schedule maintenance programs for the concerned subject accordingly (bottom layer of Figure 3-4).

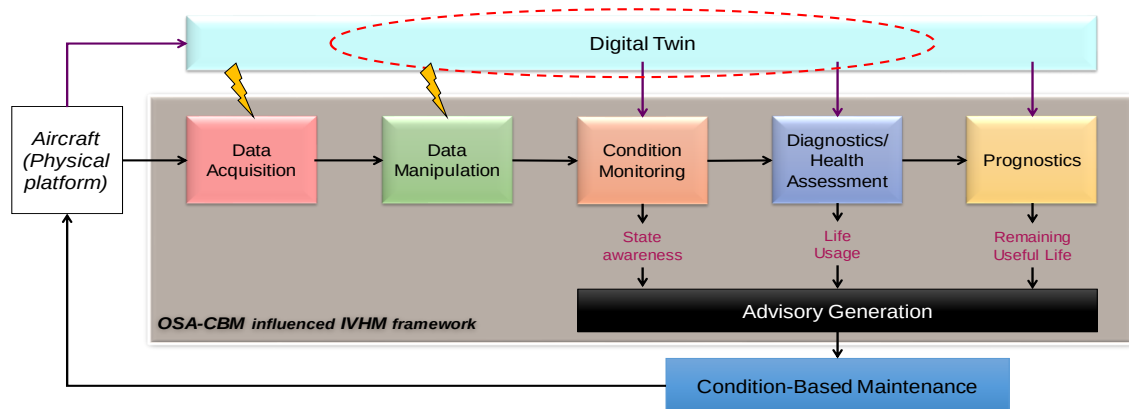


Figure 3-4 Digital Twin in an IVHM framework (adapted from [5])

To validate and ensure reliability of the results generated by the algorithms used for these three stages, a DT that is validated against the aircraft, can be used as an alternate representation of the aircraft platform within the IVHM system (top layer of [Figure 3-4](#)). To update the DT with the latest state of the aircraft's concerned subject, the data from the aircraft platform would undergo the same process of data acquisition and manipulation, similar to the initial steps followed by the IVHM system. This processed data, along with the context, could help in verification and optimization of the condition monitoring logic. Another advantage of inscribing DT to the IVHM system is that the DT would ideally possess knowledge of the entire system since its design stage (as discussed in Section 3.2). One of the existing gaps in the field of IVHM is that the IVHM system is currently treated as an add-on to a matured design process of the asset [24]. The IVHM system is pointed to be more beneficial, if it is included at the earliest stage of the asset design process, strengthening the links of requirements and knowledge between the asset and the IVHM system. Hence, implementing DT, which represents the functioning of the physical asset, along with its contextual knowledge, would improve the overall performance of the IVHM system. One more advantage is the 'what-if' simulations that can be run in the DT at that point will augment the diagnosis and prognosis process. The better the results from the IVHM system, the better maintenance plans generated in CBM.

The DT, when first implemented by NASA, was planned to function with onboard IVHM systems to get health monitoring data to forecast the life of the systems and update maintenance schedules accordingly. Over the years, the role of DT evolved to supply chain streamlining, exterminating silos, and enabling visibility to stakeholders. However, the core of DT remains the same, i.e., to emulate the functioning of its physical counterpart, thus showcasing its health status. Even if a DT is used mostly in design and development or in the manufacturing stage of a product, it continually provides data in these stages that could be used for health monitoring eventually. Hence, the implementation of DT in the framework has become a natural step forward for the IVHM field. Some applications that use DT in IVHM are discussed in the rest of this section.

NASA and the United States (US) Airforce have developed several DTs for vehicle health monitoring. Tuegel [25] proposed an Airframe digital twin that would be a cradle-to-grave model for designing and maintaining airframes. The Airframe digital twin helped

in finding the probability of non-conformances during the design phase, which helped in keeping records of the individual aircraft through its dedicated computational model. This also helped in forecasting different scenarios of the airframe's performance and failure and predict the retirement schedule depending upon the individual aircraft's performance, rather than depending on the reliability statistics of the whole fleet. The uncertainty between the physical airframe and its DT was reduced by updating the service experience through Bayesian method. Another example is from Zakrajsek et al. [26], who developed a physics-based DT of an aircraft tire to monitor its health at the aircraft touchdown. In order to predict the tire wear during touchdown a physics-based equation involving slip wear rate was derived and combined with the empirical results to create a nonlinear touchdown response surface. The DT was used to predict the touchdown wear by varying the mean sink rate, yaw angle, and speed.

The F-35 Joint Strike Fighter developed by the US Airforce and Lockheed Martin has a state-of-the-art IVHM system that provides real-time health data during the flight to ground-based Autonomic Logistics Information System (ALIS), to enable the maintainers to be prepared for their activities when the aircraft lands [16] and updates health data to all the local systems DTs at Lockheed Martin. This requires an enormous V&V effort to ensure the integrity of data and also checks for hacking and other security-related issues, resulting in increased operations and maintenance costs. To save MRO cost, Lockheed Martin applied the Digital tapestry concept of data flow tightly woven to integrate the information (Digital thread) from design and development into production DTs, solving the early issues and leaning the production process like automated hole drilling. Lockheed Martin also applied this Digital tapestry concept to validate all the major component designs and enabled the conversion of SysML models to Simulink codes to reduce the gap between the systems and design perspectives [16].

Among the aerospace OEM industries, Rolls Royce uses a DT for engine health monitoring, to enable an IntelligentEngine, which is highly connected and contextually aware. Currently, the DT is used to carry out analysis under various possible instances of blade-off events in Trent engines [27]. Rolls Royce has also combined with Microsoft to enable predictive maintenance in Trent XWB turbofan jet engines, by employing data from historical feeds and real-time monitoring [23]. Similarly, GE has developed DT for

aircraft landing gear by considering an instance as an initial model, followed by frequent updating of data [28]. The landing gear DT is made using asset data and time-series data. This DT is used to detect anomalies, forecast the next maintenance activities and predict remaining useful life of every subsystem of each landing gear, with the help of historical data collected and stored in the Predix platform and by performing predictive analytics [28].

Apart from aerospace, the DT has been used widely in automotive and maritime industries, as well as in railways. Tesla Motors creates a DT for every car it makes and uses it to provide fixes, software updates, and to schedule maintenance based on the car's usage and performance [29]. Likewise, Magargle et al. [30] from Ansys Inc. and Modelon Inc., developed a physics-based DT of automotive braking systems by combining 0-D and 3-D models with controls. They demonstrated the ability of this DT to differentiate the wear rates with normal and abnormal conditions while disconnecting the ABS controller, which would help in suggesting suitable maintenance schedules in advance for the braking system. They also used the DT to obtain sensor signatures for some fault conditions, which were further used to develop diagnostics for the automotive braking system.

The Danish Maritime Authority has adapted the DT for its maritime cluster, Blue Denmark, using 3D, simulation and analytical models, vessel documents and information models, operational data and sensor data, software-driven control algorithms and virtual communication networks [31]. The DTs are used to create value from individual elements like analysis for performance improvement, design verification and optimization, condition-based maintenance, decision support, safety assurance, testing and verification, and virtual operation. Currently, efforts are made to create a DT ecosystem that has the platform to integrate all these key elements to bring benefit to the stakeholders from the information contained in such an ecosystem [31]. GE has developed DT for railways, where, health index is calculated for network based on the defects found on the tracks and wayside [32]. Suitable maintenance plans are updated daily based on the health index. The DT is implemented for train also, for automatic control of throttle and dynamic brakes, optimization of speed and feedback on fuel consumption [32].

3.5 Challenges, Gaps & Opportunities

The applications presented in the previous sections show that the capabilities of DT have enabled it to contribute to high-value industries in a number of sectors. The DT's ability to represent any product or process that could help provide feedback on health or performance or even verification, together with the maturity in the enabling technologies have the following potentials: i) save cost involved in design, testing, inspection, certification and unplanned maintenance, ii) improve the efficiency of knowledge transfer in a product lifecycle, iii) enable transparency in data handling, controlled sharing of data, and iv) to eliminate the silos across the different stakeholders and integrate and streamline the entire supply chain. On the other hand, the implementation of DT has a few challenges to be mitigated. This section discusses some of the challenges met by DT and its prospects in IVHM.

3.5.1 Challenges faced by Digital Twin (DT) in IVHM

One of the significant decision-making factors for implementing DT is the level of fidelity [9]. A dedicated DT, which is a replica of a physical system, consumes a large amount of time for development, requires more data and increases the cost involved. However, a low fidelity DT cannot reap the benefits of investment if it is not complex enough. A trade-off must be made to achieve the required level of fidelity in developing and managing the DT. Secondly, implementing DT requires suitable decisions and investments to be made regarding the enabling technologies. As described in the earlier sections, the technologies influencing DT and IVHM are very similar. The number of sensors used must be optimized to acquire all the required data as well as to not increase the overall weight of the vehicle. Data storage capacity must be factored in to accommodate Big Data, along with the infrastructure required for data management. High speed computations are required to power sophisticated machine learning algorithms; cloud computing facilities including the internet must be accounted for. With the infrastructure established, considering all these key technologies, implementation of the DT would become feasible within the IVHM framework. Another challenge faced by DT is security. In general, due to a large amount of data sharing on the cloud, the DT ecosystem is vulnerable to cyber-attack like hacking and phishing. The credibility of data generated has to be secured, and any possibilities of cyber-attack should be diagnosed

early on and prevented. Industries have started using technologies like Blockchain to secure data and prevent cyber-attacks [33]. One more issue that challenges the implementation of DT within IVHM is data sharing. Since health information about the systems of any asset is confidential to the suppliers, sharing such data without violating their intellectual property rights is not simple [34], which prevents the DT to be a complete representation of its physical counterpart. Further readings on the challenges faced by the DT implementation can be found in these works [[11], [25], [31]].

3.5.2 Gaps & Opportunities

Even though introduced to the field of IVHM as early as 2012, it is only in the last four years that the industry has taken DT into serious consideration, partly due to technology gaps in the industry. However, as the enabling technologies are becoming more affordable, there are opportunities for further research in this area. From the literature point of view, it is to be noticed that, the DTs developed are mostly at the components and subsystems level like fuel pump assembly and some DTs are developed at the system level like the fuel system. Only a very few works [[17], [35]] consider the integration of multiple DT models representing systems like the fuel system, the engine or the electrical power system to emulate their interactions at the level of an aircraft or any other vehicle. The intellectual properties of the suppliers could pose a challenge [34], as the lack of data would prevent the implementation of a fully representative DT. However, a 1-D black box representation that could connect to other such system level DT models for emulating an overall behaviour of the vehicle could be pursued. Also, multi-party computation can be considered to access required IP data from other parties. When DT models representing various components and subsystems are integrated, the seamless simulation of any complex vehicle like an aircraft can be made possible. This will help in optimizing the results produced by the IVHM systems and in turn optimize the maintenance plans for these vehicles through CBM.

3.6 Conclusion

The DT has the potential to be the key to unlock many research areas related to IVHM. As one of the main objectives of IVHM is to integrate all available information to provide solutions to CBM, the DT can play a crucial role in being the platform that integrates all the information to project the health of the complex systems like an aircraft. A component

or system level DT, integrated with other such DTs to emulate the functioning of a complex vehicle, combined with the technology that makes use of the data to assess health of its systems, thus becomes one of the latest enabling technologies of IVHM.

References

1. Grieves M. Origins of the Digital Twin Concept. 2016. Available at: DOI:10.13140/RG.2.2.26367.61609
2. Glaessgen E., Stargel D. The Digital Twin Paradigm for Future NASA and U.S. Air Force Vehicles. 53rd Structures, Structural Dynamics and Materials Conference. 2012. pp. 1–14. Available at: DOI:10.2514/6.2012-1818
3. Gartner Inc. Gartner Survey Reveals Digital Twins Are Entering Mainstream Use. Available at: <https://www.gartner.com/en/newsroom/press-releases/2019-02-20-gartner-survey-reveals-digital-twins-are-entering-mainstream-use> (Accessed: 28 March 2019)
4. Jardine AKS., Lin D., Banjevic D. A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical Systems and Signal Processing*. 2006; 20(7): 1483–1510. Available at: DOI:10.1016/j.ymssp.2005.09.012
5. Redding L. An Introduction to Integrated Vehicle Health Management - A Perspective from Literature. In: Jennions IK (ed.) *Integrated Vehicle Health Management: Perspectives on an Emerging Field*. SAE International; 2011. pp. 17–26. Available at: DOI:10.4271/R-05
6. Ezhilarasu CM., Skaf Z., Jennions IK. The application of reasoning to aerospace Integrated Vehicle Health Management (IVHM): Challenges and opportunities. *Progress in Aerospace Sciences*. Elsevier Ltd; 2019; 105(January): 60–73. Available at: DOI:10.1016/j.paerosci.2019.01.001
7. Jennions IK. Introduction. In: Jennions IK (ed.) *Integrated Vehicle Health Management: Perspectives on an Emerging Field*. SAE International; 2011. pp. 1–8. Available at: DOI:10.4271/R-05
8. Madni A., Madni C., Lucero S. Leveraging Digital Twin Technology in Model-Based Systems Engineering. *Systems*. 2019; 7(1): 7. Available at: DOI:10.3390/systems7010007
9. Tao F., Zhang M., Liu Y., Nee AYC. Digital Twin driven prognostics and health management for complex equipment. *CIRP Annals - Manufacturing Technology*.

- CIRP; 2018; 67(1): 169–172. Available at: DOI:10.1016/j.cirp.2018.04.055
10. Wang J., Ye L., Gao RX., Li C., Zhang L. Digital Twin for rotating machinery fault diagnosis in smart manufacturing. *International Journal of Production Research*. 2018; Available at: DOI:10.1080/00207543.2018.1552032
 11. Singh S., Higgins N. Challenges of Digital Twin in High Value Manufacturing. *SAE Technical Paper*. 2018; 01(1928): 1–10. Available at: DOI:10.4271/2018-01-1928
 12. Liu Z., Meyendorf N., Mrad N. The role of data fusion in predictive maintenance using digital twin. *AIP Conference Proceedings* 1949. American Institute of Physics; 2018. pp. 1–6. Available at: DOI:10.1063/1.5031520
 13. Rios J., Morate FM., Oliva M., Hernandez JC. Framework to support the aircraft digital counterpart concept with an industrial design view. *International Journal of Agile Systems and Management*. 2016; 9(3). Available at: DOI:10.1504/IJASM.2016.079934
 14. Qi Q., Tao FEI. Digital Twin and Big Data Towards Smart Manufacturing and Industry 4.0: 360 Degree Comparison. *IEEE Access*. IEEE; 2018; 6: 3585–3593. Available at: DOI:10.1109/ACCESS.2018.2793265
 15. General Electric. GE Digital Twin. 2016. pp. 1–30. Available at: <https://www.ge.com/digital/sites/default/files/Digital-Twin-for-the-digital-power-plant-.pdf> (Accessed: 4 April 2019)
 16. West TD., Blackburn M. Is Digital Thread / Digital Twin Affordable? A Systemic Assessment of the Cost of DoD’s Latest Manhattan Project. *Procedia Computer Science*. Elsevier B.V.; 2017; 114: 47–56. Available at: DOI:10.1016/j.procs.2017.09.003
 17. Auweraer H Van der. Connecting physics based and data driven models : The best of two worlds. 2018. Available at: https://www.ima.umn.edu/materials/2017-2018/SW3.6-8.18/26842/IMA_2018_Van_der_Auweraer.pdf (Accessed: 3 April 2019)
 18. Zheng Y., Yang S., Cheng H. An application framework of digital twin and its case study. *Journal of Ambient Intelligence and Humanized Computing*. Springer Berlin Heidelberg; 2019; 10(3): 1141–1153. Available at: DOI:10.1007/s12652-018-0911-3

19. Cai Y., Starly B., Cohen P., Lee Y. Sensor data and information fusion to construct digital-twins virtual machine tools for cyber-physical manufacturing. *Procedia Manufacturing*. Elsevier B.V.; 2017; 10: 1031–1042. Available at: DOI:10.1016/j.promfg.2017.07.094
20. Jackson K., Efthymiou K., Borton J. Digital manufacturing and flexible assembly technologies for reconfigurable aerospace production systems. *Procedia CIRP*. The Author(s); 2016; 52: 274–279. Available at: DOI:10.1016/j.procir.2016.07.054
21. Dassault Systems. Boeing & Dassault Systemes Announce Extended Partnership. Available at: https://www.cenit.com/fileadmin/dam/Events/PDFs/Aerospace-day-2018/Partner_der_Luftfahrt_Wie_unterstuetzt_Dassault_Systemes_die_digitale_Transformation_von_Airbus_und_Boeing_CENIT_Aerospace_Day.pdf (Accessed: 5 April 2019)
22. Mars W V., Castanier M., Ostberg D., Bradford W. Digital Twin for Tank Track Elastomers: Predicting Self-Heating and Durability. *Proceedings of the 2017 Ground Vehicle Systems Engineering and Technology Symposium (GVSETS)*. 2017. Available at: <https://events.esd.org/wp-content/uploads/2017/08/Digital-Twin-for-Tank-Track-Elastomers-Predicting-Self-Heating-Durability.pdf>
23. Microsoft Services. The promise of a digital twin strategy. 2017. Available at: [https://info.microsoft.com/rs/157-GQE-382/images/Microsoft%27s Digital Twin %27How-To%27 Whitepaper.pdf](https://info.microsoft.com/rs/157-GQE-382/images/Microsoft%27s%20Digital%20Twin%27How-To%27%20Whitepaper.pdf) (Accessed: 5 April 2019)
24. Jennions IK., Esperon-Miguez M., Niculita O. Integrating IVHM and asset design. *International Journal of Prognostics and Health Management*. 2016; 7(2): 1–16.
25. Tuegel EJ. The Airframe Digital Twin : Some Challenges to Realization. 53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference. AIAA; 2012. pp. 1–8. Available at: DOI:10.2514/6.2012-1812
26. Zakrajsek AJ., Mall S. The Development and use of a Digital Twin Model for Tire Touchdown Health Monitoring. 58th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference. AIAA; 2017. pp. 1–16. Available at: DOI:10.2514/6.2017-0863
27. Pickup O. What is a digital twin and how does it keep Rolls-Royce machines safe? Available at: <https://www.telegraph.co.uk/education/stem-awards/digital/digital->

- twins-computer-modeling/ (Accessed: 2 April 2019)
28. GE Predix Case study. Predictive Insights : Aircraft Landing Gear Prognostics. pp. 1–3. Available at: https://d154rjc49kgakj.cloudfront.net/Case Studies/Case Study_Landing Gear_FINAL.pdf (Accessed: 1 April 2019)
 29. Schleich B., Anwer N., Mathieu L., Wartzack S. Shaping the digital twin for design and production engineering. *CIRP Annals - Manufacturing Technology*. CIRP; 2017; 66(1): 141–144. Available at: DOI:10.1016/j.cirp.2017.04.040
 30. Magargle R., Johnson L., Mandloi P., Davoudabadi P., Kesarkar O., Krishnaswamy S., et al. A Simulation-Based Digital Twin for Model-Driven Health Monitoring and Predictive Maintenance of an Automotive Braking System. *Proceedings of the 12th International Modelica Conference*. Linköping University Electronic Press, Linköpings universitet; 2017. pp. 35–46. Available at: DOI:10.3384/ecp1713235
 31. Danish Maritime Authority. Digital Twins for Blue Denmark. 2018. Available at: <https://www.dma.dk/Documents/Publikationer/Digital Twin report for DMA.PDF> (Accessed: 4 April 2019)
 32. Rao DJ. Digital Twin for the Railway Network. GE Transportation- Digital Solutions. Available at: <https://on-demand.gputechconf.com/gtc/2018/presentation/s8614-digital-twin-for-the-railway-network.pdf> (Accessed: 4 April 2019)
 33. Chavali A., Gottlieb C. Extending the Digital Treas with Blockchain in in Aerospace and Defense. Accenture; 2018. Available at: https://www.accenture.com/_acnmedia/PDF-75/Accenture-Extending-the-Digital-Thread-Blockchain-AD.pdf (Accessed: 31 March 2019)
 34. SAE JA6268: Design & Run-Time Information Exchange for Health-Ready Components. SAE International. April 2018; Available at: DOI:https://doi.org/10.4271/JA6268_201804
 35. Ezhilarasu CM. An Integrated Reasoning Framework for Vehicle Level Diagnosis of Aircraft Subsystem Faults. *Annual Conference of Prognostics and Health Management Society. PHM Society*; 2018. Available at: <https://www.phmpapers.org/index.php/phmconf/article/view/707>

4 Progress towards a Framework for Aerospace Vehicle Reasoning (FAVER)

This paper proposes a reasoning framework to diagnose faults at the vehicle level in a complex machine like an aircraft. The current focus of Integrated Vehicle Health Management (IVHM) is on diagnosing and prognosing faults at the component, subsystem, and system levels; only a few IVHM systems consider the interaction between the systems. To diagnose faults at the vehicle level, an IVHM System needs a framework that recognizes the causal relationships between systems and the likelihood of fault propagation between them. The framework should also possess an element of reasoning to assess data from all systems, to assign priorities, and to resolve ambiguities. The Framework for Aerospace Vehicle Reasoning (FAVER) that is proposed in this paper uses a Digital Twin (DT) of the aircraft systems to emulate functioning of the aircraft and to simulate the effect of fault propagation due to systems interactions. FAVER applies reasoning that can handle fault signatures from multiple systems in the form of symptom vectors, to detect and isolate cascading faults and their root causes. The blending of a DT and reasoning in this framework will enable FAVER to: i) isolate faults that have both local and cascading effects on the concerned systems, ii) identify faults that were previously unknown, and iii) resolve ambiguous faults. This paper explains the different steps involved in developing FAVER and how this framework can be demonstrated in the aforementioned scenarios with the help of different use cases. This paper also talks about the challenges to be faced while developing this framework and ways to overcome them.

4.1	Introduction
4.2	FAVER fundamentals
4.3	FAVER methodology
4.4	Development of FAVER
4.5	A Thought Experiment
4.6	Challenges
4.7	Conclusion & Future work

Figure 4-1 Overview of Chapter 4

4.1 Introduction

Any aerospace vehicle, like an aircraft, is a complex machine comprising various multi-physical systems, each having functions and objectives of their own. These systems interact with each other at different levels to attain full functionality of the aircraft [1]. In general, the aircraft systems are built in such a way that they remain stand-alone to a greater degree, to avoid unnecessary complexities. Still, due to the interactions between systems, it is not uncommon for a fault arising from one system to propagate and affect another system that the former is interacting with. Such cascading faults, whose paths are already known, are isolated in maintenance and troubleshooting activities. This is sometimes performed with aircraft maintenance systems like Honeywell's Prime Epic Aircraft Diagnostic Maintenance Systems (ADMS) [2]. However, when a fault propagation takes a new/ unexpected path and affects multiple systems, it cannot be isolated easily during maintenance. One such real-world incident is the engine rollback of a Boeing 777 at Heathrow airport in 2008 [3]. During the investigation, the reason behind the engine rollback was found to be a drop in its power due to restricted fuel flow to both the engines. Further root cause analysis found that the fuel remained in the 'sticky' temperature range (less than -10°C) for a prolonged period of time; this resulted in ice formation, which in turn was released in the fuel feed pipe and blocked the fuel oil heat exchanger and the rest of the fuel lines. Another such example is the emergency evacuation of a Fokker F28 in 2002, due to smoke in the cabin [4]. The investigation found that it was due to a crack in the Auxiliary Power Unit's compressor blade, the debris

eventually causing a crack in an oil seal and resulting in oil spray in the bleed valve, leading to smoke in the cabin [4].

In cases like the abovementioned incidents, which are met with unexpected failure propagation involving multiple aircraft systems, troubleshooting is not a straight-forward activity, and it results in extended downtime. Any Integrated Vehicle Health Management (IVHM) System that attempts diagnosing such cascading faults requires a holistic view of the aircraft, and the micro effects (confined to a system) and macro effects (affecting the system in addition to the system of origin) of the interactions between the various systems [5]. The current focus of IVHM systems involve automated procedures focused at Line Replacement Units (LRUs), subsystems and systems level, but there are very few research publications that consider ‘vehicle level’ as their primary driver [6]. This paper proposes a framework that aims to isolate the cascading faults affecting multiple systems of the aircraft. The Framework for Aerospace Vehicle Reasoning, also known as FAVER, incorporates the concept of a Digital Twin (DT) to produce simulations of what-if scenarios between the aircraft systems, along with an element of reasoning, to investigate data from the concerned systems and to isolate the root cause of cascaded faults.

4.2 FAVER Fundamentals

FAVER is a proposed framework for isolating faults in aircraft systems, particularly a fault or degradation in one system that affects another. FAVER comprises of two components: i) Digital Twin (DT) and ii) Reasoning. The following subsections present the background of both elements and explain how they will fit together to enable FAVER to isolate cascading faults in different scenarios.

4.2.1 Digital Twin (DT)

A DT is a virtual representation of any physical or functional asset that helps in monitoring the performance of the asset for a variety of outcomes like efficiency, health, or lifecycle cost. The term ‘Digital Twin’ was first coined by Dr. Michael Grieves from the University of Michigan in 2002, to mean a virtual/digital representation of any physical asset [7]. NASA brought the concept of DT into the aerospace field in 2012, to integrate the simulation with on-board IVHM systems, along with historical maintenance and fleet data to represent its exact physical twin for further analysis [8]. Over the last decade, the application of the DT has expanded, due to the abundance of data produced

by the physical systems and the digitization of data collection and processing. In the field of IVHM, the DT has been applied for anomaly detection, predictive and prescriptive analysis and operation optimization [9–11].

By the definition of the DT, it is evident that it is, at its core, a simulation model of any physical asset, and that it functions as a living model and provides results based on its application. For a given instance, in the case of predictive analysis in IVHM, the DT can provide simulation results of the system's remaining life and, in the case of manufacturing, it can give the results required for operation optimization. When extended to the field of health monitoring, the DT is capable of producing what-if scenarios to understand the hidden relationship between the systems; it is this capability of the DT which will be useful to the proposed framework for isolating cascading faults. FAVER uses the DT, as it can establish the interaction between the systems and enable simulations of a healthy state as well as possible fault propagations between the systems.

4.2.2 Reasoning

In the field of aerospace IVHM, several reasoning systems have been developed to monitor the systems health [12–15]. Reasoning is required in IVHM systems for many roles such as analyzing data from multiple sources, isolating the root cause of any faults, helping in decision-making processes, resolving ambiguities, detecting anomalies and upgrading diagnostic accuracies. Reasoning is an invaluable component for developing IVHM systems for vehicle level health monitoring [6].

In order to diagnose faults at the vehicle level, FAVER requires the reasoning element to meet with the following objectives: i) to process data from multiple aircraft systems, ii) to assess information, iii) set priorities, iv) pass judgement on the possible root causes of any fault, and v) to update FAVER's knowledge of any new fault that affects a system.

4.2.3 The synergy between the Digital Twin and Reasoning

The proposed framework, FAVER combines the versatility of a Digital Twin (DT) with the power of reasoning, in such a way that both these components compensate for each other's lacking. A DT could emulate the functioning of an aircraft through simulations of its systems at healthy and faulty (or degraded) states; yet, the output from such simulations require further intense analysis and intelligence to isolate the root cause of the fault from the data produced. On the contrary, reasoning in a health monitoring system

is only as effective as its domain knowledge either in the form of expert systems, models or datasets. Hence, FAVER aims to make use of the synergy between the ability of a DT to emulate the effects of fault propagation between the aircraft systems and reasoning's ability to investigate the data produced to isolate the root cause of the fault propagation. This blending of DT and reasoning will enable FAVER to:

- i) isolate faults that have both local and cascading effects on the concerned aircraft systems,
- ii) identify faults that were previously unknown, and
- iii) resolve ambiguous faults.

4.3 FAVER Methodology

Figure 4-2 shows the schematic of the proposed framework. The schematic is represented with two main layers, viz, i) The Digital Twin layer (bottom of Figure 4-2), and ii) the Reasoning layer (top of Figure 4-2). The two intermediate layers in Figure 4-2 represent the system level diagnostics and fault detection layers. Each vertical in Figure 4-2 represents the modules required from each system, contributing to reasoning at the top level.

Consider the Fuel System, the Engine, the Environmental Control System (ECS) and the Electrical Power System (EPS) of an aircraft (as shown in the bottom-most layer of Figure 4-2). These systems have functions of their own and part of their functions that involve interacting with the other systems. For illustration, the fuel system provides fuel to the engine, the engine provides bleed air to the ECS and shaft power required for the EPS, and the EPS provides electric power to the components in the fuel system, the engine and the ECS. These systems interactions are considered for emulating the interactions in an aircraft via DT within FAVER. Besides, the EPS is also chosen to add a multi-physical dimension to the problem chosen. The DT can be a physics-based or a function-based model, or it can be a data-driven representation as well. The fuel system is included in the DT layer as a Hardware-in-the-loop (HIL) representation.

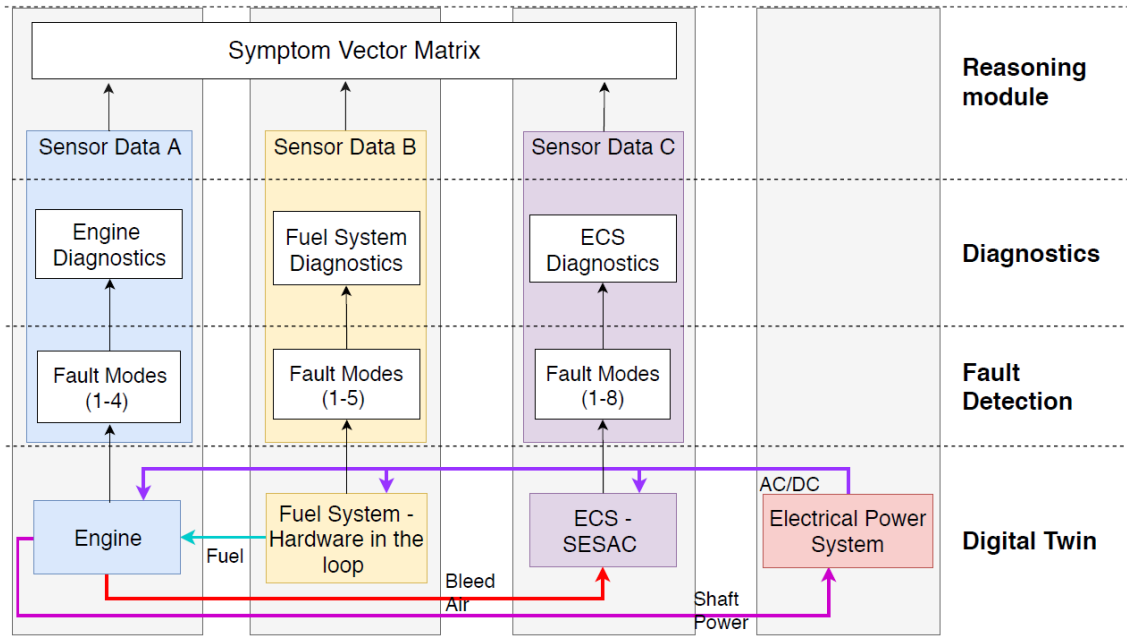


Figure 4-2 The schematic of FAVER

The second horizontal layer from the bottom in [Figure 4-2](#) represents the number of fault modes being considered at the system level. These fault modes degrade the systems from its 100% healthy state to degraded or faulty states. Only a limited number of fault modes are taken into account for the initial setup of the framework, as FAVER aims to demonstrate its capability of isolating a certain number of fault modes with a broader range of systems rather than focusing on a large number of fault modes only from a few systems. The third horizontal layer from the bottom in [Figure 4-2](#) shows that each system has its own diagnostics for the determined fault modes. These systems diagnostics are fed by sensor data from the DT for isolation of local fault with micro effects. They are incorporated in FAVER's schematic so as to contribute to overall reasoning at the vehicle level.

The reasoning is built on the diagnostic capability of the systems and their interactions. The reasoning module has access to both sensor data and diagnostic information from the systems. The system level knowledge is stored in the form of a symptom vector (sensor readings of selected parameters representing the health of each system), and any fault injected that affects one or more systems is isolated as a 'known fault', with the help of built-in knowledge about the interacting functions of the systems.

In Figure 4-2, it is to be noted that the right-most vertical for the electrical power system does not possess any diagnostic module. This is to test FAVER for its ability to isolate faults that are previously unknown, as it is impractical to assume that reasoning in any health monitoring system is aware of all possible faults that might affect the systems. Consider a scenario where a fault signature is introduced in the EPS and it affects another system like the fuel system. Since the top-level reasoning module depends upon the sensor data and system diagnostic information for isolating faults, this fault injected in the EPS will not be detected as a ‘known fault’ right away by reasoning. In general, the maintenance engineers will have to troubleshoot and isolate such a fault manually, since it will go unnoticed or misclassified by any diagnostic system. However, in this case, instead of misidentifying the electrical fault as ‘no fault’ or misclassifying it as a fault in the fuel system, FAVER would use the EPS DT with suitable virtual sensors to identify the combination of parameters that would lead to the injected fault signature. This fault signature can then be manually added to reasoning’s knowledge module with the electrical fault as its new label.¹

The EPS DT is also used by FAVER to flag ambiguity. This is especially useful for the interacting systems, as many mechanical faults have similar symptoms as the electrical faults. As the EPS supplies power to many components in multiple systems, it is possible that fault symptoms occurring due to degradation in the EPS are ambiguous with mechanical degradations. In such cases, there is a risk of misclassifying a fault mode as a mechanical fault when it was originally an electrical fault. Using the EPS DT, FAVER will be able to reverse engineer the fault symptoms that match with the mechanical fault modes, to check if they can also be produced by the electrical parameters. When a suitable combination of parameters which could lead to the symptom (that matches with the mechanical fault), ambiguity will be flagged for that fault, and the reasoning’s knowledge module will be updated.

¹ The assumption that the EPS will not be instrumented for diagnostics for FAVER is removed in the later stages of this research. The updated schematic of FAVER can be seen in Figure 5-3. While it was intended to explore the previously unknown faults by not instrumenting the EPS diagnostics, the scope of the research has changed and it was decided not to pursue with this assumption.

4.4 Development of FAVER

FAVER is being developed through four stages, as shown in Figure 4-3: i) Use case Conceptualization, ii) Development of the building blocks, iii) Implementation and Testing, and iv) Evaluation. This section talks about the different steps involved in each stage of development.

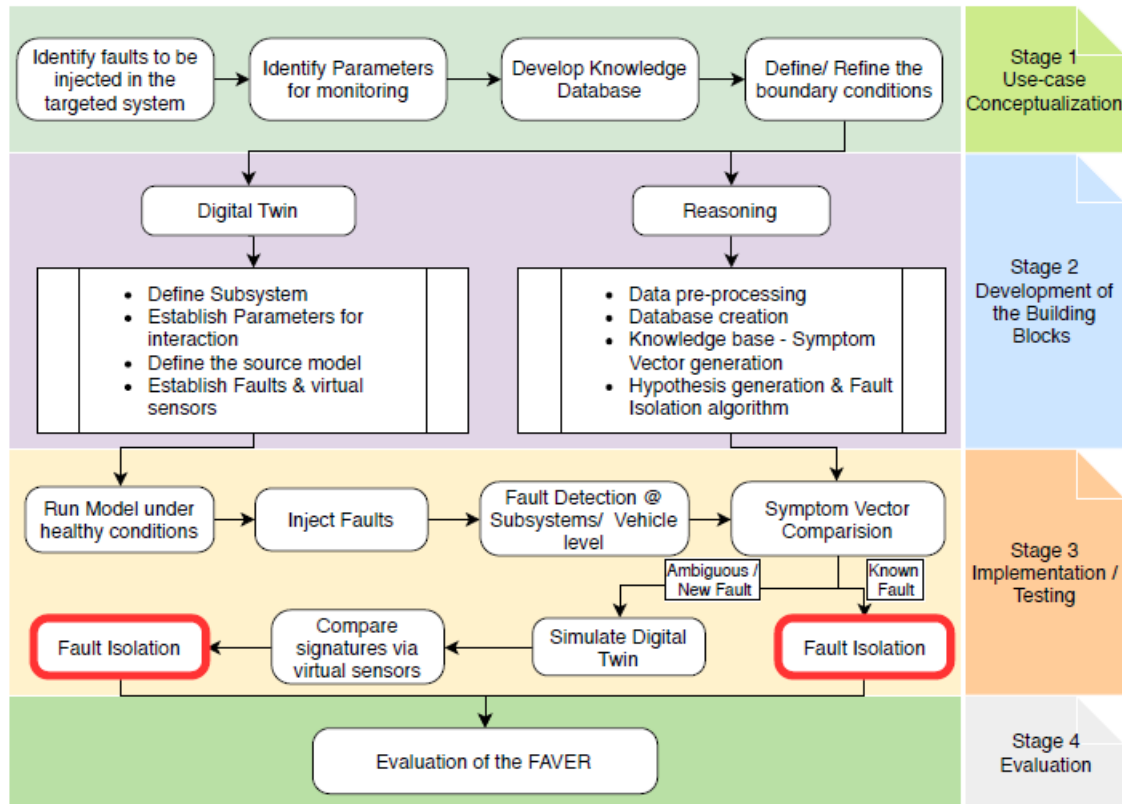


Figure 4-3 Stages in developing FAVER

Stage 1: Use case Conceptualization

This initial stage involves conceptualization of use cases that will be demonstrated through FAVER. The use cases are defined in such a way that the overall objectives of FAVER are satisfied through the demonstration. The cases of isolating single (system level) and cascading faults (vehicle level) that are previously known and unknown are framed into a number of distinct scenarios (as shown in Table 4-1) for which the use cases are defined. These scenarios will enable demonstration of FAVER to isolate faults that affect a system locally, at a micro scale as well as those that have a macro effect on other systems. It is to be noted that, for a fault mode that is previously unknown, its interaction effect cannot possibly be known and thus, the sixth case in Table 4-1 is also not possible.

The use cases will be formed for the remaining scenarios in Table 4-1 and ambiguity can be checked for the known faults. The use cases (faults to be injected) are chosen from the scenarios from Table 4-1, with reference to the four systems considered in the FAVER schematic (Figure 4-2) that emulate the interaction between the systems of an aircraft. The faults chosen for demonstration can either be a single fault affecting a local system or a cascading fault that has a macro effect on another system.

Table 4-1 Use case Scenarios

System	Fault Mode	Interaction Effect	Use Case Scenario	Effect-scale
Known	Known	-	Known Single Fault	Micro
Known	Known	Known	Known Interaction Fault	Macro
Known	Known	New	Known fault with new interaction effect	Micro
New (Previously Unknown)	New		New single fault	Micro
New (Previously Unknown)	New	New	New interaction fault	Macro

For example, consider Figure 4-4. A reduced AC supply to a motor pump in the fuel system has its origin in EPS. However, this fault in the EPS has a cascading effect, which is to reduce the power to the motor pump, which in turn affects the fuel delivery to the engine and results in reduced power from the engine (redundancy is not considered in the use cases for demonstration purposes). This ‘domino effect’ has its root cause, the reduced AC supply from the EPS, and this fault can be flagged as one with a macro effect. This example can be used to check multiple use case scenarios like i) known fault with known interaction effect, ii) known fault with new interaction effect, and iii) new fault with a new interaction effect. Similar examples are being formulated to test the capability of fault isolation by FAVER.

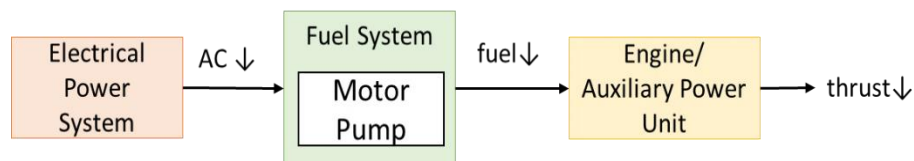


Figure 4-4 An example of a system fault with macro effect

For the use case scenarios of a known single fault and a new single fault, faults injected at the system level can be chosen and tested. Ambiguity can be introduced in the system by injecting faults in components like valves. For example, a sticking valve in the fuel system can be a mechanical fault, and also a result of a stuck relay that reduces the voltage supplied to power the valve. The symptom vector for a sticking valve will be the same, despite the cause. Introducing such ambiguity could showcase the ability of FAVER to resolve such faults. When the use cases are defined, health parameters to be monitored will be decided, and the domain knowledge required for that use case are collected.

Stage 2: Development of the building blocks

This stage involves the development of the Digital Twin (DT) of the systems and the corresponding diagnostic reasoning at both system and vehicle level. For example, when the fuel system DT is built with respect to its source model, faults to be injected are defined, sensors to isolate faults are established and the parameters required for interaction with the EPS and the engine to demonstrate one of the use case scenarios are defined. In parallel, the fuel system diagnostics with isolation algorithms is built at the systems level and the knowledge base in the form of symptom vector is populated for reasoning at the vehicle level.

Stage 3: Implementation and Testing

The third stage of developing FAVER involves implementation and testing. In this stage, the DT of a system is run in a healthy condition, followed by fault injection and data collection. The symptom vector generated is compared with the knowledge base of reasoning to identify whether the fault is previously known. If no matches are found, the DT is used to check if any other system is the root cause of the injected fault, resulting in a previously unknown or ambiguous fault². Stage 2 and Stage 3 are repeated for building the DT and diagnostics for all four systems shown in [Figure 4-2](#). The reasoning module

² While it was the intention to use the DT to check for unknown faults when this paper was written, the scope of the research had changed when the work expanded. As of now, the DTs are not used to check for unknown fault signatures in this research, and the term ‘previously unknown’ fault or system refers to a new fault or new system added to the framework in chapters 5 and 8.

at the vehicle level is expanded to accommodate the different domain knowledge and suitable isolation algorithms.

Stage 4: Evaluation

Once all the systems are accounted for, in terms of the DT, their interactions, system level diagnostics, and top-level reasoning, FAVER will be evaluated for the performance metrics like classification accuracy, misclassification rate, number of false positives and false negatives and the time taken for classification. It will also be tested for its scalability and its sustainability to add or remove an aircraft system module to the framework.

4.5 A Thought Experiment

A thought experiment was conducted with an experimental fuel rig to understand the interdependencies between various systems. The idea is to develop a conceptual model of how the reasoning is going to be demonstrated, while keeping the logic tractable. Too complex and the logic won't be traceable or obvious, too simple and it won't be effective as a framework. With the help of this thought experiment, a number of assumptions are seen to be necessary for the initial demonstration of FAVER.

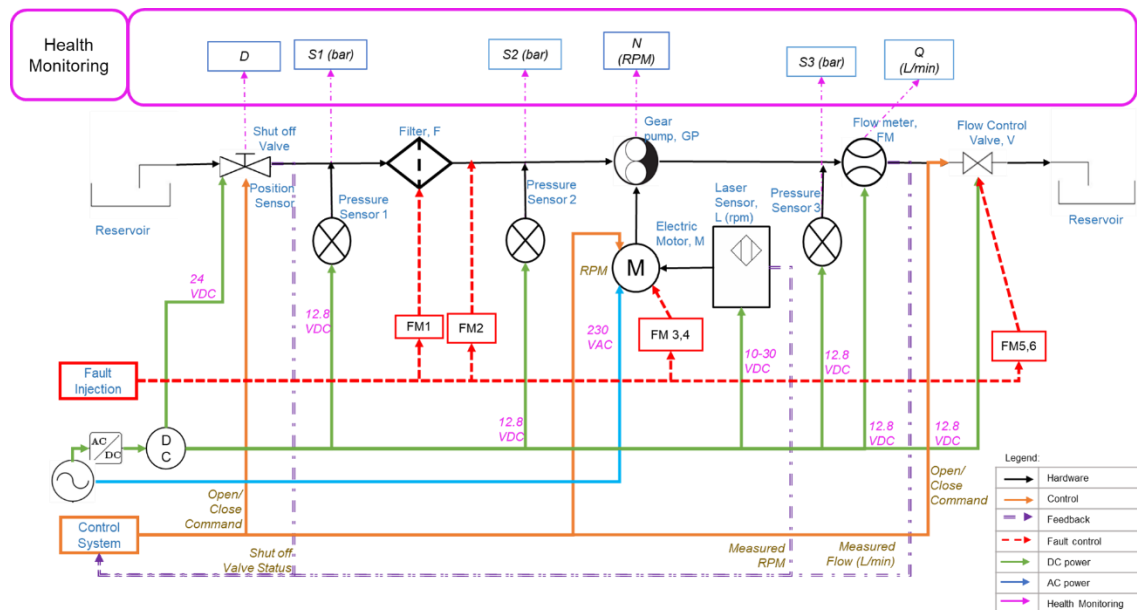


Figure 4-5 Multiple systems in a fuel rig - A thought experiment

The fuel rig on which the thought experiment was developed in the IVHM lab in Cranfield University [16] and is representative of a commercial aircraft fuel system.

Consider [Figure 4-5](#) where a small mechanical layout from the rig is presented for this thought experiment. [Figure 4-5](#) can be seen to involve seven different layers. The mechanical system (hardware layer) consists of a reservoir from which the fuel is pumped by a gear pump GP, powered by a motor M, and it passes through a shutoff valve and filter F, followed by a flow control valve V, and back to the reservoir. There are three pressure sensors (S1, S2, & S3), a laser sensor L, and a flow meter FM that provide sensor readings from the mechanical layer to the health monitoring layer (topmost layer of [Figure 4-5](#)). The electrical system is shown via DC and AC layers; the sensors, the flow meter, the flow control, and the shutoff valves are all powered by 12-24 VDC; the motor M that drives the gear pump is supplied by 230VAC power. The electrical layer is required for the functioning of the mechanical components. Similarly, the control system is shown in two layers, viz, control and the feedback layers: they consist of the control functions pump speed N, the valve positions and the flow Q through the flowmeter, and their feedbacks respectively. The fault modes are injected into the fuel rig via the control system. For the effective functioning and diagnosing of the fuel rig, it is essential that all the dependent systems, i.e., the mechanical system, the electrical power system, and the control system should work together, along with the sensors and flowmeters which connect them to the health monitoring for diagnosis.

Table 4-2 List of fault modes for the thought experiment

Mode	Fault	Type
FM1	Filter Clogging	Mechanical
FM2	Pipe Leaking	Mechanical
FM3	Gear tooth broken in the pump	Mechanical
FM4	Low power to the electric pump	Electrical
FM5	Clogged flow control valve	Mechanical
FM6	Low power to the flow control valve	Electrical

In the beginning, faults are considered only within the mechanical and electrical systems. Six fault modes that are planned to be injected are listed in [Table 4-2](#). Among these fault modes, the clogged filter (FM1) and the pipe leak (FM2) are mechanical faults, whereas the reduced pump speed can either be due to reduced electric power to the pump

(FM4) or due to a broken gear tooth in the pump (FM3). Similarly, a clogged nozzle may occur due to either mechanical blockage (FM5) or lack of electrical power (FM6).

Since the electromechanical components used in the test rig are susceptible to both mechanical and electrical faults, the symptom vectors for such fault modes resembled each other, thus introducing ambiguity to the problem. The symptom vector matrix was developed to represent the fault signatures at the system level, using which the diagnostic rules were developed to isolate both mechanical and electrical faults. In the case of the control system, with respect to the fuel rig, its role does not stop only at controlling the valves and motor speed and receiving the feedback. Rather, the control system is essential for injecting faults into the rig for diagnosis. Thus, introducing faults in the control system will render the diagnosis of fuel rig invalid. In practice the control system may well mask potential faults. Hence, for demonstrating FAVER with the fuel rig, only mechanical and electrical systems are assumed to be at fault, the control system is considered to be healthy. The same is the case for considering faulty sensors. Since the health monitoring layer, the output of which is used for diagnosing the faults in the fuel rig, is dependent upon the sensor readings, for the purpose of initial testing, the sensor data are assumed to be reliable. In a similar fashion, a few more assumptions are made to define the scope of the research, within which FAVER will be demonstrated. A complete list of the assumptions made is as follows:

1. The EPS is not instrumented for diagnostics
2. The control system is assumed to be 100% healthy
3. Only single faults are considered
4. Sensors are always healthy
5. Fault modes signatures are to be studied at steady state
6. No False Alarms are considered

As previously mentioned, these assumptions are planned to be conservatively updated or removed at later stages and in future work, to expand the framework and improve FAVER's ability to isolate faults under complex conditions. The assumption about the EPS is removed in the next chapter and diagnostics is built for the EPS in chapter 6.

4.6 Challenges

The development of FAVER poses a number of challenges. In order to enable FAVER to isolate cascading faults that are previously known and unknown and to flag ambiguity, a certain number of assumptions are made, as mentioned in the previous section. These assumptions will be treated as special conditions and the future research will remove/update them to expand the ability of FAVER to isolate faults in these special conditions.

FAVER requires validated simulations with which to build the Digital Twin (DT) for emulating the systems interactions. Developing every system from scratch is time-consuming and will not fit into the timescale for completing FAVER. Hence the DT sources are taken from previously developed and validated simulation models from research work within the IVHM Centre. Similarly, existing diagnostic methods are chosen for system level diagnosis, and reasoning is developed only at the vehicle level to isolate the faults that have a macro effect on multiple systems.

One more challenge for developing FAVER is the complexity involved in establishing interaction between the systems. As FAVER is designed to accommodate multiple systems, the framework must possess the required features to enable systems interactions at the vehicle level. If a new system is introduced, it must be checked for all possible connections with the other systems and every system interacting with the new one must be updated for enabling the interactions via DT. All the concerned systems must go through rigorous verification and validation to account for the change. For this purpose, modularity is being introduced to the framework's architecture. Every system will be encapsulated and will be treated as an independent module. The interaction between the modules will be established only through the modules meant for communication. In this case, when a new system is added, the necessary parameters will only be updated in the communication modules and the sources of other systems DT will not be disturbed. It is to be noted that, along with the challenge of complexity and sustainability, the modularity feature takes care of the issues of scalability for the framework. Even if the systems are of different scales, the scaling factor can be introduced in the communication modules, through which the DT interactions can be enabled.

4.7 Conclusion & Future Work

A Framework for Aerospace Vehicle Reasoning (FAVER), conceptualized with its main components of Digital Twin (DT) and reasoning, is discussed in this paper, along with the different stages of development and the use case scenarios. Developing and demonstrating FAVER will provide a pathway to isolate cascading faults and resolve ambiguities under multiple scenarios in the aircraft systems.

Currently, the architecture of FAVER is being designed systematically, to include all of the essential components. As most sources of aircraft systems are available for the DT from the IVHM Centre, they will be brought into the framework via the modularity feature and tested for interaction with the other systems. Once the interactions are established, the reasoning strategies will be built to isolate cascading faults that are previously known or unknown to the system, as well as to identify ambiguity, i.e., [Figure 4-2](#) will be executed.

REFERENCES

1. Ezhilarasu CM. An Integrated Reasoning Framework for Vehicle Level Diagnosis of Aircraft Subsystem Faults. Annual Conference of Prognostics and Health Management Society. PHM Society; 2018. Available at: <https://www.phmpapers.org/index.php/phmconf/article/view/707>
2. Scandura PA., Christensen M., Lutz D., Bird G. Commercial Aviation Health Management. Systems Health Management: with Aerospace Applications. John Wiley & Sons; 2011. pp. 589–603.
3. Sleight PA., Carter RDG. AIRCRAFT ACCIDENT REPORT 1/2010. 2014. Available at: DOI:<https://doi.org/EW/C2008/01/01>
4. Conradi K. AAIB Annual Safety Report 2015. 2015. Available at: https://assets.publishing.service.gov.uk/media/5656b9e2ed915d0367000008/Annual_Safety_Report_2015.pdf
5. A.N.Srivatsava., R.W.Mah., R.Bharadwaj., D.Mylaraswamy. Vehicle-Level Reasoning Systems. In: Jennions IK (ed.) Integrated Vehicle Health Management: The Technology. SAE Publications; 2013. pp. 139–155. Available at: <http://books.sae.org/r-429/>
6. Ezhilarasu CM., Skaf Z., Jennions IK. The application of reasoning to aerospace

- Integrated Vehicle Health Management (IVHM): Challenges and opportunities. Progress in Aerospace Sciences. Elsevier Ltd; 2019; 105(January): 60–73. Available at: DOI:10.1016/j.paerosci.2019.01.001
7. Grieves M. Origins of the Digital Twin Concept. 2016. Available at: DOI:10.13140/RG.2.2.26367.61609
 8. Glaessgen E., Stargel D. The Digital Twin Paradigm for Future NASA and U.S. Air Force Vehicles. 53rd Structures, Structural Dynamics and Materials Conference. 2012. pp. 1–14. Available at: DOI:10.2514/6.2012-1818
 9. General Electric. GE Digital Twin. 2016. pp. 1–30. Available at: <https://www.ge.com/digital/sites/default/files/Digital-Twin-for-the-digital-power-plant-.pdf> (Accessed: 4 April 2019)
 10. Auweraer H Van der. Connecting physics based and data driven models : The best of two worlds. 2018. Available at: https://www.ima.umn.edu/materials/2017-2018/SW3.6-8.18/26842/IMA_2018_Van_der_Auweraer.pdf (Accessed: 3 April 2019)
 11. The Aerospace Technology Institute. Insight_01- Digital Transformation. 2017. pp. 1–10. Available at: <https://www.ati.org.uk/resource/insight-01-digital-transformation/> (Accessed: 28 March 2019)
 12. Bunus P., Isaksson O., Frey B., Munker B. Model-based diagnostics techniques for avionics applications with RODON. 2nd Workshop on Aviation System Technology. 2009. Available at: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.159.3003&rep=rep1&type=pdf>
 13. Sebastian RK., Peripinayagam S., Jennions IK., Alghassi A. Prognostic reasoner based adaptive power management system for a more electric aircraft. Proceedings of the Annual Conference of the Prognostics and Health Management Society. 2016. pp. 1–11.
 14. Park HG., Cannon H., Bajwa A., Mackey R., James M., Maul W., et al. Hybrid Diagnostic System: Beacon-Based Exception Analysis for Multimissions-Livingstone Integration. 58th Meeting of the Society for Machinery Failure Prevention Technology. Pasadena, CA: JPL, NASA; 2004. Available at: <http://hdl.handle.net/2014/38232>

15. Gaudette T., Alwardt AL. The Boeing smart TPS application executive with MATLAB as a diagnostic reasoner. AUTOTESTCON (Proceedings). IEEE; 2006. pp. 714–719. Available at: DOI:10.1109/AUTEST.2006.283754
16. Yufei Lin. System Diagnosis using a Bayesian Method. PhD Thesis, Cranfield University; 2017.

5 Developing the architecture of FAVER (a Framework for Aerospace Vehicle Reasoning) using a System-of-Systems methodology

This paper discusses the development and implementation of the architecture of a framework for aerospace reasoning, ‘FAVER’. Current Integrated Vehicle Health Management (IVHM) systems focus mainly on diagnosing faults at line replaceable unit level and at the system’s level. FAVER is a framework that combines the power of reasoning with the versatility of digital twins to deal with faults that involve multiple systems at the aircraft level. In this paper, FAVER is treated as a platform and is developed with a System-of-Systems methodology through a tailored V- process model; the paper presents the process of capturing requirements according to a Holistic Requirements model and developing test plans to verify the requirements of the platform, FAVER. The paper also proposes a modular architecture for FAVER to deal with issues such as scalability, complexity, verification and validation. In addition to this, the paper presents a novel approach to handle the increased complexity of a traditional relationship matrix within an expert system. The capability of the proposed architecture is demonstrated using two aircraft systems: the electrical power system and the fuel system. Multiple use case scenarios involving these two systems are demonstrated for fault isolation using FAVER. Future direction for expanding this framework is also discussed at the end of the paper.

5.1	Introduction
5.2	FAVER architecture with a System-of-Systems methodology
5.3	FAVER architecture & Working mechanism for Known faults with use cases
5.4	Discussion & Future work
5.5	Summary

Figure 5-1 Overview of Chapter 5

5.1 Introduction

Integrated Vehicle Health Management (IVHM) is an evolving capability that enables the condition-based maintenance of complex vehicles like aircraft. It uses data from various sources such as sensors, maintenance records, and design documents to assess and predict the health of the vehicle [1]. The main objective of IVHM is to enable the vehicle to function as intended, thus saving time and cost associated with unexpected downtime and the eventual consequence of prolonged maintenance activities. IVHM aims to be a part of the asset's end-to-end process, starting from design requirements leading up to after-sales service, rather than being an add-on service to an already matured process [1].

Reasoning is an integral part of an IVHM system and can be found functioning in several capacities to help achieve the objectives of IVHM. Figure 5-2 shows the functioning of reasoners within an IVHM system.

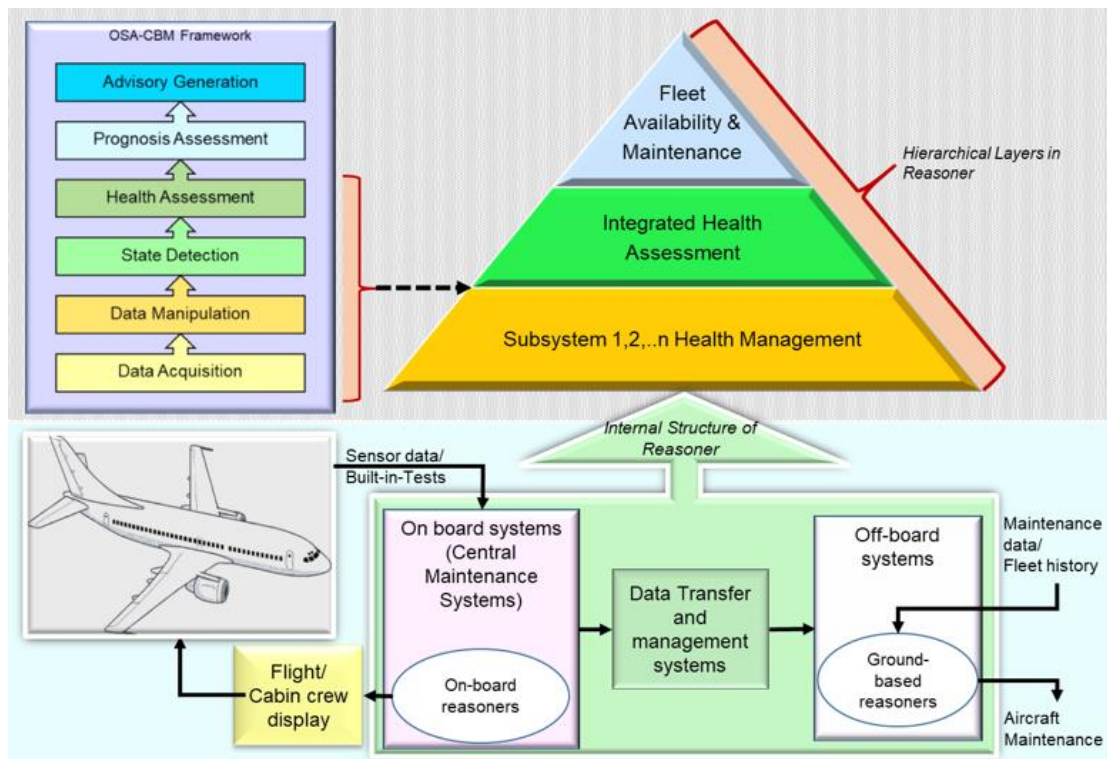


Figure 5-2 OSA-CBM influenced hierarchical structure of Reasoners in an IVHM system

The bottom segment of Figure 5-2 shows the different sections of an IVHM system viz: i) On-board systems, ii) Off-board systems, and iii) Data transfer and management systems and its interface with an aircraft. The top segment shows the hierarchical

architecture of reasoning present in the IVHM system. The on-board systems, like the Central Maintenance Computers (CMCs) in the bottom segment of [Figure 5-2](#), could be for both critical and non-critical aircraft systems. They collect and process real-time health information, using systems such as on-board reasoners and presenting them through in-flight display. The complicated information that could not be processed using the on-board reasoners is transferred to the off-board systems using the data transfer and management systems. The off-board systems use ground-based reasoners along with fleet maintenance histories and other supporting data to assess the health of the concerned systems [2–4]. The interface requirements for information exchange between the different segments within the IVHM systems are usually governed by standards like OSA-CBM (Open Standard Architecture for Condition Based Maintenance) provided by MIMOSA foundation [3].

The on-board and the ground-based reasoners in a health management system are commonly made of hierarchical architectures. As shown in the top segment of [Figure 5-2](#), each bottom layer supports the next top layer; and the high-level reasoners are built on top of the aircraft subsystem/ system's health management functions [5–8]. A few examples of hierarchical architecture for reasoning for health management are presented here. Hernandez et al. (2010) [7] developed the framework for aircraft Electrical Power Generation System (EPGS) health management system using a three-layered architecture, where the lowermost layer comprised of health management functions from the EPGS components like auxiliary power unit generators and their control units, integrated drive generators, generator control units, and bus power control units. The middle layer of the hierarchical architecture was built to integrate the health information from the lower layers and this integrated view is then used at the top layer for objectives like fleet management and mission planning operations. On the other hand, Roemer (2011) [8] demonstrated a hierarchical reasoning structure for IVHM by degrading the Line Replaceable Units (LRUs), a fuel filtering system and a drive train, to see their effects on their respective subsystems, the fuel system, and the actuator subsystem.

The health management systems are now developed using guidelines provided by standards such as IEEE AI-ESTATE for handling matured Artificial Intelligence (AI) techniques for diagnostic systems and IEEE SIMICA for supporting the use of historical

information for diagnostics and prognostics [9]. There are also several generic layered architectures used by health management systems that used standards like ISO 13374 for guidelines. In fact, OSA-CBM defines the interface methods for the functional blocks defined in ISO 13374 [3]. The most common architecture of any IVHM system can be presented by the generalized OSA-CBM framework. The OSA-CBM framework defines a layered approach with a series of steps (as seen in the left side of the top segment in [Figure 5-2](#)) starting from 1. Data acquisition, 2. Data manipulation, 3. State detection, 4. Health assessment, followed by 5. Prognosis, and 6. Advisory generation. Data is acquired through sources like sensors, fleet data, and maintenance history records from the concerned host system, which could range from an LRU like a gear assembly to an entire system like the engine. The collected data is then manipulated using several feature extraction and selection techniques. The processed and manipulated data is then used for detecting the current state of the host system and comparing with the nominal state to identify if the host system has any degradation in its health, followed by remaining life prediction. The results from steps 3-5 are then used to generate advisories regarding the health of the concerned system, which are in turn utilized for tailoring the maintenance program of the same [10]. Several health management programs have used OSA-CBM architecture for implementing IVHM systems in complex vehicles such as aircraft, spacecraft and naval ships [10–12].

While the reasoners use hierarchical and layered architectures and follow OSA-CBM framework in the health management systems to process health data, they employ various reasoning techniques to carry out the steps of diagnosis, prognosis and to generate advisories using data sources ranging from sensor data to maintenance records. For example, a combination of fuzzy and mathematical reasoning was used to assess the health of a pump in a fuel system and the applicability of causal reasoning for the same problem was also presented [13]. Similarly, a combination of model based reasoning (TRANSCEND) was applied with data-driven reasoning for the diagnosis of faults in electro-mechanical actuators [14]. COTS packages like TEAMS [15,16] and ReasonPro [17] use multiple reasoning techniques to diagnose faults in concerned systems. NASA has built several reasoning systems such as Livingstone, BEAM, SHINE for their health management programs. Among the aircrafts, CMCs in B747 and B777, Honeywell's Prime Epic Aircraft Diagnostic Maintenance Systems (ADMS), Airbus' AirTHM are

some of the state-of-the-art health management programs that use multiple reasoning techniques. As a foundation for this research, a detailed review on reasoning techniques applied in IVHM systems was previously published [9], which presented the different reasoning systems developed in the industry and academia, discussed their architectures, various reasoning techniques and the current gaps found in the field of IVHM that could be filled by reasoning techniques.

5.1.1 Gaps in the current IVHM systems

An aircraft, being a complex machine, has multiple systems interacting with each other, resulting in faults originating in one system and propagating to another due to the interactions between them. One of the major gaps found in IVHM is that the IVHM systems developed for an aircraft so far are focused mainly at the Line Replacement Units (LRUs) level (e.g. gearbox assembly), or at the system level (e.g. engine) [9]. Even for the cases of fault propagation, most researches focus majorly at the component/ LRU/ systems level such as actuator and sensor faults in HVAC systems [18], power distribution systems [19], mobile robot systems [20], and simple paper path systems [21]. However, there is a lack of research on isolating cascading faults and identifying their root causes at the aircraft level. This is especially important to avoid accidents and unexpected downtime which will happen due to fault propagation incidents involving multiple aircraft systems. For example, one of the aircraft accident investigations showed that the reason behind the engine rollback of a Boeing 777 at Heathrow airport in 2008 was due to the fuel restriction to the engine. It was later found during the root cause analysis that the fuel restriction was caused by ice which had formed at the fuel-oil heat exchanger, as the fuel has stayed in the sticking range (-10 deg C) for a prolonged duration [22]. Similar incidents were caused due to fault propagation involving multiple aircraft systems [23,24]. Troubleshooting activities in such cases are not straight forward and take several months to pinpoint the root cause of the problem.

In order to diagnose cascading faults in scenarios similar to the aforementioned incidents, the IVHM systems require a ‘vehicle level’ approach. Currently, the research area of cascading faults from a vehicle perspective is quite underexplored and there are only a few works which are dedicated to developing the IVHM systems at the vehicle level [8,25–27]. Moreover, many researches carried out at vehicle level have a centralized

architecture [8,27], which pose a problem of scalability at the later stage. To address this gap of diagnosing cascading faults at the vehicle level, a modular framework titled, a Framework for Aerospace Vehicle Reasoning (FAVER) that combines the concepts of Digital Twin (DT) and reasoning has been proposed [28].

5.1.2 FAVER – The Proposed framework

The framework of FAVER has been proposed to isolate cascading faults or degradations that originate in one system and affect another system in an aircraft [28]. FAVER consists of two important components, viz: i) Reasoning and ii) a Digital Twin (DT). The reasoning component is employed by FAVER to reason through the health information from various aircraft systems in order to set priorities and pass judgments as to which system is the root cause of the concerned fault and which are the affected systems. The Digital Twin (DT) is a virtual representation of any physical asset [1] and it is used by FAVER to emulate the interaction between the aircraft systems, to produce ‘what-if’ simulations to explore the unexpected scenarios in advance, and to validate the results produced by the reasoning component. FAVER is one of the first frameworks to combine the versatility of DT with the power of reasoning to isolate cascading faults at the vehicle level. Using this unique combination, FAVER aims to:

- 1) isolate faults that affect both local systems as well as have macro effects on the interacting systems,
- 2) find ambiguity among the faults and resolve, and
- 3) expand the framework to accommodate systems that are previously unknown [28].

At the initial stage, the conceptual framework was set up with multiple layers, as shown in [Figure 5-3](#). The bottom layer consists of DTs of four aircraft systems, viz: the engine, the Electrical Power System (EPS), the Environmental Control System (ECS), and the Fuel System (FS). These systems were chosen for developing use case scenarios as they interact with each other extensively. Some of the main interactions between these systems (as shown in the bottom layer of [Figure 5-3](#)) are i) the FS provides fuel to the engine, ii) the engine provides shaft power to the EPS and the bleed air to the ECS, and iii) the EPS provides power to the valves, pumps and other electrical components in the FS, the engine and the ECS. The middle layers of [Figure 5-3](#) show that each of these

aircraft systems (verticals) has their system diagnostics which are capable of isolating a certain number of fault modes with local effects. The top layer shows that the high level reasoning is built on top of the system level diagnostic functions, along with additional access to the sensor data from each vertical, which will be used to isolate faults with macro effects on the interacting systems.

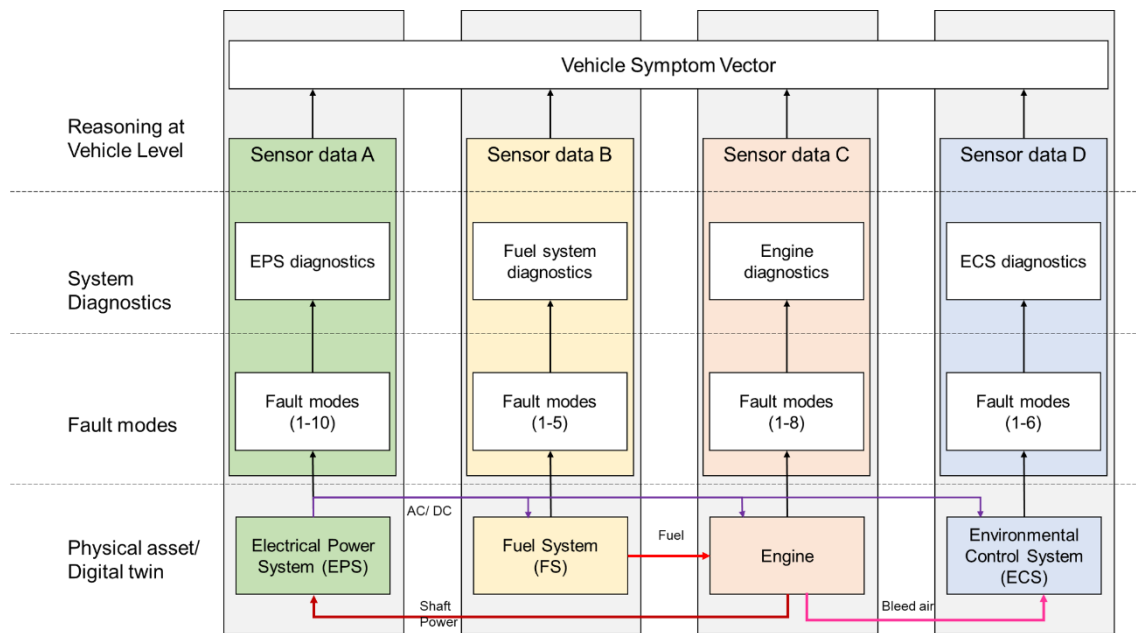


Figure 5-3 The working concept of FAVER (adapted from [28])

Several combinations of use cases were developed from the interactions between these systems, which were to be tested at the later stage. A thought experiment was conducted to examine various possible approaches towards the framework and a basic set of assumptions were derived. These assumptions are i) The control system is considered healthy, ii) Sensors are healthy, iii) Single faults are only considered, iv) Only steady-state scenarios are considered, and v) No false alarms [28]. These assumptions will be removed or updated as the works progressed.

In fact, the important assumption that was removed from the thought experiment is that the EPS will not be instrumented for diagnostics [28]. This is because, during conceptualization of FAVER, it was assumed the EPS diagnostics will be built on demand from FAVER, when a new sensor data is introduced to the high level reasoner. However, this assumption was deemed redundant, as any aircraft system with sensor data will have some sort of diagnostics already in place. In such case, considering the lack of diagnostics

for the EPS initially, only to reintroduce it to FAVER at the later stage seemed futile and roundabout. Hence FAVER was built with access to the EPS diagnostics in this paper, as the way forward. The other assumptions will be removed or conservatively updated as a part of future work on FAVER.

As the next step to the proposed framework, this paper deals with the development and implementation of FAVER's architecture using a System-of-Systems (SoS) methodology. In order to discuss the architecture of FAVER and its role in diagnosing cascading faults, Section 5.1 introduced the role of reasoners in IVHM systems, the current gaps in developing the IVHM systems, followed by the general idea behind the conceptual framework of FAVER, and how FAVER would help address the gap in IVHM systems. Section 5.2 presents the architecture of FAVER and explains how it is defined using the SoS methodology. Section 5.3 demonstrates the functioning of FAVER through selected use cases. Section 5.4 discusses the pros and cons of the reasoning techniques used in FAVER and presents the details of future work to expand this framework. Section 5.5 summarizes the features of the defined architecture of FAVER and concludes the paper.

5.2 FAVER Architecture with A System-of-Systems Methodology

While [Figure 5-3](#) provides the working concept of FAVER, the architecture of FAVER is shown in [Figure 5-4](#), which combines the DTs with reasoning, along with the supporting modules for communication and processing that make the implementation of FAVER possible. [Figure 5-4](#) shows three different layers, i) Digital Twin layer, ii) Reasoning layer, and iii) Communications layer.

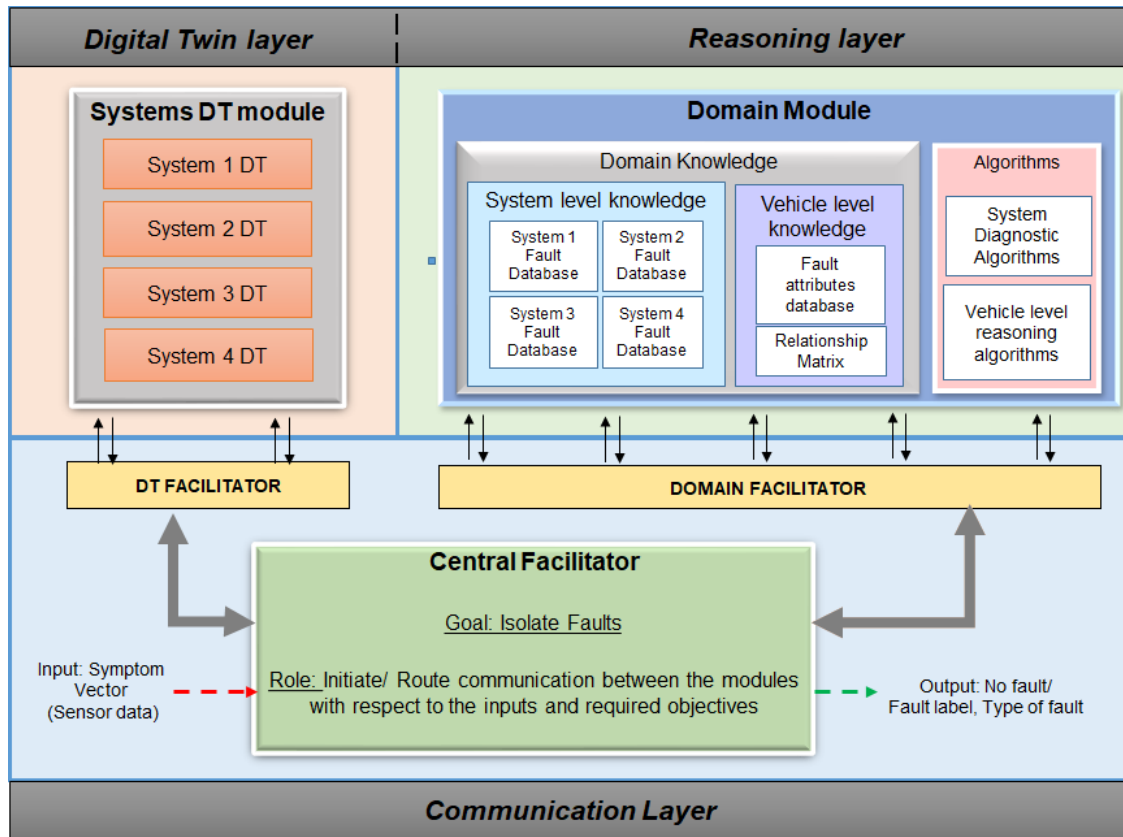


Figure 5-4 The architecture of FAVER

It can be seen from [Figure 5-4](#) that the Digital twin layer and Reasoning layer communicate only through the Communications layer, via the Central facilitator, the Domain facilitator and the DT facilitator. The Digital twin layer and the reasoning layer have modules of their own, but they do not interact with each other, but only through the facilitators. These layers and modules which form the architecture of FAVER are defined and tested using the System-of-Systems (SoS) methodology.

The System-of-Systems (SoS) describe a collection of tasks or systems that are brought together to function towards one complex goal [29]. FAVER adapts a directed SoS [30], wherein FAVER is treated as a platform of system-of-systems, and its individual components like the aircraft system DTs in the DT layer and their diagnostics which are usually independent, work together as SoS for the central purpose of diagnosing faults at the systems level. This is accompanied by the reasoning layer and other components required for communication between the layers (as seen in [Figure 5-4](#)).

Figure 5-5 shows the V- diagram developed for FAVER on the right side, aligned with the components, modules, and layers on the left, that comprise the architecture of FAVER (from Figure 5-4). Saxena and Roychoudhury (2013) [31] provided guidelines for capturing requirements for an IVHM system using a System Engineering methodology and provided a V-diagram, based on which, FAVER's V-process model has been adapted.

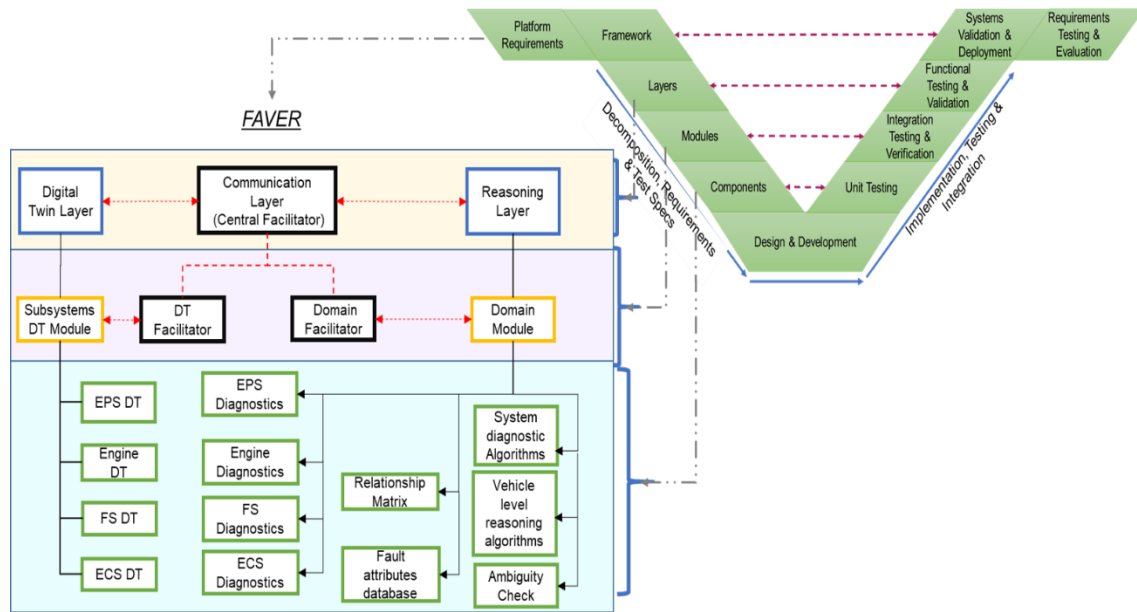


Figure 5-5 Layers and modules of FAVER developed with respect to V- diagram

In the V- process model, the initial stage is always the requirements development and tests definition. This starts from platform FAVER whose requirements and tests are defined first, to satisfy the overall objectives. Once the requirements are captured at the top level, the three layers required to compose the framework are defined in such a way that they satisfy the requirements of the platform. This is followed by capturing the requirements of these layers, which will then be satisfied by defining the subsequent modules and components, listed in Figure 5-5. Tests for verifying if the requirements are met are framed at each stage of the layer/ module/ and the component definition. After the requirements captured and tests defined in a ‘top-down’ approach as seen in the left handle of V- diagram in Figure 5-5, the components and modules are built, integrated and tested to form layers which are then put together to verify the functioning of the platform, FAVER, in a ‘bottom-up’ approach. This forms the right handle of V- diagram.

Three different types of requirements are captured according to the Holistic Requirements Model [32] in viz:

- i. Operational requirements,
- ii. Functional requirements, and
- iii. Non-functional requirements.

The operational requirements define the major purpose of the system, setting up the demand/ goal for the system. This results in defining the basic functionalities which are captured in detail by functional requirements. The non-functional requirements capture the requirements of the systems like scalability and reliability. All three types of requirements are captured for the main layers. Only the functional requirements are captured for the lower-level modules and components, as they collectively meet the operational and non-functional requirements defined for the layers.

The rest of this section discusses the requirements captured and tests defined at every step of the V- diagram from [Figure 5-5](#), as per the SoS methodology. This section also presents the unique properties of certain modules in the architecture and the justification of why such properties are required for FAVER. [Figure 5-4](#) is used as the primer for this section, along with supporting figures.

5.2.1 Platform FAVER requirements

As discussed in section 5.2, to develop the architecture, FAVER is considered as the platform for SoS. The operational, functional and non-functional requirements for building FAVER are captured in [Figure 5-6](#) to match with the overall objectives of FAVER from section 5.1.2.

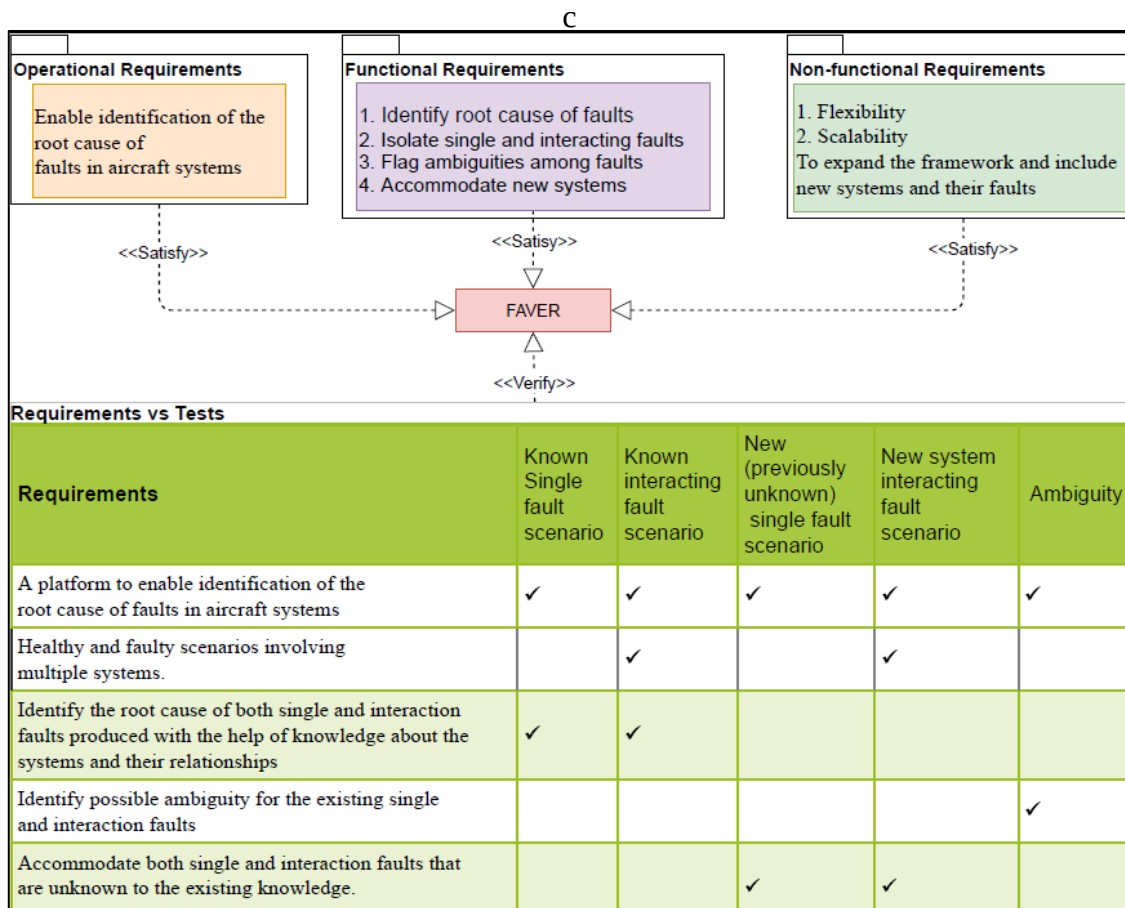


Figure 5-6 Requirements and tests for FAVER platform

Tests: FAVER should be able to meet with all the requirements from Figure 5-6, and it will be tested using several use case scenarios. As FAVER is in the initial conceptual phase of development, it has to be ensured of its ability to expand to include more aircraft systems. Hence, the measures of scalability and flexibility are chosen for non-functional requirements. Once this framework is deployed the other requirements such as reliability, maintainability and runtime can be measured and improved. Some of the developed use case scenarios shown in Figure 5-6 for testing FAVER are as following.

- i) **Known Fault isolation - Single:** A known fault injected in any of the systems should be isolated.
- ii) **known Fault isolation – Interaction:** For a known fault injected in any of the systems that affect another system, both the root cause and the effect should be isolated.

- iii) **New (previously unknown) system Fault isolation – Single:** A single fault that is previously unknown from an unfamiliar system should be accommodated by expanding the framework.
- iv) **New system fault isolation – Interaction:** A fault that is previously unknown from an unfamiliar system that could affect other existing systems should be accommodated by expanding the framework.
- v) **Ambiguous Faults:** When two or more system diagnostics flag for faults at the same time, which might or might not result in the same effect, they should be flagged for ambiguity. If simultaneous faults from two individual systems are flagged, they should be isolated individually, or if they have the same effect on one system. the original cause of the fault has to be isolated,

Having captured the requirements for developing FAVER and defining the tests to verify its functionality, the layers and modules to build the entire platform should be defined to satisfy FAVER's requirements.

To achieve the requirements of FAVER, the platform needs three layers:

- i) **The Digital Twin (DT) layer:** To develop and run simulations of aircraft systems for both single faults as well as interacting fault scenarios. The DT layer is also used to generate data for system diagnostics which is required to develop knowledge for isolating the faults.
- ii) **The Reasoning layer:** This layer is required to collect knowledge from all the concerned aircraft systems. Also, this layer is necessary due to the reasoning techniques and algorithms that will process data from the concerned systems, to isolate single/ interacting faults as well as to flag ambiguities.
- iii) **The Communication layer:** This layer is important in serving as the bridge between the other two layers and to serve as the portal for fetching input from and providing output to the user.

5.2.2 Layer, Modules & Components

The three layers necessary for satisfying the overall requirements of FAVER are discussed in this section, along with their constituent modules and components, as well as their respective requirements and tests.

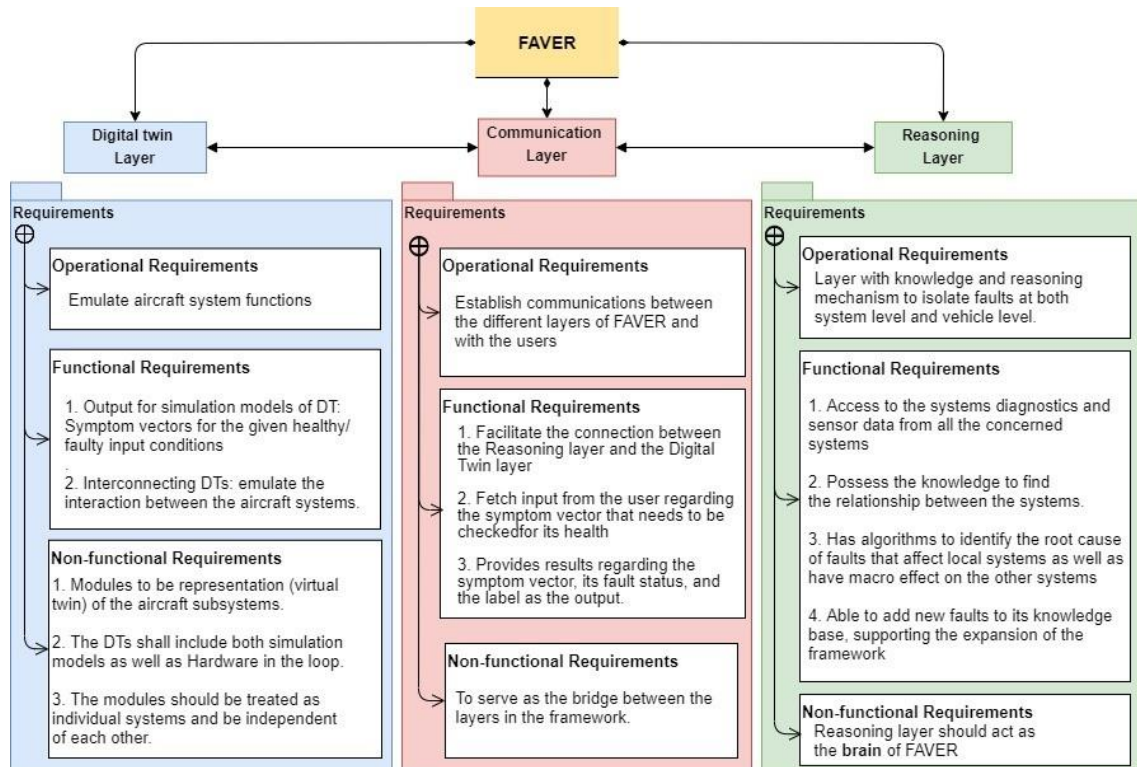


Figure 5-7 Requirements captured for layers in FAVER

5.2.2.1 The Digital Twin (DT) Layer:

As mentioned in section 5.2.1, the DT layer is required to run simulations of the aircraft systems and to generate data required for developing diagnostics for the concerned systems. In order to fulfil their requirements at the platform level, the DT layer has requirements of its own and they are listed in Figure 5-7. As per the V- diagram in Figure 5-5, these requirements of the DT layer are fulfilled by defining the appropriate systems DT module.

5.2.2.1.1 The Systems DT module for the DT layer

Each digital twin of an aircraft system is an individual module, with the function of producing simulations and providing output consisting of the monitored health parameters as symptom vectors and state of health for dependent systems, for the given input simulation conditions. The systems DT module chosen can be either physics-based, function-based models or data-driven representations for emulating the interaction of the aircraft systems. The DT layer also consists of Hardware-in-the-Loop representations of

the systems. The physics-based simulation models would accommodate adjusting of parameters to produce the output that matches the symptom vectors, for the cases of ambiguity. The role of these DTs is to run the simulation based on the input from the DT facilitator and provide the output in the form of a symptom vector required for further reasoning. The systems DT module will have the property of encapsulation and they will be independent of each other, keeping up with the modular characteristic of system-of-systems. To make the systems DT interact with each other, a facilitating module called the Digital Twin (DT) facilitator will be used (as seen in the left corner of the bottom layer in [Figure 5-4](#)). The DT facilitator will receive its commands from the Central facilitator in the communications layer and choose the appropriate system DT for simulation and receive output from that particular DT and send it back to the Central facilitator. Considering the four aircraft systems, viz: the Engine, the Electrical Power System (EPS), the Environmental Control System (ECS), and the Fuel system (FS), from the physical/ Digital twin layer of [Figure 5-3](#), the interaction between the Central facilitator, the DT facilitator, and the systems DT module will look like [Figure 5-8](#).

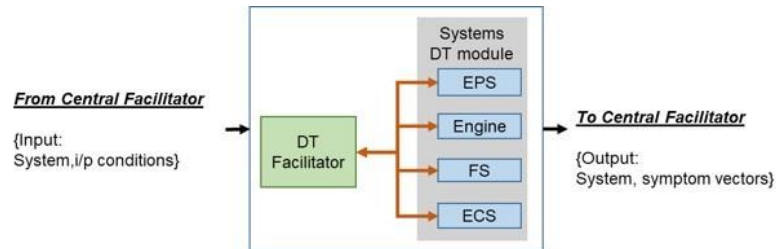


Figure 5-8 Interactions between the DT facilitator and the Systems DT module

This property of encapsulation for the systems DT module has multiple advantages for implementing the framework, which are as follows.

- a) **Scalability:** The encapsulation of the systems DT will help with the scalability issues. Each system cannot be sized to the target aircraft system. As FAVER aims to demonstrate the interaction of multiple aircraft systems, it is impractical to expect all the systems DT to be on a 1:1 scale with the other DTs. In the traditional method, when two or more systems are considered for interaction, they have to be on the same scale. However, the same is not required in the case of a modular arrangement. For example, in the use case scenarios demonstrated in the later section, the Electrical Power System (EPS) DT is a MATLAB simulation model modeled after B737 electrical schematic, while the Fuel System (FS) DT

considered is an experimental rig. The encapsulation of these DTs will enable the transfer of the health state of systems ranging from 0% degradation to 100% degradation, rather than sharing the exact output from the DT, which will not be compatible with the scale of another system DT, in this scenario.

- b) **Complexity:** Consider n system DTs existing within FAVER. When a new system DT (say, $n+1$) is being added to the existing platform, the arrangement of modularity prevents the unnecessary complexity of connecting the $n+1^{\text{th}}$ system DT to every other n^{th} system. Maintaining the independence of the system DTs enables the addition of only necessary interaction among each other.
- c) **Verification & Validation (V&V):** Keeping the modularity of the DTs also helps in saving time concerning V&V of the system DTs. One of the objectives of FAVER is to expand its framework to accommodate new systems and their faults. This involves establishing a connection between the new systems DTs and the existing DTs. Integrating the DTs by hardcoding will require V&V of each new connection made with the existing DTs. However, if the modularity is maintained and the DTs do not communicate with each other directly, only the health states and the behaviours of the interacting systems need verification, and it will save a lot of time involved in the process.

5.2.2.2 Reasoning Layer:

The reasoning layer is required by FAVER for retaining knowledge and for processing data from different systems, as well as to carry out reasoning to isolate single/ interacting faults and flag ambiguities. To achieve these requirements of FAVER, the reasoning layer has a certain set of requirements, which are shown in [Figure 5-7](#). The Domain module in the reasoning layer and its components are to be defined as listed in [Figure 5-5](#), to meet with the reasoning layer's requirements.

5.2.2.2.1 The Domain module in the Reasoning layer

To define the modules required for the reasoning layer in FAVER, the architecture of a generalized reasoning system [9] is adapted. This architecture of generalized reasoning system was developed in the work previously published as a foundation to the research on reasoning in IVHM as mentioned in section 5.1. [Figure 5-9](#) shows the architecture of a generalized reasoning system. It requires i) Domain library module with the databases

and domain knowledge, ii) Problem-solving module with inference engine and reasoner maintenance system, and iii) Tasks module with the objectives.

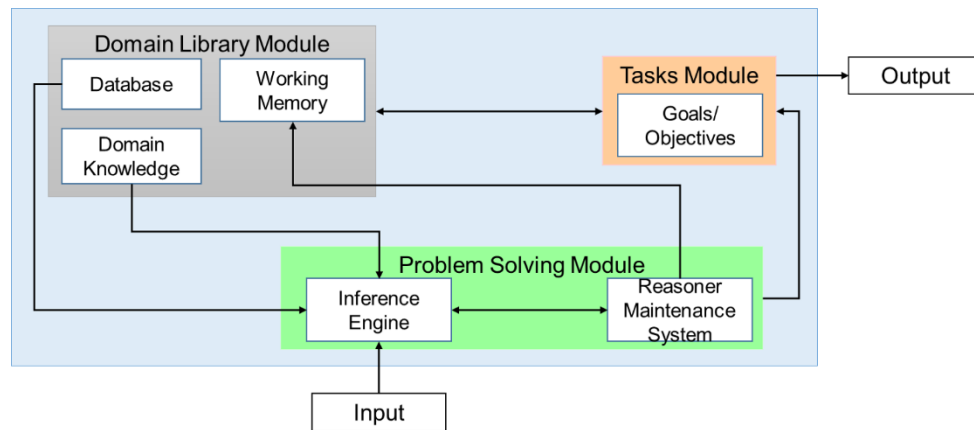


Figure 5-9 A generalized reasoning system architecture [9]

In the case of FAVER's reasoning layer (top right layer of [Figure 5-4](#)), the domain module with knowledge bases, the relationship between systems at the vehicle level in the form of relationship matrix, and the fault attributes list, and the tasks module with algorithms that have the objective of isolating faults are considered. The inference engine from [Figure 5-9](#) gets the input from the user. This module is customized to FAVER's purpose by changing it to a component of the communications layer (bottom layer of [Figure 5-4](#)), namely the Central facilitator, that would receive input from the user and provide output. Further information on the Central facilitator and the communications layer will be discussed in the next subsection.

The Domain Module in the Reasoning layer (top right side of [Figure 5-4](#)) will act as the brain of FAVER. It will have multiple components from the generalized reasoning system ([Figure 5-9](#)) which will be brought together to act as the knowledge repository of the platform. This module will consist of knowledge about the health information of the aircraft systems considered in FAVER. This is possible by having access to the local system diagnostics and sensor data. The vehicle level reasoning carried out by FAVER is built upon the diagnostic capabilities of the local systems, similar to the hierarchical structure presented in section 5.1. The Domain module will be built as an expert system that will have knowledge of the relationship between the systems at the vehicle level.

A novel knowledge base for the expert system: A traditional relationship matrix between systems and their faults will involve experts developing the relationship between each fault in both systems. Searching such a relationship matrix will give the results of whether a particular fault could be caused by any other fault from the other system. The size of such a matrix M with n systems will be as shown in (5-1).

Total number of elements in matrix M,

$$= \left(\sum_{i=1}^n (\text{No. of faults in System } i) \right)^2 \quad (5-1)$$

The matrix M gets disproportionately large as more systems are brought into the framework. For example, consider two systems A and B with two fault modes each, A1, A2, B1, and B2. As per (5-1) a traditional relationship matrix would be of size 4x4, with 16 cells, as seen in Figure 5-10. Consider the scenario where a new system C is added in FAVER with two fault modes C1 and C2. Now, the relationship matrix has to be increased in size to develop the relationship between the 6 faults. Hence the matrix will take the shape of having 36 elements for three systems with two faults each. The experts should develop the relationship for the n+1th system against all the elements belonging to n systems, as shown in (5-2). In Figure 5-10, the number of cells to be updated becomes 20.

No. of new elements to be updated in matrix M,

$$\left(\sum_{i=1}^{n+1} (\text{No. of faults in System } i) \right)^2 - \left(\sum_{i=1}^n (\text{No. of faults in System } i) \right)^2 \quad (5-2)$$

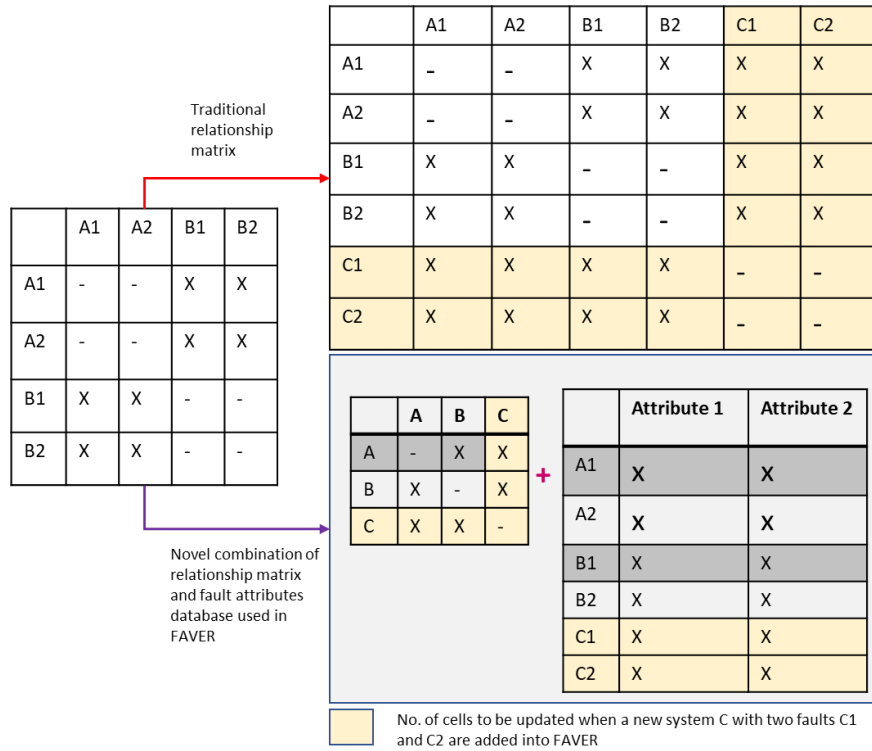


Figure 5-10 Traditional relationship matrix versus the combination of vehicle level relationship matrix with fault attributes database

In order to avoid this scenario, knowledge at the vehicle level is built using two components: i) a relationship matrix at the top level and ii) a fault attributes database. The relationship matrix M will give the cause and effect relationship of only the n systems and not their faults. The details of the system faults will be populated in the fault attribute database, where, for each fault, two attributes will be populated. 'Macro_Effect' attribute refers to the possibility of a fault to affect any other system and 'Macro_Cause' attribute refers to the possibility of fault being caused by any other system. Hence, when a new system, $n+1$ is brought into the framework, size of the existing relationship matrix n^2 gets a simple update as $(n+1)^2$; the number of new elements to be updated in the relationship matrix will be $2n+1$. The experts will have to update the fault attributes database only for two parameters per fault as shown in (5-3).

No. of new elements to be updated in the Fault attributes database for

$n + 1$ th system

(5-3)

$$= (\text{No. of faults in } n + 1\text{th system}) * 2$$

As seen [Figure 5-10](#), when the combination of relationship matrix at vehicle level and a fault attributes database is used, in place of a traditional relationship matrix, the number of cells to be updated in relationship matrix given by $2n+1$ and fault attributes database as given by [\(5-3\)](#) is five and four respectively (instead of 20 as per [\(5-2\)](#)). The difference between the number of new cells to be updated will be huge, when multiple systems with new faults are added to the framework. this will be shown through examples in the following sections in this chapter and in Chapter 8. Thus, this step saves considerable time for the experts to update the knowledge base when new systems are added to the framework.

Advantages: This novel arrangement of knowledge is similar to the traditional relationship matrix, in the sense, it is developed by the experts. However, when new systems are brought into the framework, the relationship matrix at the top level will only increase by one row and one column with $2n+1$ new elements, and the fault attributes database will increase only in rows depending upon the number of faults in the new system. In the case of a new fault from an existing system, the relationship matrix at the top level will not change in its dimensions, and the fault attributes database will have one more row added to it. This provides controlled scalability to an expanding platform like FAVER.

One more advantage of using faults attributes database is that the knowledge that needs to be populated for new fault is only for two parameters: '*Macro_Effect*' and '*Macro_Cause*'. Hence, updating the knowledge database becomes simpler with the fault attributes database. In the case of a traditional relationship matrix, for every new fault, be it cause or effect, the relationship must be populated against all the existing faults, thus consuming more time. Adding new faults for the existing systems or adding new faults from new systems will require only two attributes for each one of them³. The use of fault attributes database rearranges the same available information regarding a fault's cause and effect, in a different method resulting in a more scalable and efficient knowledge database for FAVER.

It is to be noted that FAVER's expert system is built against the previously planned Case Based Reasoning (CBR) system [28]. This is to ensure modularity and enable

³ Detailed examples can be seen in Chapter 8.

simpler modification to the existing knowledge in the expert system. If a CBR system were used as mentioned in the FAVER's methodology paper[28], it was speculated that maintaining the fault database of symptom vectors of all the aircraft systems, in order to be compared with the input symptom vectors directly would be a tedious task and a threat to FAVER's scalability. Hence, further exploration of using CBR system for vehicle level knowledge was dropped, and instead, its working mechanism was adapted for the functioning of the Central facilitator, which will be discussed in the section 5.3.

Apart from the library of vehicle level knowledge, the Domain module contains the algorithm that applies rule based approximate reasoning for known faults isolation and ambiguity checks (as seen in the top right corner of [Figure 5-4](#)). Based on the input provided by the Central facilitator and the knowledge available, this algorithm is used to isolate the root cause of the fault and check for its possible ambiguity with faults from other systems. Further details on reasoning techniques used by FAVER is presented in section 5.3.

5.2.2.3 Communications Layer

The communication layer connects the other two layers as seen in [Figure 5-5](#) (and the bottom layer of [Figure 5-4](#)). This layer is very crucial in the functioning of FAVER, as it is responsible for routing the communication between the layers and their modules. The requirements of the communication layer are as listed in [Figure 5-7](#). To establish the communication between the layers of FAVER, and between the users and FAVER, the communication layer has three main modules called the *facilitators*. As mentioned in the previous subsection, the inference engine part of [Figure 5-9](#) is transformed into the Central facilitator. The communications layer will also have two more modules, namely the Domain facilitator and the Digital Twin (DT) facilitator which will connect the communications from the Central Facilitator to the Reasoning layer and the DT layer, respectively. Various communications taking place between the layers are as follows.

- i) The communication between the DT facilitator and the Central facilitator for interactions with the systems DT module is as shown in [Figure 5-8](#), in section 5.2.2.1.
- ii) With respect to the Domain facilitator, it receives requests from the Central facilitator regarding the relationship between the systems and to access the local system

diagnostics. It also requires access to the reasoning algorithms at the vehicle level, in the cases of interaction effects and for adding new faults to the library.

- iii) The Central facilitator initiates communications with the other facilitators and it functions with the goal of isolating the cascading fault. It fetches the input from the user regarding the symptom vectors of the required systems and uses them for further reasoning process by accessing the knowledge in the Domain module via the Domain facilitator. The Central facilitator also provides the output, back to the user, regarding the symptom vector, which will be the fault label and its type.

An example of these communications between layers is provided in the use case demonstration of FAVER in section 5.3.

5.2.2.4 Tests for Layers

Test Scenarios have been developed to verify the integration and co-existence of all three layers, as shown in [Figure 5-5](#), along with the proper functioning of each layer, to enable the deployment of FAVER at the top level.

- i. **Communication between the Central Facilitator and the user:** The Central facilitator should be able to let the user choose the systems to test and enter the symptom vector for each chosen system. It should also be able to provide the fault label, its system, and its possible cause or effect as the desired output to the user.
- ii. **Communication between the Central Facilitator and the DT Facilitator:** The DT facilitator should be able to choose the correct subsystem DT based on the input from the Central Facilitator and provide the output back to the Central Facilitator.
- iii. **Communication between the Central Facilitator and the Domain Facilitator:** The Central facilitator should send a query about the relationship between the systems, as well the system diagnostics to the Domain facilitator, and receive the required results.

5.2.2.5 Tests for modules

- i. **Communication between the DT Facilitator and the Systems DT module:** The DT facilitator should be able to initiate simulation on the correct system DT and receive the simulation results.
- ii. **Communication between the Domain Facilitator and the Domain module:** The Domain Facilitator should be able to direct the communication to components in the Domain module based on the request received from the Central Facilitator.

5.3 FAVER Architecture & Working Mechanism for Known Faults with Use cases

The previous section presented the architecture of FAVER, developed in an SoS methodology, through which the requirements were captured for FAVER and all the components and modules defined along with their testing criteria. In this section, the general working mechanism to achieve FAVER's objectives regarding the known fault scenarios is presented, with the help of two systems i) the Fuel System (FS), and ii) the Electrical Power System (EPS).

The Digital Twin (DT) Layer: In the DT layer, the FS is considered as a Hardware-in-the-loop and is represented by the fuel test rig. Figure 5-11a shows the schematic of the fuel rig used to run the experiments. The experimental setup in the IVHM Center's fuel rig [33] is closed-loop, where fuel (water) is pumped from a reservoir R by a motor-driven pump P that has an internal relief valve, passing through a shut-off valve S, followed by another valve V, a filter F, flowmeters FM1, FM2, a nozzle N and back to the reservoir R. Several experiments are run using this rig for healthy and faulty conditions using LabView, with the objective of constant flow output. The data is processed via MS EXCEL and the diagnostics function is developed in MATLAB R2019a. Further details on the experiments on fuel rig can be seen in Chapter 7 [34].

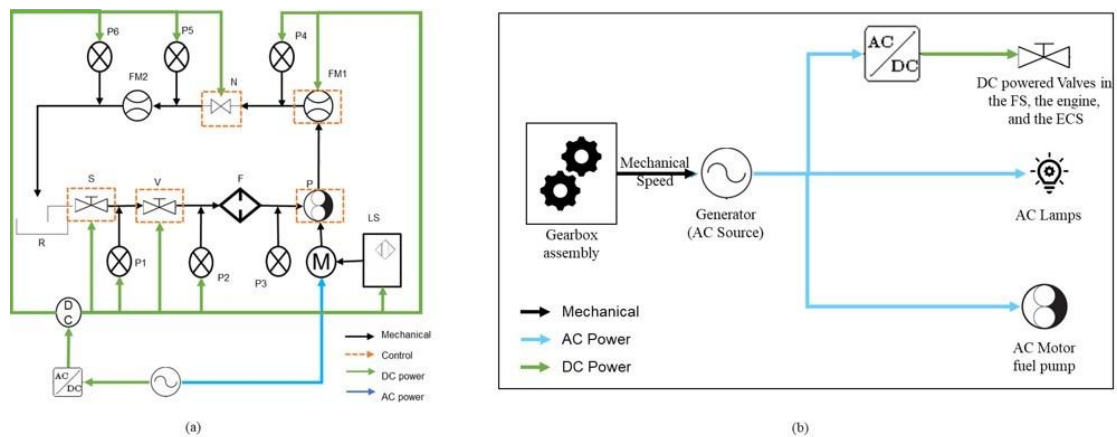


Figure 5-11 Schematics of the aircraft systems used for FAVER demonstration

(a) Fuel rig layout (b) Electrical Power System layout

For the EPS, a MATLAB Simulink R2019a model is developed to emulate the B737 EPS [35] as shown in Figure 5-11b. It is modeled to generate and supply both AC and DC power to the nodes like the FS, the ECS, and the engine (Figure 5-11b). The faults

are injected through switches and circuit breakers that connect the power supply to these nodes. The diagnostics of the EPS is developed in MATLAB R2019a. The details of the experiments and simulations conducted for the EPS, and its diagnostics are covered in detail in Chapter 6 [36]. Hence, only the outline schematics are provided, and the results are used for this demonstration. It is to be noted that the two aircraft system DTs are of different scales and yet, they are used in FAVER to inspect their interaction behaviours. This is possible due to the modularity feature of FAVER architecture, where the system behaviours are transferred through their diagnostic results for vehicle level reasoning. This feature ensures the flexibility of FAVER to be able to integrate system DTs of any format for vehicle level reasoning.

Figure 5-12 shows the activities occurring in FAVER, during the isolating the known faults. It can be seen from Figure 5-12 that the engineer/ user sets up the Digital twin layer, with the FS and the EPS, contributing to the diagnostics in the reasoning layer for each of these systems.

The Reasoning Layer: The reasoning layer consists of two levels of knowledge as mentioned in section 5.2.2.2.1. The system level knowledge is in the form of model based or data-driven diagnostics of the respective systems, for the chosen fault modes. The system level diagnostics for both the EPS and the FS are a part of the Domain module in the reasoning layer. These diagnostic algorithms are accessed by the Domain facilitator by developing functions on top of the existing diagnostic algorithms in MATLAB R2019a. For demonstration purposes, three fault modes are chosen for each system (as shown in Figure 5-12). In case of the fuel system, the fault modes chosen are as follows:

- i) **Clogged filter:** Clogging filter in a fuel line obstructs the flow. This results in increased pump speed in order to meet the demanded flow output. This fault mode is not influenced from any other system. The degradation is assumed not to be severe enough to affect the electricity supplied to the pump motor. This is only to investigate the scenario that has purely local effects and not macro effects. This assumption can be updated at any stage by updating the expert system knowledge.
- ii) **Sticking valve:** This fault mode is similar to the clogged filter in terms of symptoms and output. However, this fault can be influenced by the EPS, where, if the switch

transferring power to this valve gets disconnected, the valve will be stuck in its position without fully open, thus restricting the flow.

- iii) **Reduced flow output:** Despite the objective of this experiment being the criteria of demanded flow output, if the pump is not powered fully, the objective will not be met, resulting in reduced flow as output. The pump can be running at its lower capacity, due to several reasons. For this demonstration, it is assumed that the supply voltage for the pump from the EPS is low.

For the EPS, the chosen fault modes for the demonstration are:

- i) **AC Cabin light switch open (FluroLamp):** This fault occurs when the switch supplying power to the AC lights in the aircraft cabin is open. This fault affects only the local system (the EPS) and does not have any macro effect.
- ii) **FS Valve switch stuck open (FSV):** In this fault mode, the switch transferring the DC power to the valve in the fuel system is open. This leads to the FS valve stuck at its position, resulting in the macro effect on the valve in the fuel system.
- iii) **Reduced voltage to the AC motor pump:** This fault mode results in reduced pump speed which in turn results in the macro effect of reduced output flow in the fuel rig.

The vehicle level knowledge for the EPS and the FS is made up of an expert system comprising of

- i) A relationship matrix between the EPS and the FS: A matrix between the EPS and the FS is created to represent the relationship between the systems where the EPS affects the FS, but the FS doesn't affect the EPS in turn. Since the EPS provides electricity required for the pump and the valves in the FS, there is no direct interaction from the FS to the EPS. While the relationship matrix looks simple in this example, it expands when more systems are added to the framework, as observed in Chapter 8.
- ii) A fault attributes database that has populated all the faults tested in both these systems and also the possibility of macro cause and macro effects based on the expert's knowledge ([Figure 5-12](#)).

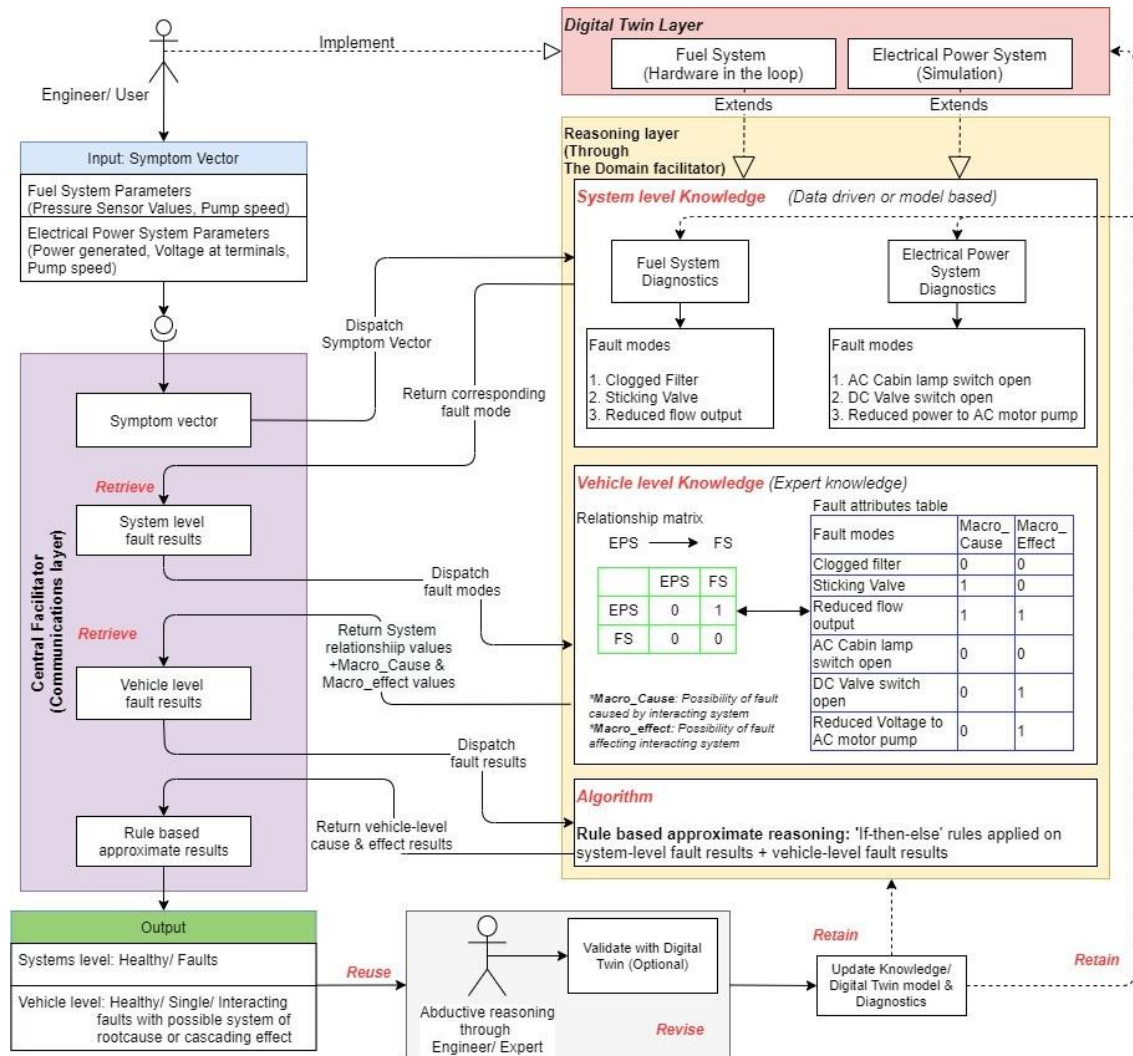


Figure 5-12 Activity diagram of FAVER to isolate known faults and ambiguity

The fault attributes database has the list of faults from both the EPS and the FS, along with the attributes, 'Macro_Cause' and 'Macro_Effect'. The rank '1' in 'Macro_Cause' column means that there is a possibility that the corresponding faults could have been caused by another system, while the rank '0' means there is no possibility of 'Macro_Cause' for that faults. The same is implied for the 'Macro_Effect' column, where the rank '1' means the corresponding faults could be able to affect another system, and the rank '0' implies no such possibility.

The relationship matrix and the fault attributes database together make the library required for the vehicle level reasoning for FAVER (Figure 5-12). Having set up both the reasoning and the DT layers for the EPS and the FS, the next step to complete the vehicle level reasoning process in FAVER is the Central facilitator

The Central Facilitator: Figure 5-12 shows the different steps carried out by the Central facilitator to isolate the faults. The Central facilitator is developed in MATLAB R2019a, where an input dialog box is created to choose the systems that are to be tested for reasoning in FAVER. The inputs required are the parameters that are necessary for running the local system diagnostic functions. The general operating mechanism of FAVER follows a Case Based Reasoning (CBR) strategy (retrieve, revise, reuse, retain) [9]. The first stage is to retrieve knowledge from the reasoning layer regarding the system level faults and the vehicle level relationship between the systems. Having fetched the inputs from the user/ engineer, in the form of symptom vectors, the Central facilitator establishes a connection with the local system diagnostics in the reasoning layer, via the Domain facilitator. This is carried out by developing functions on top of the diagnostic programs developed in MATLAB. These functions carry the symptom vector from the Central facilitator and run their diagnostics and send results as output back to the Central facilitator. The results from both the diagnostic functions are then consolidated and sent back to the domain facilitator, to consult with the library modules: the relationship matrix and the fault attributes database (Figure 5-12), to reason through the connection between both faults, and the Domain facilitator sends the result back to the Central facilitator. This step forms the ‘Retrieve’ stage of CBR.

This is followed by the ‘forward chaining’ mechanism of rule based reasoning, wherein the goals of identifying whether the known faults are single or interacting, and clearing the ambiguity are achieved with a set of ‘if-then-else’ rules. The rules are defined to provide ‘approximate’ results, which are then treated as evidence for an incomplete hypothesis by the experts to provide the final results, i.e. abductive reasoning. This forms the ‘Reuse’ stage of CBR. Any changes required in the diagnostics or the DTs are updated by the expert, i.e., the ‘Revise’ stage of CBR, followed by the ‘Retain’ stage where the updated knowledge and the models are retained.

While Figure 5-12 displays the list of activities carried out within the architecture of FAVER, along with the different reasoning mechanisms used to isolate the root causes of interacting faults and their cascading effects, Figure 5-13 shows the different use case scenarios developed using the EPS and the FS. From section 5.2.1, three out of five tests are considered for satisfying the platform requirements of FAVER: i) Known fault isolation: Single, ii) Known fault isolation: interaction, and iii) Ambiguous faults. Four

use cases with a total of six scenarios are developed for testing these requirements, which are as shown in Figure 5-13.

- i) Both the EPS and the FS are healthy
- ii) The EPS has a single fault and the FS is healthy. The FS has a single fault and the EPS is healthy (*macro_cause*, *macro_effect* ranks are ‘0’ in both cases)
- iii) Both the EPS and the FS have faults simultaneously, but they do not have macro effects, giving rise to ambiguous scenario (*macro_cause*, *macro_effect* ranks are ‘0’ in both faults)
- iv) Both the EPS and the FS have faults simultaneously and the *macro_effect* rank of fault from the EPS and the *macro_cause* rank of fault from the FS are ‘1’.

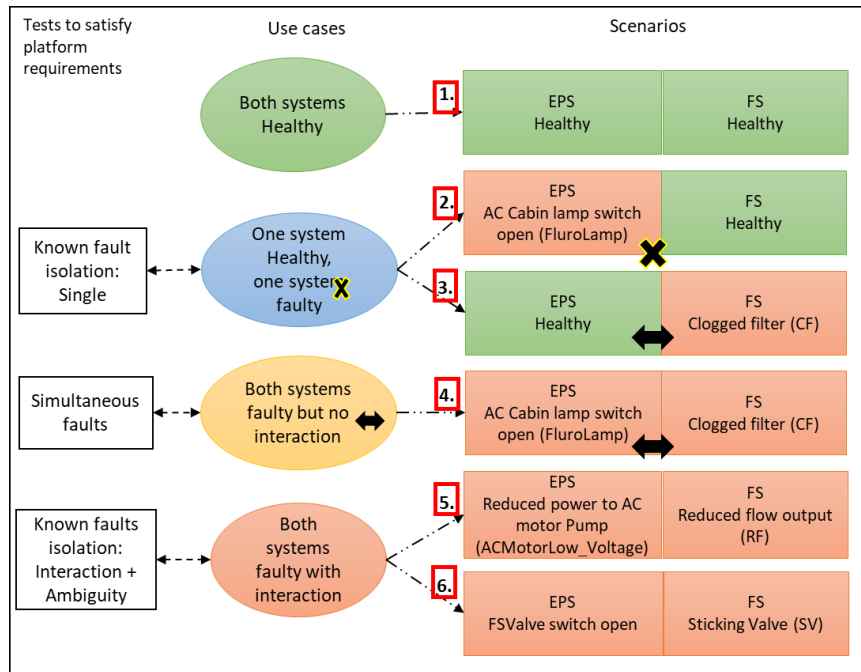


Figure 5-13 Use case scenarios for FAVER demonstration

The above-mentioned use case scenarios are tested in FAVER through the steps described in Figure 5-12. Figure 5-14 shows the results for the six scenarios mentioned in Figure 5-13 (Scenario numbers are highlighted in red colored boxes in both Figure 5-13 and Figure 5-14). In Figure 5-14, result 1 is for the scenario of both systems being healthy (scenario 1 from Figure 5-13). Result 2 is for ‘AC Lamp switch open’ fault in the EPS and result 3 is for the ‘Clogged filter’ fault in the FS (scenarios 2 and 3 from Figure 5-13). For the fourth scenario in Figure 5-13, as both the symptom vector inputs will be faulty,

both the system diagnostics will be flagged. This is now solved by checking the vehicle level expert knowledge, especially the fault attributes database. Since the ranks of *Macro_Cause* and *Macro_Effect* for both the faults, (AC Cabin lamp switch open in the EPS, and the clogged filter in the FS) are '0' in the fault attributes database in Figure 5-12, these two are not interacting faults. The corresponding result can be seen from result 4 in Figure 5-14. For the fifth case in Figure 5-13, both the system diagnostics will be flagged resulting in ambiguity. With the expert knowledge at the vehicle level, based on the *Macro_Cause* and the *Macro_Effect* ranks of the concerned faults, the 'Sticking valve' fault in the FS and the 'FS Valve switch open' fault in the EPS are suggested as interacting faults by the rule based approximate reasoning in the central facilitator (as seen in Figure 5-14). The sixth scenario from Figure 5-13 have a similar reasoning process and corresponding results are shown in Figure 5-14. In this case, however, both the *Macro_Cause* and the *Macro_Effect* attributes of the fault 'Reduced pump flow' from the FS have the rank '1'. This is inferred by the reasoning as a fault with the possibility of root cause from the EPS and the effect cascading to any other system that the FS is interacting with.

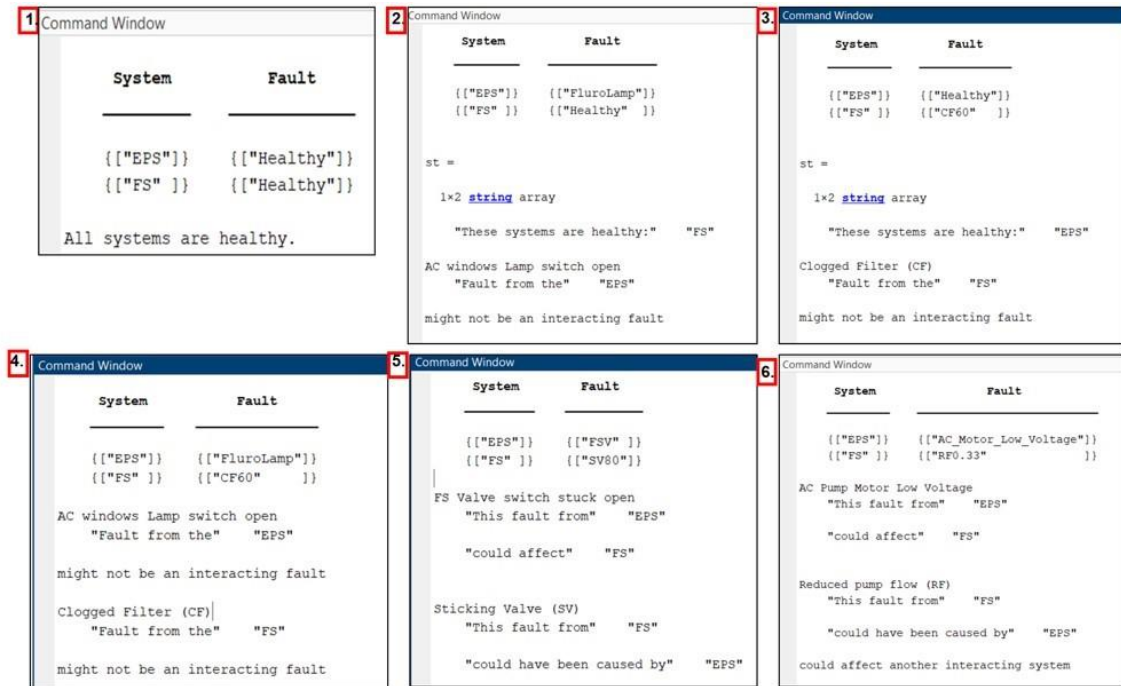


Figure 5-14 Results generated by the Central Facilitator in FAVER for the use case scenarios

5.4 Discussion & Future Work

The previous section presented the working mechanism of FAVER through its architecture and demonstrated its functioning through multiple use cases. FAVER implements a combination of reasoning techniques like the CBR, rule based approximate reasoning, and abductive reasoning to process the symptom vectors from multiple systems, identifying their health states, and narrowing down the list of possible root causes and cascading effects that could have occurred due to their interaction between the systems. It can be observed that from [Figure 5-14](#) that the results generated by the reasoning in FAVER are only approximate. These serve as evidence to the expert/engineer who will, in turn, make the final decisions on the exact root cause and the effect of the isolated fault, thus being the key to abductive reasoning, as shown in [Figure 5-12](#). While this reasoning process reduces the accuracy of the results produced by FAVER, it ensures that the user/engineer gets the narrowed list of possible system interactions to focus on, rather than troubleshooting of faults tracing back to every connected system, thus reducing the downtime and associated cost in maintenance.

In this paper, the most requirements captured are verified at different stages as per [Figure 5-5](#) for satisfying the requirements of FAVER, and they are summarized in [Table 5-1](#).

At modules level: The modules are defined in such a way to satisfy the requirements for the layers. The systems DT module consists of the representations including the DTs and hardware-in-the-loop that emulate the performance of aircraft systems, and provide simulations for healthy and faulty scenarios, as required by the DT layer in [Figure 5-7](#). Similarly, the Domain module has access to system diagnostics, possesses vehicle level knowledge, and the algorithm for vehicle level reasoning, and acts as the brain of FAVER. The communication between the Domain facilitator and the Domain module (from section 5.2.2.5) is demonstrated through the use cases in section 5.3.

At layers level: From section 5.2.2.4, the following tests are verified through the use case demonstration in the previous section:

- i) Communication between the Central facilitator and the user (by fetching the symptom vector and providing the output)

- ii) Communication between the Central facilitator and the Domain facilitator (fetching system level and vehicle level knowledge from the reasoning layer)

Besides, the systems DT module with encapsulation features and the Domain module with the novel expert system knowledge ensure the scalability of the framework, as required by the DT layer and the Reasoning layer respectively, to satisfy the overall requirement of FAVER as captured in Figure 5-6.

At platform level: Three tests out of five from section 5.2.1 (Figure 5-6), viz: i) known fault isolation-single, ii) known fault isolation- interaction, and iii) ambiguity are carried out in section 3 to verify that FAVER's requirements are met.

Table 5-1 Satisfied requirements of FAVER

REQUIREMENTS SATISFIED IN THIS PAPER	
FAVER (AS A PLATFORM) REQUIREMENTS	Operational & Functional Identified root cause of faults between the EPS and the FS <i>Faults isolation:</i> Isolated single and interacting faults in the EPS and the FS. <i>Root cause identification:</i> Identified the root causes of interacting faults between the EPS and the FS <i>Ambiguity:</i> Resolved ambiguities when two system diagnostics flagged for faults, by finding the root cause. <i>Accommodate new systems:</i> Will be demonstrated in the future work Non-functional <i>Flexibility:</i> In the demonstration of interacting faults, the FS DT is HIL and the EPS DT is a MATLAB simulation model <i>Scalability:</i> Will be demonstrated in the future work
DT LAYER REQUIREMENTS	Operational & Functional <i>Emulate aircraft system functions:</i> The EPS DT and the FS DT are representatives of aircraft systems <i>Output:</i> Symptom vectors from the EPS and the FS DTs used in demonstration for healthy and faulty scenarios to be used for FAVER reasoning <i>Interconnecting DTs:</i> the EPS supplying power to the FS Pump and Valves are simulated in the EPS DT Non-functional <i>Modules:</i> The EPS and the FS DT are independent of each other and functionable on their own. The EPS DT is a simulation model and the FS DT is a HIL

COMMUNICATION LAYER REQUIREMENTS	<p>Operational & Functional Established communications between the Central facilitator and the reasoning layer. Fetched input symptom vector from the user and provided output regarding the health status, with fault label and possible root causes and cascading effects.</p> <p>Non-functional Serves as a bridge between the user and the reasoning layer. No direct communication between the reasoning layer and the DT layer. However, the connection between the Central facilitator and the DT layer is not established yet and currently carried out manually.</p>
REASONING LAYER	<p>Operational & Functional Contains knowledge of system level diagnostics, vehicle level knowledge in expert system. Contains Rule based reasoning algorithm to identify faults with micro and macro effects.</p> <p>Expansion: Will be demonstrated in the future work</p> <p>Non-functional Serves as the brain of FAVER, possessing all the knowledge required for the functioning of FAVER.</p>

There are two more tests at the platform level, which focus mainly on bringing the previously unknown systems into the framework, i.e. expanding FAVER for accommodating new systems and new faults. This will involve

- i) Capturing the requirements for including new systems into the framework
- ii) Testing the scalability of the devised reasoning in the Central facilitator
- iii) Capturing the required changes/ updates to be made in the reasoning and the DT layers of FAVER
- iv) Testing and validating the requirements

These tests will be carried out as the next step forward, by considering the engine and the ECS into the framework, as shown in [Figure 5-3](#). This action will increase the complexity of the reasoning algorithms used. One of the important issues to be taken care of while expanding the framework in the case of ambiguity. With more systems involved, the faults from these systems will be more ambiguous; the reasoning algorithm in FAVER should be able to reason through the vehicle level knowledge in the form of relationship matrix and the fault attributes database to address these ambiguity issues. Tests for communication between the systems DT and the DT facilitator at the modules level, and also between the Central facilitator and the DT layer at the layers level are not

demonstrated in this paper, as currently, the DT layer is connected to the Central facilitator manually, and not through automation.

It is also worth noting that the expert systems, in general, are time-consuming and dependent completely on experts to update the knowledge. An alternate method for updating the knowledge using data-driven reasoning should also be explored as part of future work.

5.5 Summary

To summarize, this paper highlights the following:

- i) The general reasoning architecture of IVHM systems for aircraft is presented, along with the current gaps in the field, especially in the areas of cascading faults.
- ii) A framework for aerospace vehicle reasoning, FAVER, is the framework that integrates the power of reasoning with the versatility of Digital Twins (DTs) to isolate single and cascading faults at the aircraft level, addressing the above gap.
- iii) FAVER is treated as a platform, and its architecture (Figure 5-4) is defined and developed using a System-of-Systems (SoS) methodology.
- iv) A detailed V- process model (Figure 5-5) is used for SoS methodology to capture the requirements of FAVER, to meet its objectives and test criteria for development, integration, and testing of the architecture.
- v) One of the important features is the modularity of FAVER's architecture (Figure 5-4); the modules, especially those from the DT layer, are independent and encapsulated, adhering to the characteristics of the SoS. Besides, this modularity helps with the complexity, scalability, and verification & validation issues of FAVER.
- vi) A novel combination of relationship matrix with fault attributes database is introduced in the place of traditional relationship matrix in the expert system for the vehicle level domain knowledge of FAVER. This combination helps in reducing the complexity and time involved in developing the expert's knowledge when the framework is expanded.
- vii) The working mechanism of FAVER using rule based approximate reasoning and abductive reasoning combined with a case based reasoning strategy is presented through a step-by-step activity diagram in Figure 5-12.
- viii) A demonstration of FAVER is provided with two systems: the EPS and the FS. For the demonstrations, the EPS DT used is a MATLAB simulation model, while the FS

is a hardware-in-the-loop. This is done to show the flexibility of FAVER in providing a platform for any type of system representation to be considered for the vehicle level health reasoning. The developed use case scenarios meet with three out of five requirements of FAVER, as listed in [Figure 5-6](#) (known single faults, known interaction faults, and ambiguity). The use scenarios are presented in the demonstration, as shown in [Figure 5-13](#) and [Figure 5-14](#).

The demonstration of FAVER showed the isolation of known faults and solving of ambiguity when two aircraft systems are considered. The pros and cons of the reasoning techniques used in FAVER are discussed. The tests that are verified for the captured requirements of FAVER are listed. The details of the work to expand FAVER to include the engine and the ECS into the framework in Chapter 8 are also presented. It is to be noticed that, in the current demonstration, the DT modules are accessed for validation only manually. Methods to include the validation with the DTs automatically should also be explored at the later stage of developing FAVER. Once these are tested and expanded, the framework can be complete with automated reasoning. Such a complete framework with automated reasoning to isolate faults at the aircraft level, using digital twins for validation will certainly fill the gap of diagnosing cascading faults in IVHM systems. Addressing this gap would help save time and cost associated in unexpected downtime and maintenance activities, thus fulfilling the objective of IVHM.

REFERENCE

1. Ezhilarasu CM., Skaf Z., Jennions IK. Understanding the role of a Digital Twin in Integrated Vehicle Health Management (IVHM). IEEE International Conference on Systems, Man and Cybernetics (SMC). Bari, Italy: IEEE; 2019. pp. 1484–1491. Available at: DOI:10.1109/SMC.2019.8914244
2. Chang S., Wang Y. Civil aircraft IVHM system analysis using model based system engineering. 2017 Second International Conference on Reliability Systems Engineering (ICRSE). IEEE; 2017. pp. 1–5. Available at: DOI:10.1109/ICRSE.2017.8030800
3. Gorinevsky D., Smotrich A., Mah R., Srivastava A., Keller K., Felke T. Open architecture for integrated vehicle health management. AIAA Infotech at Aerospace 2010. 2010; (April): 1–8. Available at: DOI:10.2514/6.2010-3434
4. Keller K., Wiegand D., Swearingen K., Reisig C., Black S., Gillis A., et al. An

- architecture to implement integrated vehicle health management systems. AUTOTESTCON (Proceedings). 2001; (3 14): 2–15. Available at: DOI:10.1109/autest.2001.948902
5. Scandura PA. Integrated Vehicle Health Management as a system engineering discipline. 24th Digital Avionics Systems Conference. Washington, DC: IEEE; 2005. p. 10. Available at: DOI:10.1109/DASC.2005.1563450
 6. Hernandez L., Mullins M., Morris C., Keller K. A Framework for Developing an EPS Health Management System. SAE International Journal of Aerospace. 2010; 3(1): 2010-01–1725. Available at: DOI:10.4271/2010-01-1725
 7. Roemer M. A Hierarchical Reasoning Structure to Support Aerospace IVHM. Aerospace Technology Conference and Exposition. 2011; 120(1EJ): 1176. Available at: DOI:- 10.4271/2011-01-2665
 8. Bharadwaj, Raj; Mylaraswamy, Dinkar; Cornhill, Dennis; Biswas G., Koutsoukos, Xenofon; Mack D. Vehicle Integrated Prognostic Reasoner (VIPR) Final Report. NASA Langley Research Centre; 2013. Available at: <https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20130011181.pdf>
 9. Ezhilarasu CM., Skaf Z., Jennions IK. The application of reasoning to aerospace Integrated Vehicle Health Management (IVHM): Challenges and opportunities. Progress in Aerospace Sciences. Elsevier Ltd; 2019; 105(January): 60–73. Available at: DOI:10.1016/j.paerosci.2019.01.001
 10. Discenzo F., Nickerson W. Open systems architecture enables health management for next generation system monitoring and maintenance. OSA-CBM Development Group; 2001. pp. 1–12. Available at: DOI:10.1.1.123.1206&rep=rep1&type=pdf
 11. Keller K., Amo A Del., Jordan B. Aircraft Electrical Power Systems Prognostics and Health Management (AEPHM). SAE Technical Papers Series. 2004; Available at: DOI:2004-01-3162
 12. Chidambaram B., Horn PL., Gilbertson DD., Pigg P., Talbot MA., Cerise KC. Recorders, Reasoners and Artificial Intelligence – Integrated Diagnostics on Military Transport Aircraft. Aerospace Conference. IEEE; 2006. pp. 1–9. Available at: DOI:10.1109/AERO.2006.1656074
 13. Del Amo A., Keller K., Swearingen K. General reasoning system for health management. Annual Conference of the North American Fuzzy Information

- Processing Society - NAFIPS. 2005; 2005: 19–24. Available at: DOI:10.1109/NAFIPS.2005.1548500
14. Balaban E., Saxena A., Narasimhan S., Roychoudhury I., Koopmans M., Ott C., et al. Prognostic Health-Management System Development for Electromechanical Actuators. *Journal of Aerospace Information Systems*. 2015; 12(3): 329–344. Available at: DOI:10.2514/1.I010171
 15. Azam M., Ghoshal S., Deb S., Pattipati K., Haste D., Mandal S., et al. Integrated diagnostics and time to maintenance estimation for complex engineering systems. *Aerospace Conference, Big SKy*. IEEE; 2014. Available at: DOI:10.1109/AERO.2014.6836478
 16. Mackey R., Brownston L., Castle JP., Sweet A. Getting diagnostic reasoning off the ground: Maturing technology with TacSat-3. *IEEE Intelligent Systems*. 2010; : 27–35. Available at: DOI:10.1109/MIS.2010.124
 17. Byington CS., Kalgren PW., Johns R., Beers RJ. Embedded Diagnostic/Prognostic Reasoning and Information Continuity for Improved Avionics Maintenance. *IEEE Systems Readiness Technology Conference. Proceedings AUTOTESTCON*. 2003. pp. 518–524. Available at: DOI:10.1109/AUTEST.2003.1243595
 18. Papadopoulos PM., Reppa V., Polycarpou MM., Panayiotou CG. Distributed Diagnosis of Actuator and Sensor Faults in HVAC Systems. *IFAC-PapersOnLine*. Elsevier B.V.; July 2017; 50(1): 4209–4215. Available at: DOI:10.1016/j.ifacol.2017.08.816
 19. Daigle M., Bregon A., Roychoudhury I. Qualitative Event-based Diagnosis with Possible Conflicts Applied to Spacecraft Power Distribution Systems*. *IFAC Proceedings Volumes*. IFAC; January 2012; 45(20): 265–270. Available at: DOI:10.3182/20120829-3-MX-2028.00084
 20. Daigle MJ., Koutsoukos XD., Biswas G. Distributed Diagnosis in Formations of Mobile Robots. *IEEE Transactions on Robotics*. April 2007; 23(2): 353–369. Available at: DOI:10.1109/TRO.2007.895081
 21. Su R., Wonham WM., Kurien J., Koutsoukos X. Distributed diagnosis for qualitative systems. *Sixth International Workshop on Discrete Event Systems*, 2002. *Proceedings. IEEE Comput. Soc*; 2002. pp. 169–174. Available at: DOI:10.1109/WODES.2002.1167684

22. Sleight PA., Carter RDG. AIRCRAFT ACCIDENT REPORT 1/2010. 2014. Available at: DOI:<https://doi.org/EW/C2008/01/01>
23. Conradi K. AAIB Annual Safety Report 2015. 2015. Available at: https://assets.publishing.service.gov.uk/media/5656b9e2ed915d0367000008/Annual_Safety_Report_2015.pdf
24. South African Civil Aviation Authority. Aircraft Incident Report. 2008. Available at: <https://www.skybrary.aero/bookshelf/books/827.pdf>
25. A.N.Srivatsava., R.W.Mah., R.Bharadwaj., D.Mylaraswamy. Vehicle-Level Reasoning Systems. In: Jennions IK (ed.) Integrated Vehicle Health Management: The Technology. SAE Publications; 2013. pp. 139–155. Available at: <http://books.sae.org/r-429/>
26. Nwadiogbu EO., Mylaraswamy D., Menon S., Voges HC., Hadden G. US008346429B2: Vehicle Health Monitoring System Architecture for Diagnostics and Prognostics Disclosure. United States Patent; 2013. Available at: DOI:10.1016/j.(73)
27. Khan F., Eker OF., Sreenuch T., Tsourdos A. Health Index and Behaviour Based Vehicle Level Reasoning System. Conference on Prognostics and Health Management (PHM). IEEE; 2015. Available at: DOI:10.1109/ICPHM.2015.7245042
28. Ezhilarasu CM., Skaf Z., Jennions IK. Progress towards a Framework for Aerospace Vehicle Reasoning (FAVER). Proceedings of the Annual Conference of the PHM Society. 2019; 11(1): 1–9. Available at: DOI:<https://doi.org/10.36001/phmconf.2019.v11i1.887>
29. Nielsen CB., Larsen PG., Fitzgerald J., Woodcock J., Peleska J. Systems of systems engineering: Basic concepts, model-based techniques, and research directions. ACM Computing Surveys. 2015; 48(2): 1–41. Available at: DOI:10.1145/2794381
30. Vaneman WK. The system of systems engineering and integration «vee» model. 10th Annual International Systems Conference, SysCon 2016 - Proceedings. IEEE; 2016; (1): 1–7. Available at: DOI:10.1109/SYSCON.2016.7490599
31. Saxena A., Roychoudhury I., Goebel K., Lin W. Towards Requirements in Systems Engineering for Aerospace IVHM Design. AIAA Infotech@Aerospace

- (I@A) Conference. Reston, Virginia: American Institute of Aeronautics and Astronautics; 2013. pp. 1–15. Available at: DOI:10.2514/6.2013-4659
32. Burge S. The Systems Engineering Tool Box. 2011. pp. 1–20. Available at: <https://www.burgehugheswalsh.co.uk/Uploaded/1/Documents/Holistic-Requirements-Model-Tool-v2.pdf> (Accessed: 30 November 2019)
 33. Yufei Lin. System Diagnosis using a Bayesian Method. PhD Thesis, Cranfield University; 2017.
 34. Ezhilarasu CM., Skaf Z., Jennions IK. A Generalised Methodology for the Diagnosis of Aircraft Systems. IEEE Access. 2021; 9: 11437–11454. Available at: DOI:10.1109/ACCESS.2021.3050877
 35. Boeing. THE BOEING 737 TECHNICAL SITE. Available at: <http://www.b737.org.uk/electrics.htm> (Accessed: 3 March 2020)
 36. Ezhilarasu CM., Jennions IK. A System-Level Failure Propagation Detectability Using ANFIS for an Aircraft Electrical Power System. Applied Sciences. 2020; 10(8): 2854. Available at: <https://doi.org/10.3390/app10082854>

6 A System-Level Failure Propagation Detectability Using ANFIS for an Aircraft Electrical Power System

The Electrical Power System (EPS) in an aircraft is designed to interact extensively with other systems. With a growing trend towards more electric aircraft, the complexity of interactions between the EPS and other systems has grown. This has resulted in an increased necessity of implementing health monitoring methods like diagnosis and prognosis of the EPS at the systems level. This paper focuses on developing a diagnostic algorithm for the EPS to detect and isolate faults and their root causes that occur at the Line Replaceable Units (LRUs) connecting with aircraft systems like the engine and the fuel system. This paper aims to achieve this in two steps: (i) developing an EPS digital twin and presenting the simulation results for both healthy and fault scenarios, (ii) developing an Adaptive Neuro-Fuzzy Inference System (ANFIS) monitor to detect faults in the EPS. The results from the ANFIS monitor are processed in two methods: (i) a crisp boundary approach, and (ii) a fuzzy boundary approach. The former approach has a poor misclassification rate; hence the latter method is chosen to combine with causal reasoning for isolating root causes of these interacting faults. The results from both these methods are presented through examples in this chapter.

6.1	Introduction
6.2	Modeling and Simulation of an Aircraft Electrical Power System (EPS)
6.3	Diagnostic Reasoning for the EPS
6.4	Results and Discussion
6.5	Conclusions

Figure 6-1 Overview of chapter 6

6.1 Introduction

The Electrical Power System (EPS) of an aircraft is generally designed to have many interactions with other aircraft systems. For example, the EPS provides electricity required for various valves in the aircraft like the fuel system, the Environmental Control System (ECS), and the pneumatic systems [1]. It is the conduit source of power for the avionics and provides power for all the instruments in the cockpit. The EPS is provided with the required power by the engines and the Auxiliary Power Unit (APU), which in turn generates electricity and provides for the valves and switches in both engines and the APU. The EPS is required for cabin lighting, as well as for the functioning of electrical and electronic appliances like cabin entertainment systems and sensors [2]. With the aircraft industry moving towards low CO₂ emissions, clean energy, and light weight, more focus is being given to the concepts of fully electric propulsion systems and all-electric aircraft.

As the dependencies on the EPS grow within an aircraft, more importance is given for detecting and isolating faults in the EPS at an early stage. The effect of any fault in the EPS could easily propagate to other aircraft systems that the EPS interacts with. Some faults in the EPS result in common cause failures, affecting multiple aircraft systems at a given time. Electrical failures in the Boeing 747-4H6 in 2012 [3] and the Airbus A319-111 [4] in 2009 are classic examples of common-mode failures. In the case of the B747-4H6 event in 2012, the root cause of the event was a latent hardware fault on the bus tie breaker whereas, in the A319-111 event in 2009, the incident occurred due to an intermittent fault in the Generator Control Unit which led to the loss of the left electrical network. Both these faults resulted in intermittent blanking of cockpit displays, intermittent loss of power to one of the AC bus bars, and resultant degradation of multiple aircraft systems. In the case of B747-4H6, the failure led to the disengagement of all three autopilots and the pilots had to take over the landing manually. In A319-111, the APU generator took over from the failed generator, but many systems, including all means of radio connections, remained inoperative throughout the flight duration. Aircraft incidents like these show the necessity of developing diagnostics for the EPS at the system level. This would not only detect and isolate the faults that affect the EPS but also prevent the possibility of their propagation to the interacting systems. The diagnosis also helps with the research focusing on health monitoring at the aircraft level, in order to isolate faults

propagating to multiple systems [5]. In general, cascading faults result in effects that can be observed in systems other than the point of origin, increasing ambiguity, and resulting in extended downtime and prolonged maintenance as a consequence. Hence diagnostics developed at the systems level, combined with reasoning at the vehicle level, will enable isolation of such faults and their root causes. This will help in reducing the time and money involved in troubleshooting during maintenance [5].

Several research works have been dedicated to developing diagnostic methods for various faults in the aircraft EPS. The Advanced Diagnostic and Prognostic Testbed (ADAPT), an EPS testbed developed by NASA Ames Research Centre, has been used in multiple scenarios to evolve diagnosis and prognosis technologies [6–9]. ADAPT evaluates diagnostic algorithms and software, enables the insertion of faults, and provides a mechanism for evolving the algorithms to be implemented into the vehicles [10]. However, the majority of work found in the EPS diagnostics literature has focused mainly on faults at the components level like wires [11] and relays [8], and the subsystems level like generators [12] and induction motors [13,14]. While most researchers work on the performance of the linked systems like the engine and the fuel pump [1,15,16], there is a lack of investigation in isolating the faults from the EPS, which have propagating effects into other aircraft systems.

In this paper, an ‘Adaptive Neuro-Fuzzy Inference System (ANFIS) monitor’ is developed in MATLAB R2019a to detect the presence of faults in the EPS, within a customizable time period, defined by the user. A hybrid combination of ANFIS monitor with causal reasoning is applied to isolate the root cause of faults found at the Line Replaceable Units (LRUs) connecting the EPS with other aircraft systems. With this objective, in Section 6.2 of this chapter, the digital twin for an aircraft EPS is modeled in MATLAB Simulink R2019a along with its connections to the aircraft engine, the Environmental Control System (ECS), and the fuel system through LRUs as simple as valves and pumps. This is followed by exercising this digital twin under healthy and faulty conditions. Section 6.3 discusses the steps involved in developing the diagnostic reasoning for EPS using ANFIS and presents the ANFIS monitor that is developed for the detectability of faults in the EPS. Section 6.4 presents two approaches to processing the results generated by ANFIS: (i) the crisp boundary approach and (ii) the fuzzy boundary approach. These two methods are compared based on their misclassification

rate (the ratio of the number of misclassified simulation runs to the total number of simulation runs). Causal reasoning is combined with the better method to isolate the faults and their root causes affecting the aircraft system LRUs. Section 6.5 provides the conclusion of this chapter.

6.2 Modelling and Simulation of an Aircraft Electrical Power System (EPS)

In a typical twin-engine aircraft like a Boeing 737, the EPS is powered by two engine-driven generators. The EPS network generates 3-phase, 115/200 V, 400 Hz AC power using either the Constant Speed Drive unit, the Variable Speed Constant Frequency unit, or the Integrated Drive Generator to link with the engines. The EPS network is connected to the engine-driven generators through a series of bus arrangements. The two gen buses are powered by either generator, and they supply power to main loads like hydraulic pumps and windows heating. Electricity is then passed on to the main buses, which supply power to the loads like the fuel boost pumps [17]. The alternating current generated is also stepped down to 28 V AC power for the instrument lighting and converted to 28 V DC power to provide power to components of other aircraft systems like the engine bleed valves or the fuel system shut-off valves. The standby buses for AC and DC are powered either using the AC transfer bus and the DC bus or with batteries via inverters and battery buses, respectively. The AC power sources from the two engine-driven generators do not function in parallel; when an AC power source of a gen bus takes priority, the other existing source is disconnected. The APU driven generator operates identically to the engine-driven generators. It powers the aircraft on ground and serves as a backup for either engine during flight [17,18].

As discussed in the previous paragraph, since only one AC source is active at a given time, this paper considers only the section of the EPS that is powered by one AC source for modelling and the subsequent analysis. To emulate the behaviour of an aircraft electrical power system, the EPS digital twin is developed in MATLAB-Simulink R2019a software, taking references from some of the existing works [19–21]. More details on the modules used to build the EPS digital twin and the mathematics behind them can be found in these references [22–24].

6.2.1 The EPS Digital Twin

The EPS digital twin consists of a constant speed source from the engine shaft that is fed to a generator of 40 kVA capacity. This generator is defined by a simplified synchronous machine in MATLAB Simulink, which produces 115/200 V AC power at 400 Hz. For the given frequency f , the synchronous speed N_s at which the required voltage is produced by the synchronous machine is calculated using the following formula:

$$N_s = f(2/P) * 60 = 400 \text{ Hz} * (120/4) = 12,000 \text{ rpm} \quad (6-1)$$

where P is the number of poles in the generator. With this result, all the simulations of the EPS digital twin will be run at a constant speed of 12,000 rpm [18]. The synchronized machine also takes regulated field voltage as an input, which is fed from the IEEE type 1 voltage regulator combined excitation system. Considering the dissipation losses in the circuit, the total apparent power generated by this synchronous machine is calculated using the following formula from [21]:

$$\text{Apparent Power, } p = 3V_{rms} I_{rms} \text{ (kVA)} \quad (6-2)$$

where rms = root mean square.

As mentioned in Section 6.1, with the growing demand for more electrification of advanced aircraft, the EPS is designed to distribute power to several aircraft systems through its constituent LRUs. For this EPS digital twin, the following LRUs in the aircraft systems are considered for this simulation, with the necessary parameters taken from their respective technical specification sheets provided by their commercial suppliers [25–28]. [Figure 6-2](#) shows the defined MATLAB Simulink model developed for the EPS representative discussed in this paper.

The 115/200 V AC power is provided to the fuel pump AC induction motor. The induction motor is defined by an asynchronous machine rated at 9 KW power and 8000 rpm, with a squirrel cage rotor that results in the rotor speed required to run the fuel pump. The paper refers to the power distributed to this LRU as Power_FSPump; the motor's speed, torque, and stator current are referred to as FS_Motor_Speed, FS_Motor_Torque, and FS_Motor_Stator_I, respectively.

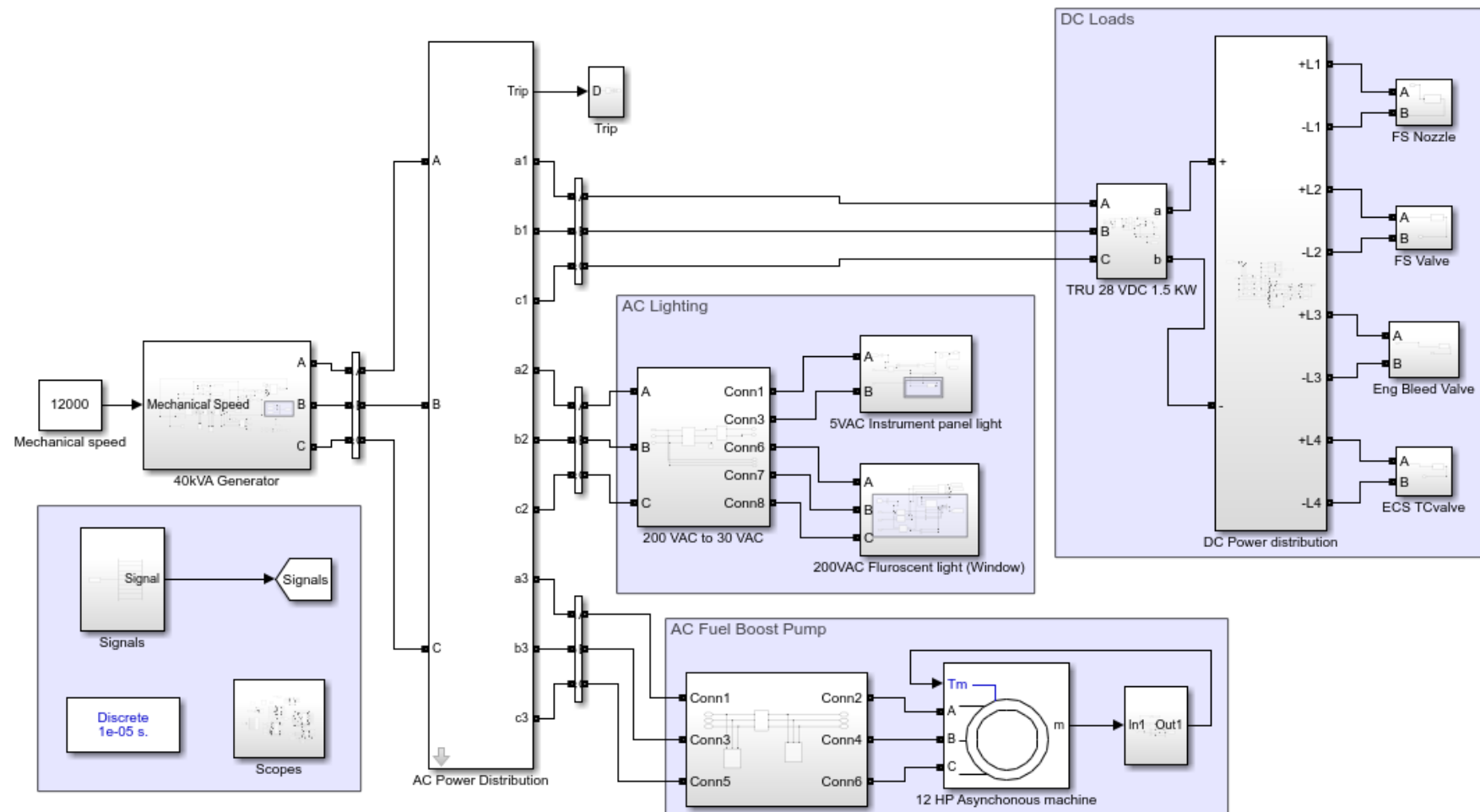


Figure 6-2 MATLAB-Simulink model for the aircraft Electrical Power System (EPS)

The AC power is distributed to the Transformer Rectifier Unit (Power_TRU) and is converted to 28 V DC. This DC power is then supplied to a valve and a nozzle in the fuel system, to a bleed air system high-pressure shut-off valve in the engine, and a temperature control valve in the ECS. This paper refers to these LRUs as FS_Valve, FS_Nozzle, Eng_BleedValve, and ECS_TCVValve, respectively. The AC power distributed to the AC Lamps (Power_ACLamp) is directly provided for the cabin window fluorescent lights (AC_Fluro) and stepped down to 5 V AC for the cockpit instrument panel lights (AC_Instru). The AC power generated is referred to as Power_ACLoad and is distributed at each node representing the aircraft systems and is also calculated using (6-2). Random noise is added in the measurement devices, in order to avoid replicating an ideal condition for the EPS.

Table 6-1 presents the performance of the EPS digital twin at a constant speed of 12,000 rpm, considering a steady state as the operating condition. This simulated condition is considered as the healthy state condition or the baseline for the EPS digital twin. Following the simulation with the healthy scenario, the next section discusses the faults induced in the EPS digital twin.

Table 6-1 Measured values during the Healthy Run

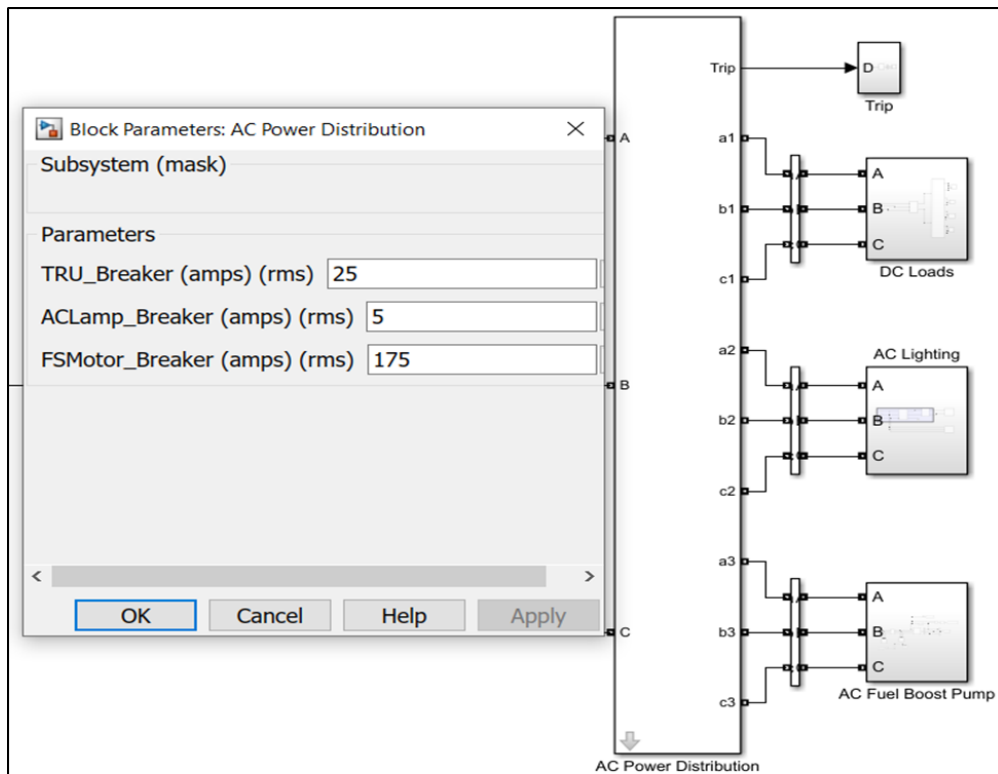
Monitored Parameters	Healthy Measurements	Unit
Power_ACLoad	4182.908	VA
Power_FSPump	4001.362	VA
Power_ACLamp	73.349	VA
Power_TRU	111.102	VA
FS_Motor_Speed	311.813	rad/sec
FS_Motor_Stator_I	4.822	A
FS_Motor_Torque	5.612	Nm
FS_Nozzle_I	0.908	A
FS_Valve_I	0.889	A
Eng_BleedValve_I	0.901	A
ECS_TCVValve_I	0.890	A
AC_Fluro_I	0.259	A
AC_Instru_I	4.892	A

6.2.2 Fault Injection in the EPS Digital Twin

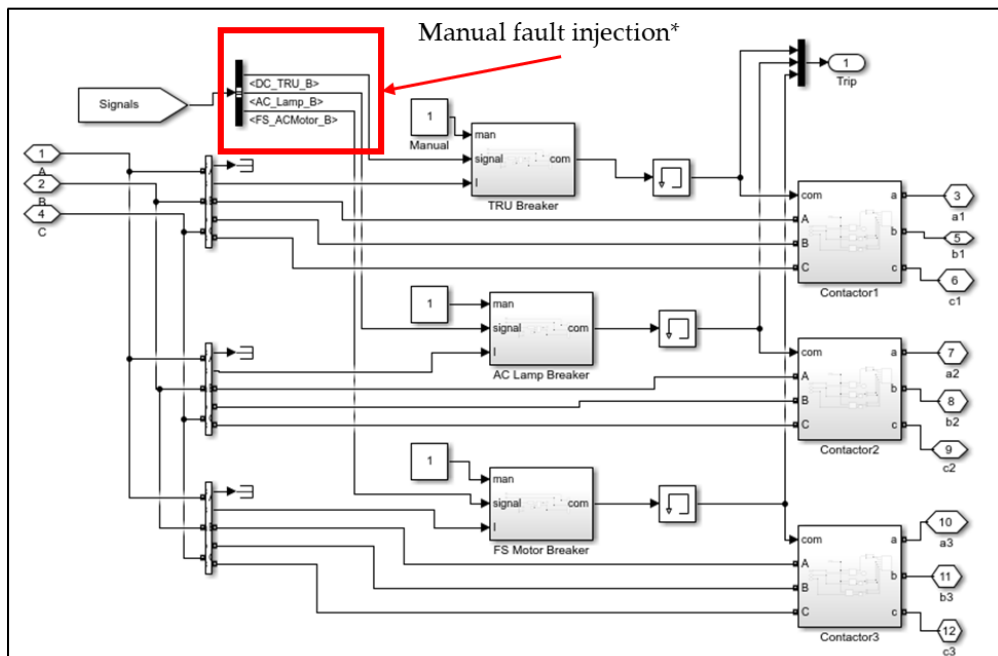
The EPS supplies power to required LRUs of the aircraft systems through a series of switches and circuit breakers that protect the electrical circuits from events like overcurrent surge. These circuit breakers are modelled to trip on their own when the current exceeds the defined limit, as shown in [Figure 6-3a](#). These circuit breakers can also be tripped at the beginning of the simulation, on purpose, by providing a fault signal as input to these components. In [Figure 6-3b](#), DC_TRU_B, AC_Lamp_B, and FS_ACMotor_B are the fault input signals that would induce circuit breaker tripping in the distribution node that supplies AC Power to TRU, AC lamps, and FS motor, respectively. Injecting faults in the circuit breakers will stop the current flow to the respective node that supplies power to the LRU connecting to a particular aircraft system like the engine, the fuel system, or the ECS.

Similarly, faults can be injected via the insulated gate bipolar transistor (IGBT) switches that connect to the valves and nozzles within the DC circuit (as seen in [Figure 6-4](#)), as well as through the switches in AC lamp circuits (not shown in the figure). The faults in the DC circuits from [Figure 6-4](#) emulate the stuck open scenarios by opening the IGBT switches to a partial degree. For testing purposes, FS_valve switch is stuck at 80%, FS_Nozzle switch is stuck at 70%, Eng_BleedValve is stuck at 60%, and the ECS_TCVValve is stuck at 50%. [Figure 6-4](#) also shows the noise signals added to voltage measurements. Similar noise is added to voltage measurements in AC lamp circuits.

In the case of the FSPump AC induction motor, there are several methods to inject faults that would result in the reduction of its rotor speed output. For this paper, the variation of the supply voltage method is chosen [29]. This is carried out by inserting a step-down transformer in the motor circuit, as shown in [Figure 6-5](#). Through this, the supply voltage to the pump motor is reduced to induce the fault.



(a)



(b)

Figure 6-3 Fault injection in circuit breakers connecting generators to other aircraft systems

(a) User input current limits set for Circuit breakers (b) Manual tripping input inside the AC Power distribution mask

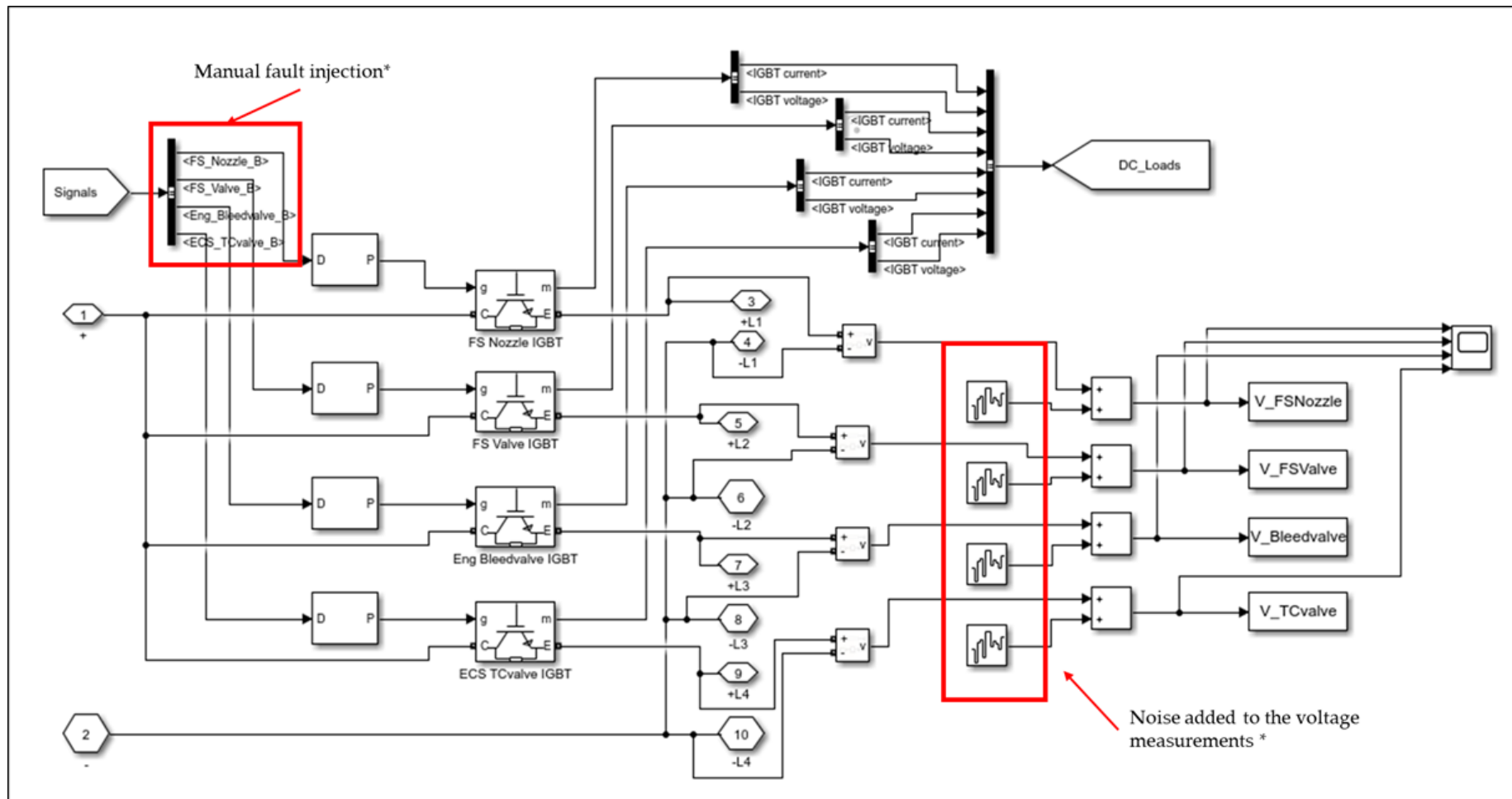


Figure 6-4 Fault injection to DC loads via IGBT switches and noise signal added to voltage measurements

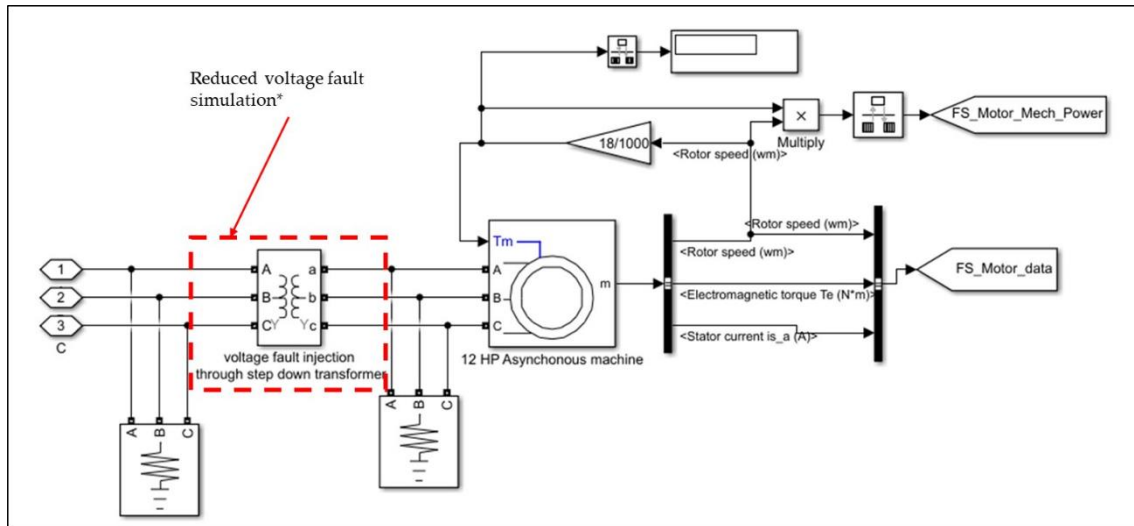


Figure 6-5 Fault injection in AC Pump motor by reduced voltage supply (F10)

Table 6-2 Faults injected in the EPS Digital Twin

Fault_id	Type of fault	Location	Fault Label	Timeline of fault injection
F1	Switch stuck open @ 80%	DC supply to fuel system nozzle	FS_Nozzle_SwitchStuckopen	t=0.25s to t=0.5s
F2	Switch stuck open @70%	DC supply to fuel system shut-off valve	FS_Valve_SwitchStuckopen	t=0.75s to t=1.0s
F3	Switch stuck open @60%	DC supply to engine bleed valve	Eng_BleedValve_SwitchStuckopen	t=1.25s to t=1.5s
F4	Switch stuck open @50%	DC supply to ECS Temperature control valve	ECS_TCVValve_SwitchStuckopen	t=1.75s to t=2.0s
F5	Circuit breaker trip	AC power distribution node to TRU for DC circuits	DC_TRU_B	t=2.5s to t=2.75s
F6	Circuit breaker trip	AC power distribution node to AC fuel pump motor	FS_ACMotor_B	t=3.0s to t=3.25s
F7	Circuit breaker trip	AC power distribution node to AC Lamps	AC_Lamp_B	t=3.5s to t=3.75s
F8	Switch stuck open	AC supply to Instrument panel lamp	AC_Instru_Switchopen	t=4.0s to t=4.25s
F9	Switch stuck open	AC supply to cabin window fluorescent light	AC_fluro_Switchopen	t=4.5s to t=4.75s
F10	Voltage reduction	Line supply voltage to AC fuel pump motor	FS_Motor_lowvoltage	Simulated separately

Considering only abrupt faults, as either intermittent or constant over time, faults chosen for diagnosis in the EPS digital twin are listed in [Table 6-2](#). As discussed above, the faults are chosen in such a way that, when injected, their effects can be observed in the overall power generated by the generator and the overall power consumed by their corresponding LRUs. These faults are also the most common electrical faults explored in the aircraft industry. For demonstration purposes and to avoid a lengthier presentation, all faults listed in Table 2 (F1, F2, ..., F9) are injected for a duration of 0.25 s each, instead of injecting individual faults per simulation run. Fault F10 is simulated separately.

The effects of the faults on the systems and their components and how they affect other systems are observed and compared with the healthy case simulation from [Table 6-1](#). The faults either affect the power supply to the entire system along with their subsystems or only the LRUs based on the location of the fault injection. This can be observed in [Figure 6-6](#) (the fault ids are mentioned in the red box, near their respective effects in the plots). For example, when the fault F5 is injected (circuit breaker connecting the power distribution node to TRU is tripped) at $t = 2.5$ s to $t = 2.75$ s as per [Table 6-2](#), the fault cuts off the current supply to all the valves and nozzles in the DC circuit ([Figure 6-6](#) middle column at $t = 2.5$ s to $t = 2.75$ s). However, when the faults F1, F2, F3, and F4 (scenarios of switch stuck open at the LRUs), are injected at their respective simulation times from [Table 6-2](#), they affect the current consumed by the concerned LRU only. This can be observed from the left column second plot (Power_TRU) and the entire middle column of [Figure 6-5](#). Likewise, when the fault F7 (circuit breaker tripped at the distribution node connecting the AC lamps) is injected at $t = 3.5$ s to $t = 3.75$ s as per [Table 6-2](#), the current is reduced to both the AC instrument panel lamp and at the AC cabin window fluorescent lamp (can be observed in right column bottom two plots of [Figure 6-6](#)). However, the faults F8 and F9 (scenarios of switch stuck open at the LRUs) injected at the simulation times listed in [Table 6-2](#), affect the current consumed by the respective LRUs only. These cases can be observed in the left column bottom plot (Power_ACLamp) and right column bottom two plots of [Figure 6-6](#). In the case of faults in the AC motor, when the circuit breaker is tripped at the AC power distribution node (Fault F6 at $t = 3.0$ s to $t = 3.25$ s), the motor does not receive any power, and hence there is no speed and current (left column third plot (Power_FSPump) and right column top two plots of [Figure 6-6](#)). It is to be noted that, in the top plot on the left column of [Figure](#)

6-6 (Power_total), only the effect of fault F6 is observed. This is due to the difference in magnitude of power consumed by Power_FSPump when compared to the Power_TRU and Power_ACLamp. In the case of fault F10, when the supply voltage is reduced, the rotor speed is lower than the nominal speed that is observed during the healthy case simulation. The reduced supply voltage scenario is shown in Figure 6-7. Thus, all the faults listed in Table 6-2 affect the overall power output from the simulation of EPS digital twin, along with the power consumption of the aircraft systems the EPS is connected to.

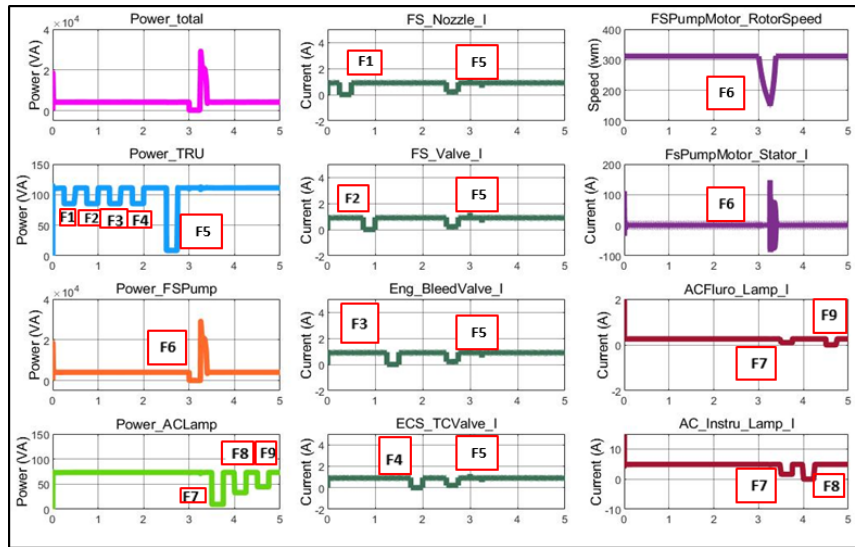


Figure 6-6 EPS performance during various fault scenarios (faults F1, F2...F9).

Left column: Total power generated and power consumed by DC TRU, FS Pump, and AC Lamp (kVA). Middle column: Currents at DC Loads. Right column, top two plots: FS Pump characteristics. Right column, bottom two plots: Currents at AC Lamps

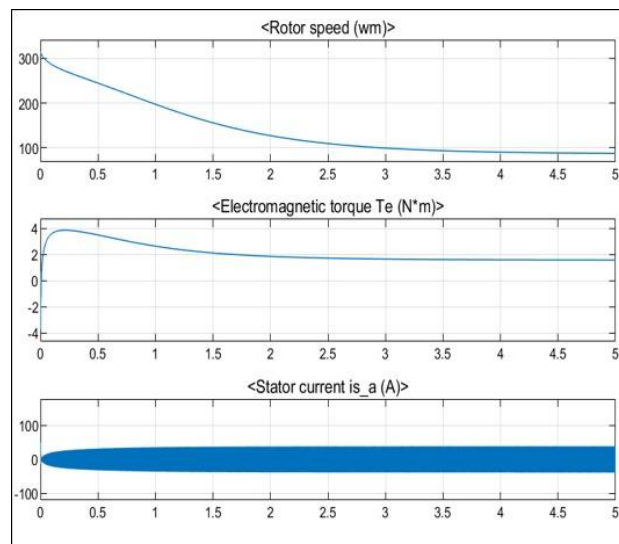


Figure 6-7 Pump motor characteristics for reduced voltage (F10)

6.3 Diagnostic Reasoning for the EPS

The simulations run for both healthy and faulty conditions in the EPS digital twin show that faults induced in the EPS affect the LRUs of other systems, thus propagating their effects to these interacting systems. In order to detect and isolate the faults propagating to other aircraft systems from the EPS, a diagnostic reasoning engine is developed through a series of steps, as discussed below.

- i. Data is collected by running the simulation at both healthy and all known fault scenarios. A set of parameters was chosen to monitor the health of the EPS digital twin during these simulations, and the raw data was collected for these parameters through sampling.
- ii. Preprocessing of the sampled raw data is undertaken by calculating the root mean square (rms) values of the monitored parameters followed by computing the mean values of the steady-state data.
- iii. Factor analysis is carried out using the computed mean values to understand the effects of injected faults on the monitored parameters. This step helps in shortlisting the factors that need to be considered for building the diagnostic reasoning engine and ignoring the redundant parameters.
- iv. Thresholds are determined to define the diagnostic rules for the reasoning engine. If crisp thresholds are established between the parameters for all the simulated scenarios, these threshold values can be used to define the diagnostic rules for the reasoning engine. However, from [Figure 6-6](#), it is observed that for certain faults like F7 or F1, the apparent power does not vary much from the 100% healthy case results. In such scenarios, defining a crisp boundary of the parameter in order to differentiate a healthy case from the fault cases and to differentiate a fault case from another, would require a large number of very specific rules. To avoid this, the thresholds will hence be defined by fuzzy boundaries, and diagnostic rules will be framed using fuzzy reasoning.
- v. This step involves developing fuzzy diagnostic reasoning. In order to avoid specificity in the threshold values, the membership functions of the fuzzy system are determined using artificial neural networks. The following subsections present the fundamentals of ANFIS and how it is implemented for the EPS diagnostics.

6.3.1 Brief Description of ANFIS

The Adaptive Neuro-Fuzzy Inference System (ANFIS) is the combination of artificial neural networks (ANN) with Takagi–Sugeno fuzzy reasoning [30]. A general fuzzy inference system (FIS) operates by the following steps:

- i. Input parameters are fuzzified by defining membership functions,
- ii. Logical operators like AND, OR, and NOT are applied to combine the membership functions of the input parameters to get a set of ‘if-then’ rules along with their weights,
- iii. The consequent parts of the rules are implemented depending upon their firing strengths (weights), and
- iv. The outputs are calculated, aggregated, and de-fuzzified to provide the final output.

A generalized example to demonstrate the above-mentioned steps is shown in [Figure 6-8](#). Consider a first-order Sugeno FIS for given two-parameter inputs A and B presented in the following sets of equations from [30], as seen in [Figure 6-8a](#).

$$\begin{aligned} \text{Rule 1: If A is X1 and B is Y1,} \\ \text{then } z1 = p1*A + q1*B + r1 \end{aligned} \tag{6-3}$$

$$\begin{aligned} \text{Rule 2: If A is X2 and B is Y2,} \\ \text{then } z2 = p2*A + q2*B + r2 \end{aligned} \tag{6-4}$$

where $p1, p2, q1, q2, r1$, and $r2$ are linear parameters, and $X1, X2, Y1$, and $Y2$ are non-linear parameters. The inputs of A and B are fuzzified by assigning membership functions, say μ and ν . These membership functions represent the degree to which A and B satisfy pertaining to their fuzzy labels $X1, Y1, X2$, and $Y2$. Logical operator AND⁴ is applied to establish the relationship between the A and B using their membership functions. For example, consider the weights that could be calculated by the following equations:

$$w1 = \text{AND} (\mu X1 (A), \nu Y1 (B)) \tag{6-5}$$

⁴ In Fuzzy logic And $(x,y) = \min (x,y)$

$$w2 = \text{AND} (\mu_{X2} (A), \nu_{Y2} (B)) \quad (6-6)$$

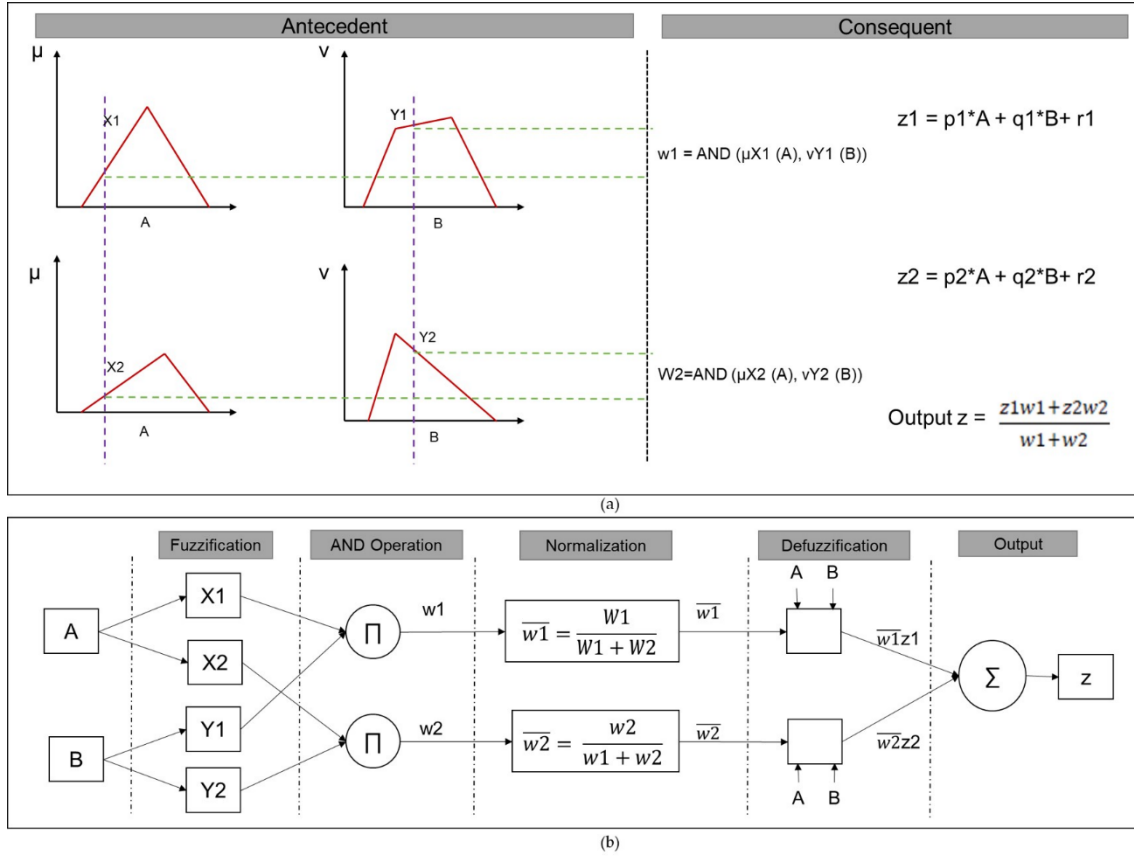


Figure 6-8 (a) First order Takagi-Sugeno Fuzzy Reasoning (b) ANFIS structure (adapted from [30])

Once the weights of the fired rules, w_1 and w_2 , are computed, they are combined with the outputs z_1 and z_2 . The total output ‘ z ’ is given by calculating the weighted average of the outputs, as given in (6-7) [30]. While A and B are fuzzy sets in the antecedent, ‘ z ’ is a crisp function in the consequent side.

$$\text{Output } z = (z_1 * w_1 + z_2 * w_2) / (w_1 + w_2) \quad (6-7)$$

The ANN is used to tune the membership functions in the Takagi–Sugeno FIS, in order to optimize the relationship between the input parameters and their respective fuzzy labels, resulting in the algorithm, ANFIS (as seen in Figure 6-8b). The neural network in ANFIS estimates errors and adjusts the membership function parameters to reduce the errors. The parameters are estimated either by using backpropagation or by a hybrid method that combines least squares estimation with backpropagation.

6.3.2 An ANFIS Monitor for the EPS

To detect and isolate faults in the EPS, an ANFIS function is developed using the Neuro-Fuzzy designer toolbox in MATLAB R2019a [31,32]. Using all the monitored parameters to develop ANFIS is impractical, as it consumes a lot of memory and ends up developing redundant rules. For example, from [Figure 6-2](#), it is seen that power can be calculated and monitored from 11 nodes (one node from the generator, three nodes distributing power, and power supplied to four DC nodes, two AC lamp nodes, and one AC pump node). Considering these 11 nodes, and AC motor rotor speed as the 12th monitored parameter, if every parameter is defined with three membership functions, the number of rules developed to map the relationship between the membership functions will be:

$$\text{No. of rules} = 12 * 11 * (3^2) = 1188 \quad (6-8)$$

In reality, the EPS interacts with many aircraft systems through the LRUs. Monitoring power distributed at all these nodes will not be efficient for the diagnostic process as it will consume a lot of memory and time, and the simulation might not even converge in certain scenarios. Hence instead of using all monitored parameters, only a few should be selected for further analysis to detect the presence of a fault in the EPS. In this paper, the Power_Generated parameter is chosen for the purpose of diagnosis. This is due to the observation made from [Table 6-1](#), where the total power generated in row 1 is almost equal to the total power consumed in rows 2–4, the slight difference accounting for the losses and the noise measurements. This eliminates the need to monitor both the power distributing and consuming nodes. Any presence of fault could ideally be detected by monitoring only the power generated from the EPS. A Gaussian membership function with three categories (low, medium, high) is defined for the monitored parameter, Power_Generated, using the Neuro-Fuzzy designer toolbox. Hence, the number of rules generated in this case becomes three. The ANFIS function is implemented as a monitor that screens the parameter Power_Generated, for a defined window of time. This defined time period is programmed to be any number of windows according to the user's interest, thus making the ANFIS monitor customizable.

6.4 Results and Discussion

Simulations are run for healthy and various faulty scenarios and the processed steady-state data from Power_Generated. The total data processed for analysis is split into training (40% of the data) and testing (60% of the data). In general, ANFIS generates the output ‘z’ in numbers between 0–5 with decimal points (0 being the lowest power generated and 5 being the highest). These values of ‘z’ are processed in two different methods for further isolation of faults.

6.4.1 Method 1: The Crisp Boundary Approach

The output values, ‘z’, with decimals generated by ANFIS, are rounded up to discrete ranks, providing crisp boundaries between the regions. This results in six discrete values; thus, six distinct regions, from Region 5 to Region 0, as shown in Figure 6-9. Any surge in power generated, due to an unsteady state is classified as ‘undefined’. ANFIS is trained with the training data, and its ability to estimate the output based on the power generated is evaluated using the testing data, where it produced a root mean square error of 0.31.

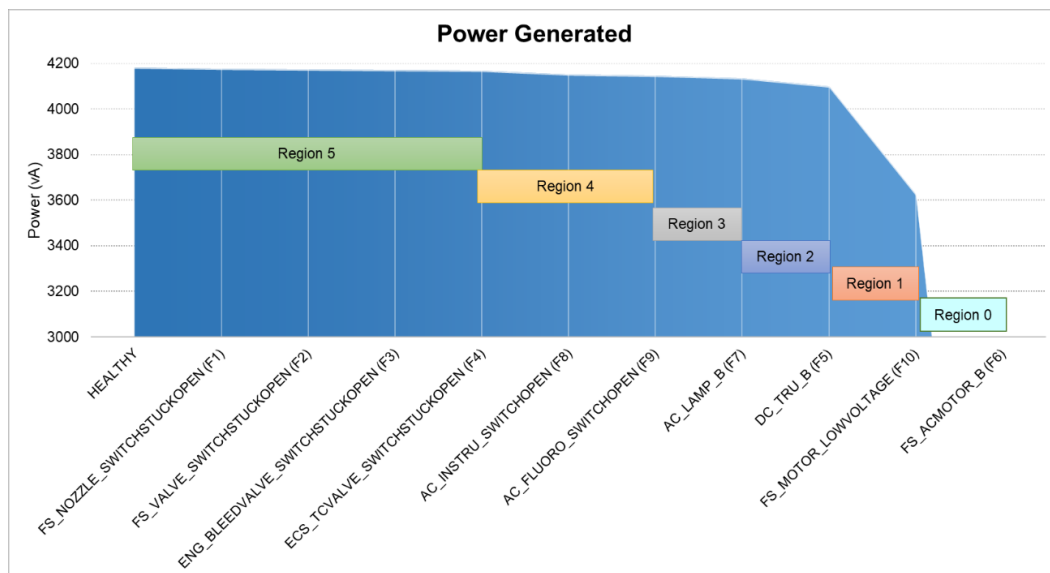


Figure 6-9 Healthy and faulty scenarios classified based on apparent power generated using Crisp boundary approach

The trained ANFIS function is programmed to detect any presence of faults in the EPS after every simulation run. For demonstration purposes, the ANFIS monitor screens the steady-state simulation run time every 0.25 s to check the state of total power generated by the EPS and in this paper, two cases are tested.

6.4.1.1 Healthy Case

When the EPS is run for a healthy scenario (Figure 6-10a), the result generated by the ANFIS monitor is as shown in Figure 6-10b. The ANFIS monitor checks the simulation run every 0.25 s, and the results show ‘Region 5’ for every time window.

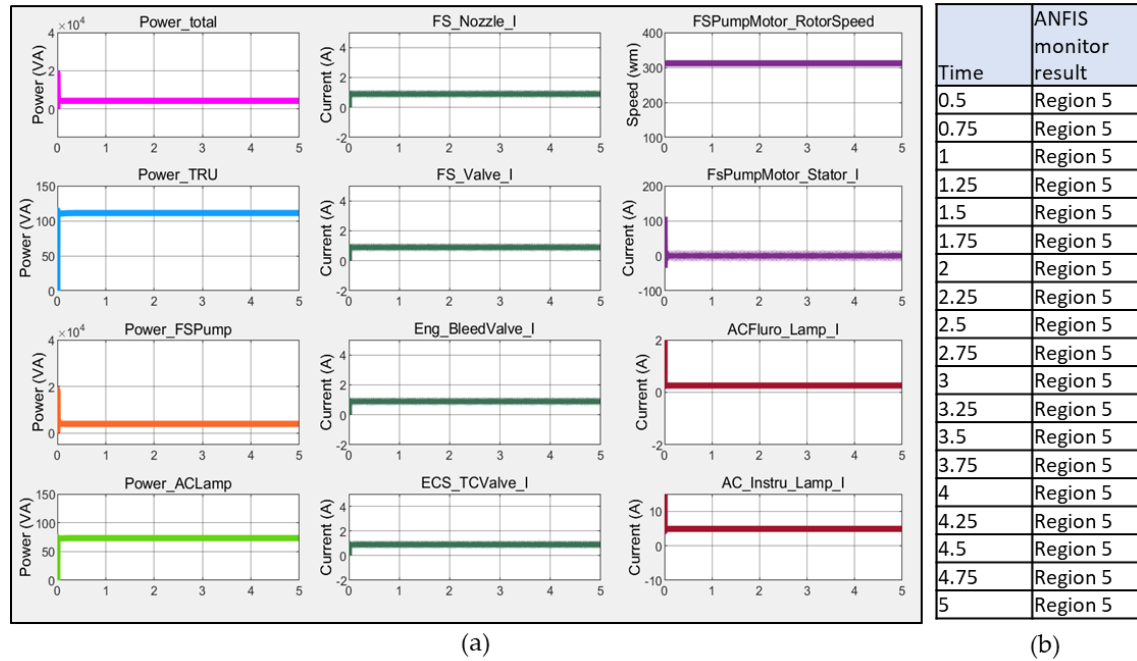


Figure 6-10 Healthy scenario using the Crisp boundary approach

(a) EPS performance in terms of Apparent power (left column), DC currents (middle column), Pump performance (top two plots in the right column), AC Lamps currents (bottom two plots in the right column) (b) The corresponding results generated by ANFIS monitor using Method 1 approach.

6.4.1.2 Faulty Case

In an ideal testing scenario, only a single fault scenario is tested. However, for the purposes of demonstration, the same faults from Table 6-2 (for 0.25 s each) are tested in this case and shown in Figure 6-11. It can be observed in Figure 6-11b that the ANFIS monitor has not classified the regions correctly. For example, fault F7 ($t = 3.5$ s to $t = 3.75$ s) is classified as ‘Region 4’ at $t = 3.75$ s instead of ‘Region 3’ and fault F5 ($t = 2.5$ s to $t = 2.75$ s) is classified as ‘Region 3’ at $t = 2.75$ s instead of ‘Region 2’, as assigned in Figure 6-11b.

This inaccuracy in the detection of regions is due to the error in rounding up the values of ‘z’ while assigning the regions to each fault. The misclassification rate is high, at 67%, as seen in Figure 6-12, due to the inaccurate region assignments using crisp boundaries.

Applying any additional approach like causal reasoning to this method to distinguish the faults from each region will not be beneficial since the regions are not classified appropriately. In order to reduce the effect of this misclassification problem and to narrow down the fault to the concerned LRU, a fuzzy boundary approach is followed along with causal reasoning in Method 2.

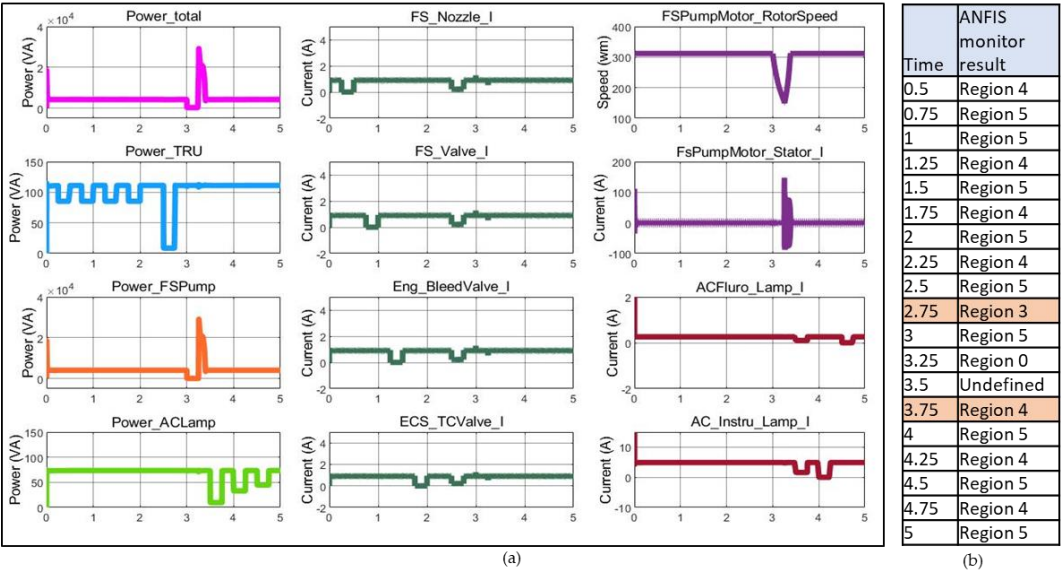


Figure 6-11 Faulty scenario using the Crisp boundary approach

(a) EPS performance in faulty scenarios (b) The corresponding ANFIS monitor results

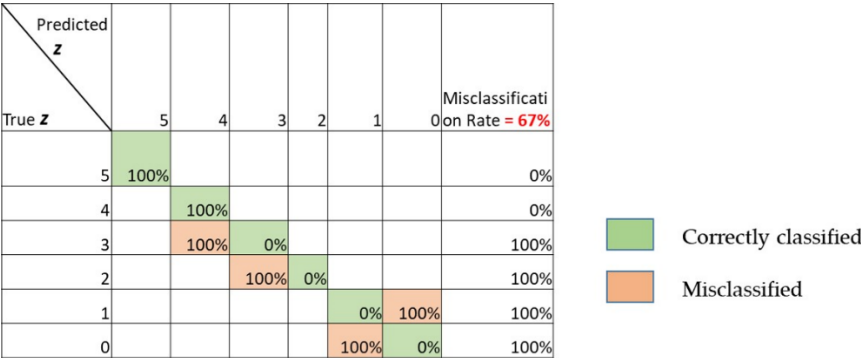


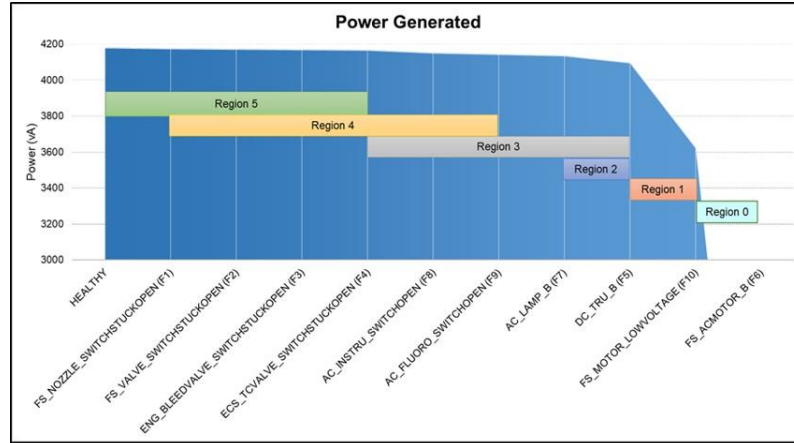
Figure 6-12 Confusion matrix for the Crisp boundary approach

6.4.2 Method 2: The Fuzzy Boundary Approach

In method 2, the ‘fuzzy boundary’ concept is used since the crisp boundaries for processing the ANFIS output values ‘z’ did not work, as shown in Method 1. Regions 5 to 0 are now representing multiple faults overlapping other regions (Figure 6-13) instead

of representing only the ‘defined’ faults without any overlaps as in method 1. For example, from [Figure 6-13a](#), it is seen that the fault F5 comes under both ‘Region 2’ and ‘Region 3’. This method provides an accurate classification of faults into their regions, as shown in the confusion matrix in [Figure 6-13b](#). In order to isolate the faults that propagate to other aircraft systems, the second method uses a hybrid combination of causal reasoning with ANFIS. Causal reasoning is a reasoning strategy that draws conclusions based on the cause and effect relationship of the concerned subjects [33]. In general, developing causal reasoning for a system such as EPS is a time-consuming and tedious task due to the large number of interactions between the EPS and other systems. However, with the combination of the ANFIS monitor similar to that in method 1, their results can be used to narrow down the region of search for cause and effect relationships. The algorithm for implementing the method 2 approach is provided in Appendix A.

Using the ANFIS monitor, the ‘effect’ criteria for a particular region of faults is now narrowed down, which is followed by carrying out a certain number of tests to find out the exact ‘cause’ of the observed effect. This will be done by developing the diagnostic matrix that maps the required tests to be carried out for each region, and once the criteria of the tests are satisfied, the result will lead to the exact fault isolation. The diagnostic matrix for isolating the faults in the EPS is shown in [Table 6-3](#). For every region identified by the ANFIS monitor, the voltages of the concerned LRUs are checked to isolate the faulty LRU. Carrying out every test in the table during every simulation run will consume a lot of time and memory, especially if the task is only to detect the presence of any fault. Thus, the application of the ANFIS monitor to detect the presence of a fault, followed by diagnosis using the diagnostic matrix, has the significant advantage of isolating the fault in the EPS in a shorter duration, with lesser memory.



(a)

Predicted Z	5	4	3	2	1	0	Misclassification Rate = 0%
True Z							
5	100%						0%
4		100%					0%
3			100%				0%
2				100%			0%
1					100%		0%
0						100%	0%

(b)

Figure 6-13 (a) Fuzzy boundaries representing Healthy and Faulty scenarios based on the Power generated, (b) Confusion matrix for Method 2 ANFIS rank processing with Fuzzy boundaries

Table 6-3 Diagnostic matrix for faults vs. tests

Region	Tests	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
5	All DCValves_V = V_Nominal	-	-	-	-	-	-	-	-	-	-
	Any DCValve_V < V_Nominal	X	X	X	X	-	-	-	-	-	-
4	Any DCValve_V < V_Nominal	X	X	X	X	-	-	-	-	-	-
	Any ACLamp_V < V_Nominal	-	-	-	-	-	-	-	X	X	-
3	Any ACLamp_V < V_Nominal	-	-	-	-	-	-	-	X	X	-
	All ACLamps_V < V_Nominal	-	-	-	-	-	-	X	-	-	-
	All DCValves_V < V_Nominal	-	-	-	-	X	-	-	-	-	-
2	All DCValves_V < V_Nominal	-	-	-	-	X	-	-	-	-	-
1	AC_Motor_Speed < Speed_Nominal && AC_Motor_Torque > AC_Motor_Torque_min	-	-	-	-	-	-	-	-	-	X
0	AC_Motor_Speed < Speed_Nominal && AC_Motor_Torque < AC_Motor_Torque_min	-	-	-	-	-	X	-	-	-	-

An example of the end result, isolating a fault in the EPS, using the ANFIS monitor and the diagnostic matrix, is shown in Figure 6-14. The variable 'monitor_res' in Figure 6-14

shows the summarized output from the ANFIS monitor, which is then followed by the tests from the diagnostic matrix to isolate the F6 fault. Figure 6-15 shows the isolation of fault F5. Similarly, with the help of the diagnostic matrix, the tests are able to isolate the LRUs that are propagating faults to the other systems. While the ANFIS rank processing with the fuzzy boundaries classifies the regions of faults accurately, the overall classification accuracy of method 2 is 99.1%. This is due to the slight overlaps between the healthy and DC Valves nominal voltages that are tested in Region 5. The tests carried out from the diagnostic matrix in Table 6-3 are able to isolate the faults accurately in the rest of the regions.

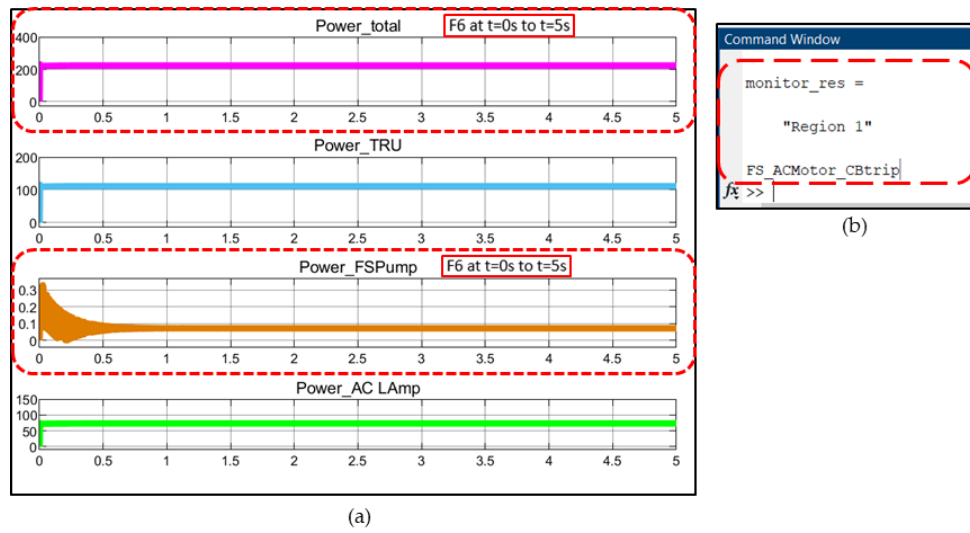


Figure 6-14 (a) Corresponding reaction in apparent power for ACMotor_Fault Injection (F6) (b) Method 2 results with the ANFIS monitor and Causal reasoning

The diagnostic algorithm is tuned in to detect only the steady-state, single fault criteria, and hence, in any case of multiple or intermittent faults, the variable ‘monitor_res’ will give the result as ‘Intermittent,’ and the further diagnostic tests will be cancelled. This can be seen in Figure 6-16a, where the fault F6 is injected between $t = 2.5$ s to $t = 3.0$ s, the ‘monitor_res’ variable is ‘Intermittent,’ and there are no further diagnostic results. Similarly, in Figure 6-16b, multiple faults are injected for 0.5 s duration, (F5 between $t = 2.5$ s to 3.0 s, and F8 between $t = 1.5$ s to $t = 2.0$ s) and the end result is shown as ‘Intermittent’. Thus, the hybrid combination of the ANFIS monitor with causal reasoning is capable of detecting and diagnosing both steady and unsteady state fault scenarios in the EPS, and it is also tunable as required.

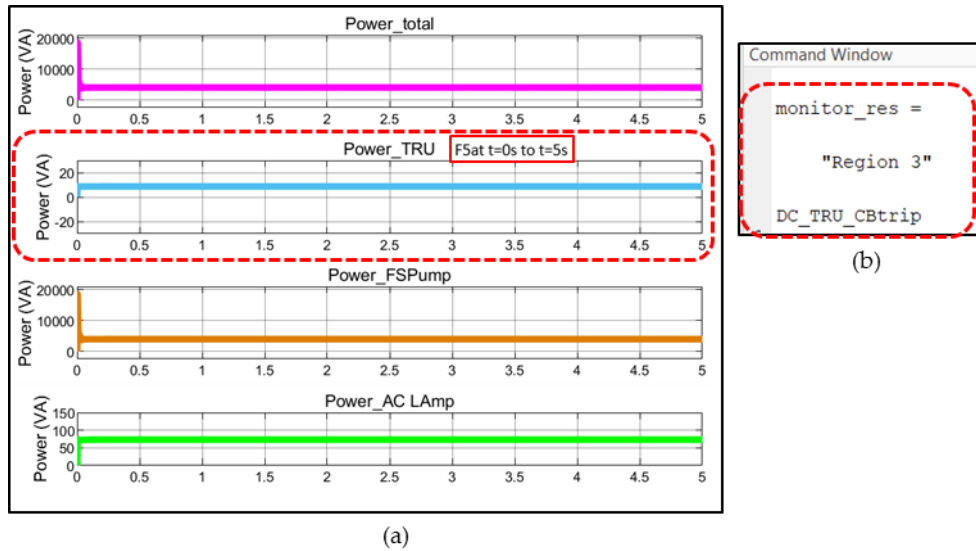


Figure 6-15 (a) Corresponding reaction in apparent power for DC_TRU_CBtrip (F5) fault injection (b) Method 2 results with the ANFIS monitor and Causal reasoning

6.5 Conclusions

This paper discussed the importance of developing diagnostics for an Electrical Power System to isolate faults, including those that affect other systems like the engine, the fuel system, and the Environmental Control System of an aircraft. The development of an EPS digital twin, along with its simulation of healthy and faulty states, are presented. An ANFIS monitor is developed to detect the presence of faults in the EPS based on the power generated. Isolation of faults to particular LRUs is attempted by processing the ANFIS ranks in two methods: i) a crisp boundary approach, and ii) a fuzzy boundary approach. While the former provided a very high misclassification rate of 67%, the second method was able to classify the faults to their regions accurately. This fuzzy boundary method is combined with causal reasoning by developing tests for every region through a diagnostic matrix, through which the faults that propagate to LRUs of other aircraft systems are isolated. The developed diagnostic algorithm with ANFIS is tunable to

monitor the power generated at desired time windows and is also capable of identifying the intermittency during the simulation run.

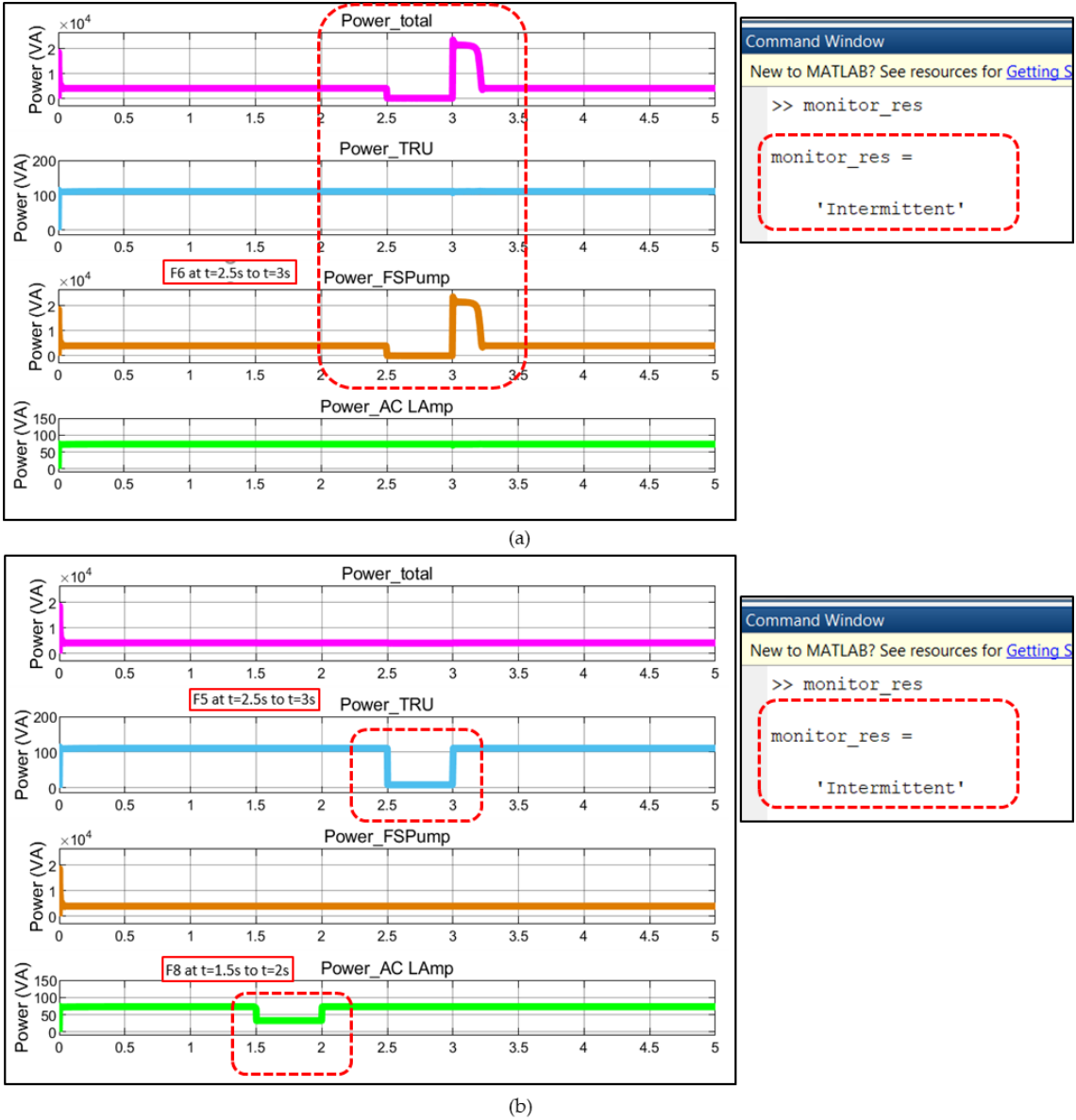


Figure 6-16 (a) Diagnostics for one intermittent fault (F6) (b) Diagnostics of multiple intermittent faults (F5, F8)

With these results from EPS diagnostics, it is possible to pinpoint the location and time where a fault has occurred, thus reducing the time taken to troubleshoot the root cause of faults effectively. This method will help in preventing ambiguous scenarios like the aircraft incidents presented in Section 6.1, where the effect is significantly observed

in the interacting system rather than the system of origin. As the combination of the ANFIS monitor with causal reasoning does not take much time to isolate the faults during the simulation run, this diagnostic method could be tried in both ground systems as well as onboard highly electrified aircraft for rapid diagnosis in the case of real-time fault scenarios. As this diagnostic method is not heavy on computation requirements, it is also helpful for monitoring the health of the aircraft at the vehicle level, considering the interaction between its multiple systems. A framework of digital twins of multiple interacting systems like the EPS, the engine, the ECS, and the fuel system, along with their system-level diagnostics isolating the interacting faults, would help in monitoring the overall health of the aircraft [5].

References

1. Zumberge J., Wolff J., McCarthy K., O'Connell T., Walters E., Russell G., et al. Integrated aircraft electrical power system modeling and simulation analysis. SAE Technical Papers. 2010; Available at: DOI:10.4271/2010-01-1804
2. Rosero JA., Ortega JA., Aldabas E., Romeral L. Moving towards a more electric aircraft. IEEE Aerospace and Electronic Systems Magazine. March 2007; 22(3): 3–9. Available at: DOI:10.1109/MAES.2007.340500
3. Air Accidents Investigation Branch. AAIB Bulletin: 4/2014 9M-MPL EW/C2012/08/04. 2014. Available at: <https://www.gov.uk/aaib-reports/boeing-747-4h6-9m-mpl-17-august-2012>
4. Air Accidents Investigation Branch. Aircraft Accident Report No: 4/2009. Available at: https://assets.publishing.service.gov.uk/media/552659a7e5274a141800000d/Summary_AAR_4-2009_Airbus_A319-111__G-EZAC_09-09.pdf
5. Ezhilarasu CM., Skaf Z., Jennions IK. Progress towards a Framework for Aerospace Vehicle Reasoning (FAVER). Proceedings of the Annual Conference of the PHM Society. 2019; 11(1): 1–9. Available at: DOI:<https://doi.org/10.36001/phmconf.2019.v11i1.887>

6. Sweet A. Testing HyDE on ADAPT. NASA STI Technical Memorandum. 2008. Available at: <https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20080017209.pdf>
7. Knox WB., Mengshoel OJ. Diagnosis and Reconfiguration using Bayesian Networks: An Electrical Power System Case Study. Proc. of the IJCAI-09 Workshop on Self-\star and Autonomous Systems (SAS): Reasoning and Integration Challenges. 2009. pp. 67–74.
8. B.W. R., O.J. M. Methods for probabilistic fault diagnosis: An electrical power system case study. Annual Conference of the Prognostics and Health Management Society, PHM 2009. 2009; : 1–12.
9. McIntire MG., Keshavarzi E., Tumer IY., Hoyle C. FUNCTIONAL MODELS WITH INHERENT BEHAVIOR: TOWARDS A FRAMEWORK FOR SAFETY ANALYSIS EARLY IN THE DESIGN OF COMPLEX SYSTEMS. ASME 2016 International Mechanical Engineering Congress and Exposition. Phoenix, Arizona, USA: ASME; 2016. pp. 1–8. Available at: DOI:10.1115/IMECE2016-67040
10. Mengshoel OJ., Darwiche A., Cascio K., Chavira M., Angeles L., Poll S., et al. Diagnosing Faults in Electrical Power Systems of Spacecraft and Aircraft. Artificial Intelligence. 2008; (650): 1699–1705. Available at: DOI:10.1.1.149.7338
11. Cao Y., Liu G., Liu Z. Condition Based Maintenance for Aircraft Electrical Systems. SAE Technical Papers. 2009; Available at: DOI:10.4271/2009-01-3163
12. Batzel TD., Swanson DC. Prognostic health management of aircraft power generators. IEEE Transactions on Aerospace and Electronic Systems. IEEE; 2009; 45(2): 473–483. Available at: DOI:10.1109/TAES.2009.5089535
13. Faiz J., Ghorbanian V., Joksimovic G. Fault Diagnosis of Induction Motors. 1st Annual International Interdisciplinary Conference, AIIC. Azores, Portugal: Institution of Engineering and Technology; 2017. Available at: DOI:10.1049/PBPO108E

14. Sekar BD., Dong MC. Bayesian fuzzy inference nets online fault diagnosis of induction motor. ICICS 2009 - Conference Proceedings of the 7th International Conference on Information, Communications and Signal Processing. IEEE; 2009; (007): 1–5. Available at: DOI:10.1109/ICICS.2009.5397664
15. Seok J., Kolmanovsky I., Girard A. Integrated/coordinated control of aircraft gas turbine engine and electrical power system: Towards large electrical load handling. 2016 IEEE 55th Conference on Decision and Control (CDC). IEEE; 2016. pp. 3183–3189. Available at: DOI:10.1109/CDC.2016.7798747
16. Shi X., Qu J., Wu H., Cai J., Yang Z. Modelling and simulation of the logical structure of aircraft power distribution systems. International Journal of Simulation: Systems, Science and Technology. 2016; 17(27): 37.1-37.9. Available at: DOI:10.5013/IJSSST.a.17.27.37
17. Boeing. THE BOEING 737 TECHNICAL SITE. Available at: <http://www.b737.org.uk/electrics.htm> (Accessed: 3 March 2020)
18. Madonna V., Member S., Giangrande P., Galea M. Electrical Power Generation in Aircraft : review , challenges and opportunities. Research Gate; 2018. Available at: DOI:10.1109/TTE.2018.2834142
19. Olivier Tremblay L-AD. Aircraft Electrical Power Generation and Distribution. MathWorks. Available at: <https://www.mathworks.com/help/phymod/sps/examples/aircraft-electrical-power-generation-and-distribution.html> (Accessed: 23 February 2020)
20. Steve Miller. Aircraft Power Network in Simscape Electrical. MathWorks. 2018. Available at: <https://www.mathworks.com/matlabcentral/fileexchange/44124-aircraft-power-network-in-simscape-electrical/> (Accessed: 23 February 2020)
21. Mushiri T., Mbohwa C. Research on the use of matlab in the modeling of 3-phase power systems. Proccedings of the World Congress on Engineering. 2015. pp. 363–369.
22. MathWorks. Asynchronous Machine. Available at:

- <https://www.mathworks.com/help/physmod/sps/powersys/ref/asynchronousmachine.html> (Accessed: 23 February 2020)
23. MathWorks. Simplified Synchronous Machine. Available at: <https://www.mathworks.com/help/physmod/sps/powersys/ref/simplifiedsynchronousmachine.html> (Accessed: 23 February 2020)
 24. MathWorks. Excitation Systems. Available at: <https://www.mathworks.com/help/physmod/sps/powersys/ref/excitationsystem.html> (Accessed: 23 February 2020)
 25. OECO. Transformer rectifiers. Available at: <https://www.oeco.com/wp-content/uploads/2015/07/transformer-rectifier-units.pdf>
 26. Eaton. Fuel Boost Pump Type 20004 and Canister Type 20005. Available at: <https://www.eaton.com/Eaton/ProductsServices/Aerospace/Fuel/FuelBoostPumps/index.htm>
 27. Aerodynamix Inc. Aircraft Instrument Panel Lighting. Available at: <https://aerodynamix.com/aircraft-instrument-panel-lighting/> (Accessed: 3 March 2020)
 28. Valcor Aerospace. 2-Way Normally Open or Normally Closed Shut-Off Solenoid Valve. 2015. Available at: <https://www.valcor.com/v4700-2-way-shut-solenoid-valve/>
 29. Diyoke GC., C. O., Aniagwu U. Different Methods of Speed Control of Three-Phase Asynchronous Motor. American Journal of Electrical and Electronic Engineering, Vol. 4, 2016, Pages 62-68. 2016; 4(2): 62–68. Available at: DOI:10.12691/AJEEE-4-2-3
 30. Jang JSR. ANFIS : Adaptive-Network-Based Fuzzy Inference System. IEEE Transactions on Systems. IEEE Transactions on Systems, Man and Cybernetics. 1993; 23(3): 665–685. Available at: DOI:10.1109/21.256541
 31. MathWorks. Fuzzy Logic Toolbox. Available at:

<https://www.mathworks.com/help/fuzzy/> (Accessed: 23 February 2020)

32. Bystrov D., Westin J. Neuro-Fuzzy Logic Systems. MathWorks. 2015. pp. 69–76.
Available at: <https://users.du.se/~jwe/fuzzy/NFL/F10.PDF>
33. Ezhilarasu CM., Skaf Z., Jennions IK. The application of reasoning to aerospace Integrated Vehicle Health Management (IVHM): Challenges and opportunities. Progress in Aerospace Sciences. Elsevier Ltd; 2019; 105(January): 60–73.
Available at: DOI:10.1016/j.paerosci.2019.01.001

7 A Generalized Methodology for the Diagnosis of Aircraft Systems

An aircraft is made up of a number of complicated systems which work in harmony to ensure safe and trouble-free flight. In order to maintain such a platform, many diagnostic and prognostic techniques have been suggested, mostly aimed at components but some at the system level. Together these form a patchwork approach to the overall problem of efficiently informing aircraft maintenance to the OEMs (Original Equipment Manufacturers), the operators/airlines, and the MRO (Maintenance, Repair and Overhaul) organizations. It involves these organizations having to support several different approaches to aircraft health management and is therefore, inefficient and costly. In the current work, a streamlined methodology is put forward. This is based on OSA-CBM (Open System Architecture for Condition Based Maintenance), and can be applied to any aircraft system. Integral with this is the use of mRMR (minimum redundancy maximum relevance) for feature selection, the resulting symptom vector being used for fault diagnosis. This approach is demonstrated on three test cases: the engine, the environmental control system (ECS), and the fuel system. In each case the digital twin setup, simulation conditions for healthy and faulty scenarios, the methodology based on OSA-CBM, and diagnostics are detailed. Diagnostics is carried out for each system in turn, using four machine learning supervised algorithms. The best performing algorithm for each system will then subsequently be used in a vehicle level reasoner called FAVER (A Framework for Aerospace Vehicle Reasoning), which requires these system diagnoses as a starting point for vehicle reasoning and fault ambiguity resolution.

7.1	Introduction
7.2	The OSA-CBM Methodology
7.3	Use cases
7.4	Diagnosis
7.5	Summary and Future work

Figure 7-1 Overview of Chapter 7

7.1 Introduction

An aircraft is a complex machine made up of multiple systems, as detailed by the FAA [1]. These systems are usually built in isolation and assembled together on a common platform. Every aircraft system, such as the engine, the environmental control system (ECS), the fuel system (FS), the electrical power system (EPS), the pneumatic system, or the auxiliary power unit (APU) has functions of its own, and objectives to satisfy at different levels [2]. These objectives may range from the component level (e.g., the blade of a compressor in an engine), the line replaceable unit (LRU) level (e.g., gearbox assembly in an engine), the subsystem level (e.g., the engine compressor), the system level (e.g., the engine) or the overall vehicle level (the aircraft). One of the most important objectives of these aircraft systems, at any level, is to perform safely at their optimal health and efficiency, contributing to the best performance of the aircraft, as well as saving the cost and time involved in the maintenance and unexpected downtime. In order to achieve this objective, Integrated Vehicle Health Management (IVHM) uses technology across numerous sources like sensor data, fleet history, maintenance records, and design documents, to monitor and assess the health of the concerned asset (component/ LRU/ subsystem/ system/ vehicle), to diagnose any fault present in the asset, and then to prognose its remaining useful life. In this way, IVHM capability enables aircraft Condition-Based Maintenance (CBM), resulting in the achievement of cost and time saving [3].

Currently, IVHM's principal focus is on components and LRUs, with subsystems and systems receiving less attention. For example, the Model-based Avionics Prognostic Reasoner (MAPR) provided a solution for non-intrusively monitoring the health and predicting remaining useful life for Electro Mechanical Actuators (EMAs), by using exclusive algorithms and reasoning techniques [4]. Similarly, a high frequency vibration monitoring system was developed to detect and isolate incipient faults in critical rotary components in engines, gear trains, and generators. [5]. On the other hand, a framework based on symbolic dynamic filtering was developed to isolate faults in engine subsystems by interpreting and fusing data from multiple sensors, in order to extract maximum information from the features [6]. These works show the extensive research carried out to isolate faults in components, LRUs, and subsystems in the aircraft.

However, none of these developments offers a rigorous process for the development of aircraft system diagnostics. A process that can effectively and efficiently inform aircraft maintenance to the OEM (Original Equipment Manufacturer), and the MRO (Maintenance, Repair, and Overhaul) organizations is needed. Such a system, based around OSA-CBM, is proposed here.

The idea for such an approach originated from a wider project named FAVER (Framework for Aerospace Vehicle Reasoning). The effect of the health state of one aircraft system over another due to their interactions, i.e. health assessment at the vehicle level, are not explored much in the technical literature, contributing to one of the significant gaps in the field of IVHM [7]. Aircraft accidents like the engine rollback incident on a Boeing 777 (2008) caused by a block in the fuel oil heat exchanger, and the emergency evacuation incident in a Fokker F28 (2002) caused by smoke in the cabin as a result of a crack in the APU compressor blade, are real life examples for multiple aircraft systems affected due to their interactions [2]. Isolating such cascading faults involves complex troubleshooting activities resulting in loss of time, and increased cost, due to extended downtime and prolonged maintenance. To address this gap, FAVER (Figure 7-2) was proposed, to isolate root causes and the effects of cascading faults considering the health of multiple aircraft systems using the concepts of reasoning and digital twins [2]. In FAVER there is a need for diagnostics built for each system under consideration, and hence the idea for generating such diagnostics in a rigorous manner emerged.

7.1.1 FAVER: A Framework for Aerospace Vehicle Reasoning

While the concept of digital twins has been used across IVHM in a variety of roles [3], FAVER simulates multiple aircraft systems interacting with each other, to demonstrate and isolate the effects of cascading faults at the vehicle level. Figure 7-2 shows the overall working mechanism of FAVER. As seen in the bottom row of Figure 7-2, FAVER demonstrates the interaction between aircraft systems using digital twins for the Electrical Power System (EPS), the engine, and the Environmental Control System (ECS), and Hardware-in-the-loop (HIL) for the fuel system. The fuel system supplies fuel to the engine, the engine provides shaft power to the EPS and bleed air to the ECS, and the EPS provides electricity to the other three systems. Once the digital twins produce results for healthy condition, and with certain fault modes, their health information is processed

through diagnostics set up at the individual system level, as shown in the middle rows of Figure 7-2. This is the subject of the current paper. Later, symptom vectors made from the health information from simulations of these individual digital twins, along with their diagnostic results, will be used by FAVER's reasoning as shown in the top row of Figure 7-2, to identify the root cause and the effects on the interacting systems at the vehicle level.

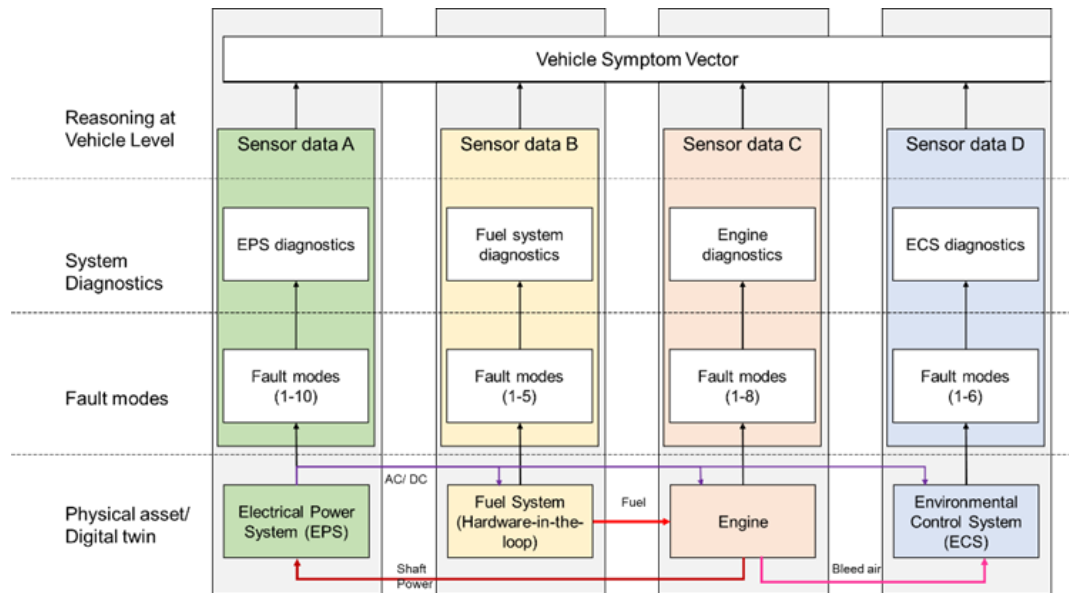


Figure 7-2 The schematic of FAVER

7.1.2 The Need for A Generalized Methodology for The Diagnostic Process

Across the industry, maintenance of aircraft is mostly done by the OEMs or MROs. The format of health data acquired from aircraft systems and the techniques applied by those carrying out, and troubleshooting, maintenance will vary considerably from platform to platform, due to a lack of standardization and development.

The functioning of IVHM systems at the systems level are not affected by the differences in the way health data is processed. However, in the case of a platform reasoner such as FAVER, since the reasoning is done at the overall aircraft level, processing data and carrying out the diagnosis for all concerned aircraft systems becomes complicated and tedious, especially when they are done using different methodologies.

Hence, there is a need to implement a generalized methodology to identify and streamline the repetitive steps involved in the diagnostic process. This will help save time

spent in developing diagnoses for each aircraft system and make the overall aircraft level health reasoning faster and more practicable; to serve this purpose the Open Systems Architecture for Condition Based Maintenance (OSA-CBM) framework is employed.

This paper demonstrates the application and usefulness of OSA-CBM as a generalized methodology for developing diagnostics for different aircraft systems. Section 7.2 focuses on the OSA-CBM methodology, and the steps involved in the process. Section 7.3 demonstrates the role of the OSA-CBM methodology through three different aircraft system use cases: the engine, the ECS, and the fuel system. Section 7.4 provides the diagnosis part of these use cases, and Section 7.5 summarizes the paper and discusses ideas for future work.

7.2 The OSA-CBM Methodology

OSA-CBM was developed in 2001 by an industrial team partially funded through a Dual Use Science and Technology (DUST) program [8]. Its aim was to define and develop open standards for distributed CBM. OSA-CBM implements ISO 13374 [9], Condition Monitoring and Diagnostics of Machines, to define the open standard. This standard breaks down the data collection, processing, and information flow into the six layers shown in Figure 7-3. It defines the interfaces (data structure and interface methods) between each step and so, indirectly, what can happen in each of the layers [10]. Each layer is now briefly described.

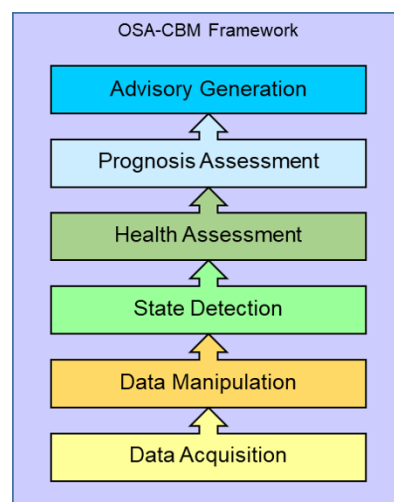


Figure 7-3 The Open Standard Architecture for Condition Based Maintenance (OSA-CBM) framework

Data Acquisition: Data is acquired from the target asset using sensors and other measuring devices, historical data, maintenance records, design documents, and other relevant information.

Data manipulation: The collected data is preprocessed and made useful for further processing. This step could use feature extraction and feature selection processes.

State Detection: Compares features against expected values or operational limits and outputs enumerated condition indicators (e.g. low, normal, and high).

Health Assessment: This stage determines if the health of a monitored system or subsystem is degraded. Several types of diagnostic algorithms, such as model based, data-driven, or hybrid algorithms, are used to assess the health of the asset. This layer generates a diagnostic record that proposes one or more possible fault conditions with associated confidence.

Prognostic Assessment: The remaining useful life and the future performance of the asset are then projected based on its current health. In some cases, uncertainty propagation is also estimated using prognostic algorithms.

Advice Generation: Provide recommended actions and alternatives. Recommendations include maintenance action schedules, modifying the operational configuration, or modifying mission profiles.

Because OSA-CBM is a standard and is not an instantiation of that standard, the above layer descriptions are open to interpretation and are given here for general guidance. OSA-CBM is defined using unified modelling language (UML), and is designed to enable multiple types of information to be processed without involving the technical interfaces. The OSA-CBM methodology can be applied at any stage of CBM, ranging from real-time health monitoring to portable maintenance and cloud services [8].

So far, in IVHM systems, the OSA-CBM framework steps are applied at the component, subsystem, and at the systems level. For example, at the LRU level, GE Aviation demonstrated the benefits of an IVHM system by monitoring the health state of an electromechanical actuator with parameters like the Hall effect sensor and supply current [11]. At the higher subsystem level, Boeing, the Air Force research laboratories, and Smith Aerospace together developed a program called Aircraft Electrical power systems

Prognostics and Health Management (AEPHM) and demonstrated health management of several subsystems for electrical actuation, fuel pump/ valves, and arc fault protection by monitoring parameters like torque efficiency, motor performance, and vibration analysis [12]. At the systems level, Safran has developed an Engine Health Management program by monitoring multiple parameters such as oil consumption monitoring, oil and fuel filters monitoring, actuation control loop monitoring, bearings vibration monitoring, and global performance monitoring [13]. Few more examples of application of OSA-CBM can be found in these references ([14]–[16]). It can be seen from the examples that the aircraft systems have multiple sensors, for multiple fault modes, linked to multiple diagnostic algorithms. In this paper, instead of following different processes for each aircraft system, a common methodology is sought and the series of steps from the OSA-CBM framework, leading up to diagnosis, will be implemented for all concerned aircraft systems. The last two steps in the framework, viz, Prognostic Assessment and Advisory Generation, are not used in this paper, as the goal is mainly to develop diagnostics. However, the results produced from diagnostics developed in this paper for the aircraft systems, i.e., the symptom vectors, can be used for further prognostic assessment to provide maintenance solutions in practice.

7.2.1 Contributions in this paper

In this work three aircraft systems are examined: the engine, the ECS, and the fuel system. Previously, a simulation model of the EPS (Electrical Power System) was developed, with Adaptive Neuro-Fuzzy Inference System (ANFIS) being implemented for its diagnosis [17]. It fits into the current framework but will not be replicated here. Diagnostics developed for these four aircraft systems are later used for vehicle level reasoning in the platform level reasoner, FAVER, which is not covered in this paper.

In this paper, the following contributions are made with regard to these three aircraft systems:

- i. The engine and the ECS digital twins are represented by MATLAB simulation models, while the fuel system is represented by hardware-in-the-loop (HIL), to show the flexibility in this approach. These models are run for healthy scenarios and a certain number of faults are injected (both local and interacting faults), and their data recorded.

- ii. Adhering to the OSA-CBM methodology, data from the digital twins is processed, and a feature selection algorithm (mRMR) is applied on all these systems to narrow down the health parameters necessary for developing the symptom vector as an input for the diagnosis.
- iii. The State Detection and Health Assessment steps of OSA-CBM are fused into one step by considering the diagnosis of faults as a classification problem. Here, the classification will take care of both detecting if the symptom vector is healthy or faulty. In the case of a faulty symptom vector, the classification algorithm will also diagnose the fault.

Diagnostics for all three aircraft systems follow the data-driven approach in order to maintain uniformity and save time. Only single faults are injected, simultaneous or multiple faults are not considered [2].

7.3 Use Cases

This section presents the setup of system level diagnostics for three aircraft systems: the engine, the environmental control system (ECS), and the fuel system. These have been chosen to demonstrate the use of the OSA-CBM methodology and for their fit into the overall FAVER architecture.

7.3.1 The Engine

The engine is a primary safety critical aircraft system. It produces thrust, powers the EPS, and provides bleed air required for anti-icing and the ECS of the aircraft. Among the types of jet engines, the turbofan is the most used by the civil airliners, as it optimises fuel consumption for the thrust produced. [Figure 7-4](#) shows a block diagram of a high bypass turbofan engine. The turbofan engine is made up of the following standard components: fan, compressor, combustion chamber, turbine, and nozzle. The engine sucks air through the fan (stations 2-21 from fig 3), and a portion of it is ducted into the core (station 23), while the rest of the air is bypassed (station 13) through the flow splitter to the bypass nozzle (station 19). The core flow is then compressed (stations 23-3) before being sent to the combustion chamber (stations 3-4), where the compressed air is mixed with fuel and burnt, producing hot gases and increasing the temperature. The hot air then expands across the turbine blades (stations 4-5), and then passes through the core nozzle (stations

7-9). The cold bypassed air passes through a similar nozzle (stations 17-19) and mixes with the core flow to produce the thrust that moves the aircraft forward [18]. Bleed air is extracted from the compressor (station 3) based on demand and supplied to the ECS and for anti-icing and de-icing.

7.3.1.1 Digital Twin Setup

The model chosen for the engine digital twin is the Pratt & Whitney JT9D open source turbofan engine model provided by T-MATS software in MATLAB Simulink [19]. This model features a high bypass turbofan engine with a bypass ratio of 4.8:1, one stage fan, three stage Low Pressure Compressor (LPC), 11 stage High Pressure Compressor (HPC), annular combustion chamber, two stage High Pressure Turbine (HPT), four stage Low Pressure Turbine (LPT), and a convergent nozzle.

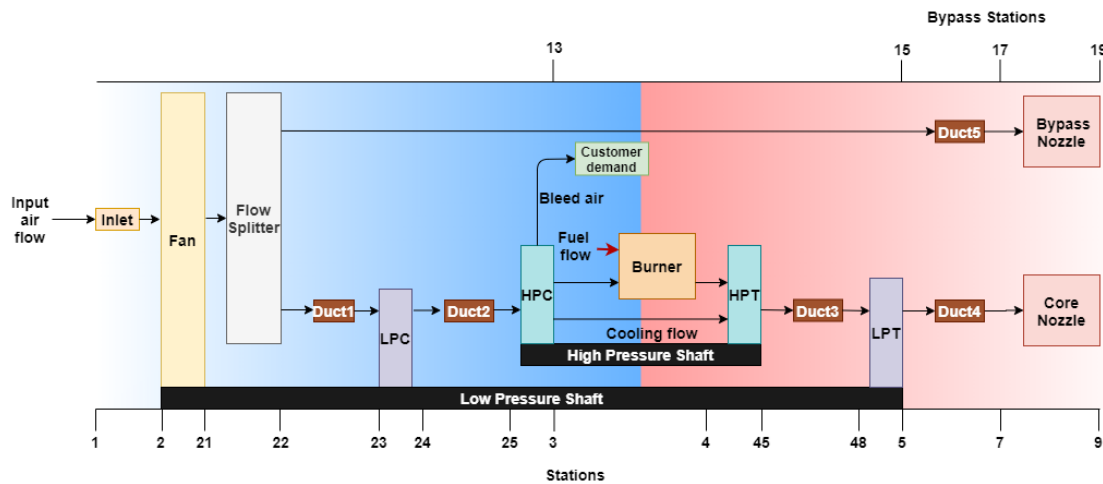


Figure 7-4 Turbofan engine block model in T-MATS (adapted from [20])

The healthy scenario is run with input conditions of 34000 ft altitude, 0.8 Mach number, and 1.91 pounds per second (pps) fuel mass flow rate, along with the efficiencies set as default in T-MATS [20]. A total of eight fault modes including local and interacting faults, are simulated in T-MATS for this paper, by reducing the efficiency of the subsystems as well as introducing leakages and blockages in the ducts and valves. The efficiency degradation and the leak induced in the fault modes are randomly chosen, and hence the resulting engine performance for each degradation mode can be viewed for trend but not cross-compared for quantitative effect. For the sake of brevity in this engine simulation only one mode of degradation is simulated for each subsystem. This single

mode of degradation for each fault is sufficient for demonstrating the approach advocated here, the diagnostics could always be updated with multiple degradations and different fault modes at a later date. The following are the eight selected faults injected in this simulation model; the first five are local faults, and the last three are interacting faults.

Local faults: These engine faults do not affect other aircraft systems. However, as a result of these fault modes, the subsystems function with reduced efficiency, contributing to overall reduced efficiency for the engine.

- i. **Fan FOD:** Foreign Object Damage in fan blade reducing the efficiency of the fan by 2.7%
- ii. **LPC Contamination:** Blade corrosion in LPC reducing the efficiency of LPC by 3%
- iii. **HPC Fouling:** Fouling in HPC reducing the efficiency of HPC by 1.1%
- iv. **HPT Blade Broken:** Partial loss of blade in HPT reducing the efficiency of HPT by 1%
- v. **LPT Blade Broken:** Partial loss of blade in LPT reducing the efficiency of LPT by 3%

Interacting faults: These faults are either caused by, or affect, aircraft systems other than the engine.

- vi. **Bleed Valve Stuck:** The bleed valve is stuck at 60 degrees. This fault affects the engine indirectly but directly affects the bleed air supplied to the ECS.
- vii. **CDP Leakage:** This fault is due to 0.5% leakage in the duct carrying the bleed air from the engine to the ECS. Similar to the previous fault, this fault has the potential to affect the bleed air supplied to the ECS but does not affect the engine directly.
- viii. **Reduced Fuel flow:** The input fuel flow is reduced by 10%, a fault that could have been caused by a reduction of fuel pump speed. The reduced fuel input affects the engine performance directly.

The simulation model is adapted to include a simple Proportional Integrator (PI) controller, which is programmed to meet the demanded thrust as the overall objective.

This is achieved by calculating Thrust Specific Fuel Consumption (TSFC) as shown in (7-1):

$$TFSC = m/f \quad (7-1)$$

where m is the fuel mass flow rate and f is the gross thrust. TSFC is provided as feedback to the PI controller, where the input fuel flow is altered by the controller to achieve the demanded thrust. As can be seen from Figure 7-5a, for all fault modes except **Reduced Fuel flow** the demanded thrust objective (27500 lbf) is met, which is due to the PI controller. In the case of the **Reduced Fuel flow** fault mode, the loss of mass flow rate affects the gross thrust, as they are directly proportional to each other, as shown in equation (1). This can also be seen in Figure 7-5b, where the mass flow rate of the fuel increases for all the fault modes in order to achieve the thrust, except for the **Reduced Fuel flow** fault mode where the mass flow stays constant and hence does not achieve the demanded thrust.

Parameters like flow, total temperature, total pressure, enthalpy, and fuel-air ratio are measured at several stations, along with other parameters such as fan speed, core speed, torque, and TFSC. The data collected from the simulations are then used for developing further diagnostics for the failure modes mentioned in this section.

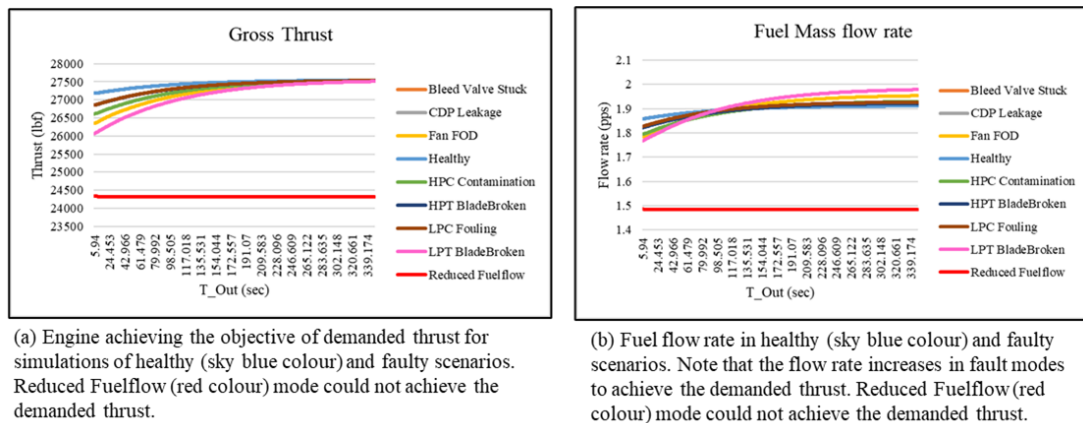


Figure 7-5 Gross thrust and fuel flow rate in Healthy and faulty scenarios

7.3.1.2 Data Acquisition and Data Manipulation

The engine digital twin configured in T-MATS is run in MATLAB R2019b for both healthy and faulty scenarios until the objective of thrust demanded is reached.

The next step is data manipulation, where the raw data collected from the simulations are formatted in MS EXCEL for further process in the OSA-CBM methodology. Only steady state data are used in the process; the rest of the data are ignored. Further, the number of health parameters to be monitored are narrowed down only to the essential features. This is because many parameters extracted from the simulation would be redundant to each other, hence monitoring all parameters from the simulation results would be time-consuming as well as would require higher processing power. In order to avoid this issue, factor analysis for each simulation model is carried out, and only the most influencing features are chosen for condition monitoring and diagnosis of the system.

Health parameters like flow, enthalpy, temperature, pressure, fuel-air ratio are measured at several stations along the gas flow path with other calculated measures like thrust, TSFC, core speed, fan speed, and torque. Additional measurements are made near the bleed valve, which does not fall under the gas flow path. These result in a total of 85 features from which data is extracted to monitor the performance of the engine. In order to save time and reduce the complexity of the monitoring process, only the important features that influence or represent the engine performance are selected for the diagnostic process. This is carried out in two steps.

Step 1: Only the ‘measurements’ from the Digital Twin that would correspond to sensors in a real engine, such as temperature (T_t), pressure (P_t), mass flow (W_f), speed, TSFC, and thrust, are chosen. This results in 35 features.

Step 2: From the 35 features, the second stage of selection is carried out using the Minimum Redundancy Maximum Relevance (mRMR) algorithm. The mRMR algorithm is a selection algorithm that finds the optimal set of features with respect to the response variable by identifying the parameters with maximal and mutually dissimilar relevance and minimal redundancy. This is done by calculating the mutual information between features and the response variable and between features itself [21].

In general, mutual information MI between two variables S,T is given by (7-2)⁵. This helps in finding how the variable S is helpful in reducing the uncertainty of variable T.

$$MI(S,T) = \sum_{ij} P(S = s_i, T = t_j) \log \left(\frac{P(S = s_i, T = t_j)}{P(S = s_i)P(T = t_j)} \right) \quad (7-2)$$

The mutual information is calculated for mRMR algorithm in the following fashion. Consider an empty set G , representing the optimal set of features ($G=\{\}$). Let H be the set containing the features, x,z ($H=\{x,z\}$), and let y be the response variable. The mRMR algorithm aims to add suitable features to the set G from set H . It does so by choosing the feature x (from set H) with maximum relevance (P_x), with respect to the response variable y , as calculated by (7-3) and assigning it to set G . In the next step, the feature z from set H is compared with x from set G , to minimise the redundancy by calculating (Q_x) between through (7-4). The mutual information quotient (MIQ) is then calculated using (7-5) and is used to rank all features in set G [22]:

$$P_x = MI(x, y) \quad (7-3)$$

$$Q_x = \left(\frac{1}{|G|} \right) \sum_{z \in G} MI(x, z) \quad (7-4)$$

$$MIQ = \frac{P_x}{Q_x} \quad (7-5)$$

$|G|$ is the number of features in set G . Once the ranks are assigned for all the features in set H , the optimal set G is filled with features that have maximum MIQ. In MATLAB R2019b, the importance of the feature is quantified by calculating its score. The score is the ratio of difference between the MI of target and considered feature to the average of MI of target and previously considered features. In short, the score represents the importance of the target feature with respect to the previously considered feature. The higher score for a feature indicates the importance of the predicting factor with respect to

⁵ If variable S and T are independent, $MI = 0$; if S and T are same, $MI = \text{entropy of } S$, where entropy is the property that measures the amount of randomness or surprise in the variable S. For the discrete random variable S with possible outcomes $s_1, s_2, \dots, s_n$ occurring with probability $P(s_1), P(s_2), \dots, P(s_n)$, entropy of S is defined as $H(S) = - \sum_{i=1}^n P(s_i) \log P(s_i)$

the response variable. The bigger difference in scores between the features show the confidence of the algorithm in choosing one feature over the other [22].

While the ranks of features show exclusivity of the features using mutual information, the scores show their importance as predictors with respect to the response variables. In the case of engine simulation, the set H contains the 35 features, and set G will be populated with the features having higher ranks. This is done by calculating MI for all 35 features with respect to the response variable, i.e., the fault classes. This step will calculate the relevance of these features for better representation of the fault classes. Similarly, MI is calculated for pairwise comparison of these 35 features and measures the redundancy of these features. MIQ of these 35 features are then calculated and set G is populated with features having maximum MIQ, i.e., better ranks. In this step, the features which are least relevant are ranked the lowest. Finally, the scores of these ranked features are calculated by comparing the MI of each of these 35 features with respect to the average of the rest. [Figure 7-6](#) shows the top 15 scored features as calculated by the mRMR algorithm.

The final list of features required for condition monitoring and diagnosis of the engine is narrowed down based on the scores from 35 to only four, viz: i) Core speed (HP shaft speed), ii) Pressure at LPC exit (station 24): Pt_S24, iii) temperature at LPT exit (station 5): Tt_S5, and iv) pressure at the exit of bleed air duct: Pt_AfCDP.

[Figure 7-6](#) shows that the core speed and Pt_S24 possess the topmost scores among the monitored features. This indicates that they are better predictors of the engine health performance. This can be correlated with [Figure 7-7a](#) and [Figure 7-7b](#), where the features, core speed, and Pt_S24 show clear distinction among different fault modes from the healthy case of the engine. (Only steady state is considered in this simulation. Hence, in [Figure 7-7](#), data from $t=230s$ is chosen for analysis and interpretation. The entire timeline from $t=0$ is shown in [Figure 7-7](#) for clarity). The turbine exit gas temperature from [Figure 7-6](#), Tt_S5, can understandably predict the health of the engine (as seen in [Figure 7-7c](#)), because of its location in the gas path. As corroborating evidence, across the literature, these three features are proven predictors of engine health performance [23]–[25].

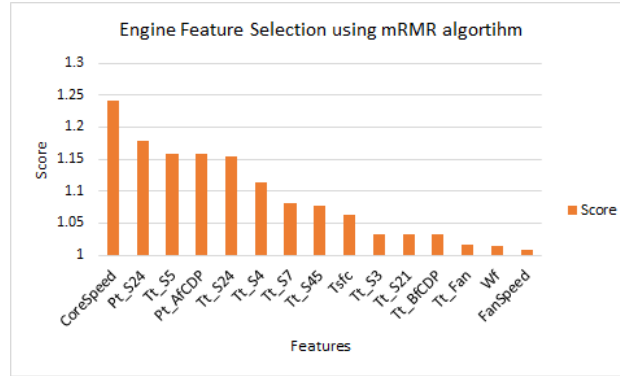


Figure 7-6 Top 15 features selected by mRMR algorithm for state detection and diagnosis of the Engine

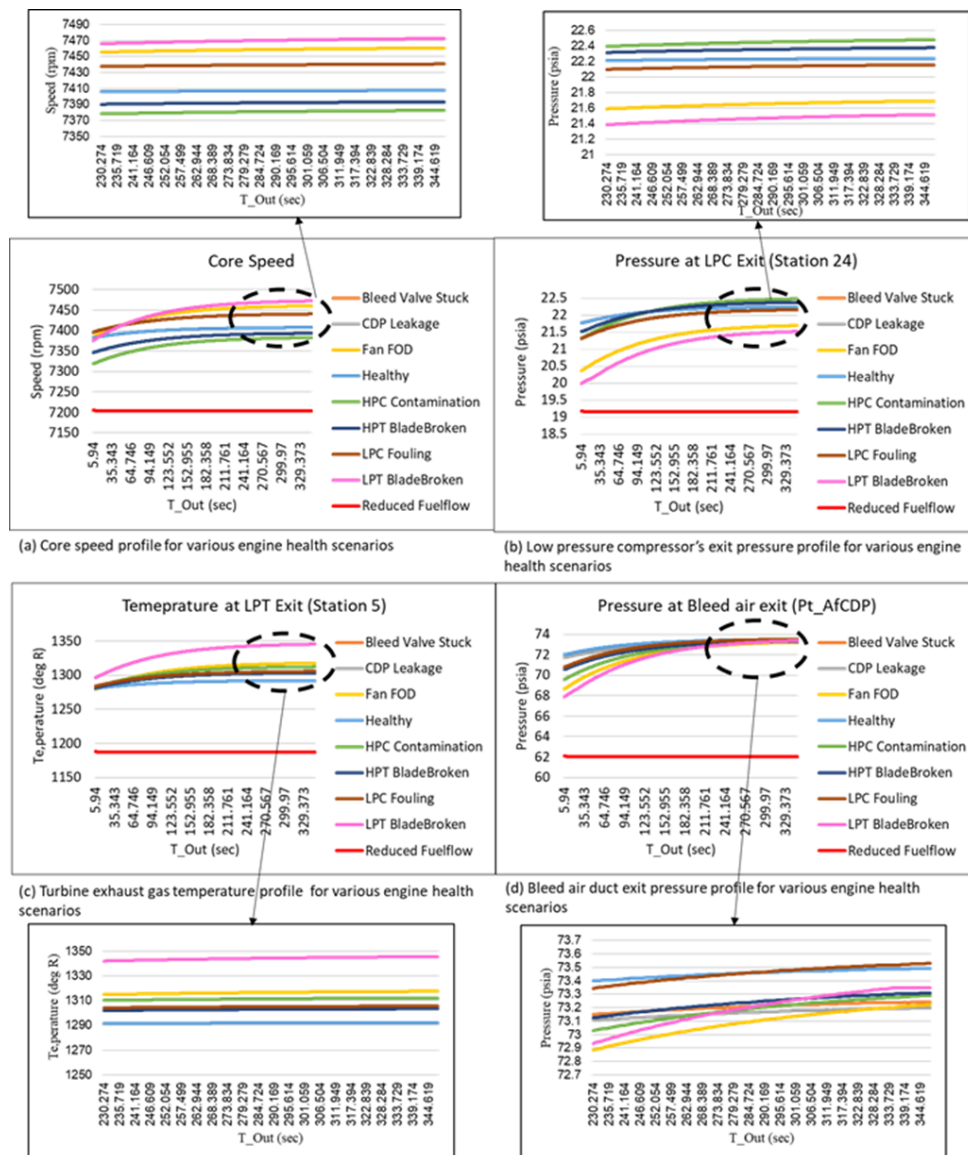


Figure 7-7 Performance of engine features for healthy and faulty scenarios

The strength of the mRMR algorithm is demonstrated in choosing the Pt_AfCDP feature in Figure 7-6. Identifying the importance of Pt_AfCDP would not be normally possible without the help of mRMR algorithm, as this feature is not directly related to any of the stations in the engine model, not located in the gas path of the engine, and was placed only to measure the exit pressure in the bleed air system ducts. However, the pressure sensor placement has resulted in this feature Pt_AfCDP to function as a predictor of engine health. This can be observed from Figure 7-7d, where the pressure at the bleed air exit has a distinct profile for each fault mode and is differentiated clearly from the healthy case. Thus, mRMR algorithm has successfully selected the most important predictors of the engine health performance as well as aided the selection of optimal sensor placement for monitoring the engine health. The next step is to use these four features to detect the current state of the engine and to diagnose fault modes.

7.3.1.3 State Detection

In the state detection step of OSA-CBM, the current condition of the system is monitored, to understand if the system is healthy, and, if not, to proceed for diagnosis. Figure 7-7 shows the comparison of the healthy mode with all eight fault modes for each feature selected to monitor the engine's current condition. These features are chosen because their profiles are distinct for every fault mode, making them easier to distinguish from the healthy profile and also to isolate the exact fault mode.

From Figure 7-7, it can be clearly seen that the **Reduced Fuel flow** fault mode is far from the healthy profile in all four features, making it easy to detect. In other words, in this fault mode, reduction of the fuel flow rate by 10% leads to the engine running off-design, resulting in reduced core speed, pressures, and temperatures measured at the stations (Figure 7-7) and reduced gross thrust (Figure 7-5).

In Figure 7-7a, it can be observed that the fault modes **HPC Contamination** and **HPT Blade Broken** have lower core speed when compared to the healthy profile, whereas all the other fault modes have higher core speeds. This is because the core speed refers to the high pressure shaft speed, which connects HPT and HPC (from Figure 7-4). Hence the loss of efficiency in either HPC or HPT would result in reduced core speed. However, in case of loss of efficiency in any other subsystem like the fan, LPC, or LPT, the high pressure shaft will run faster to obtain the demanded thrust, resulting in higher core speed when compared to the healthy profile.

In [Figure 7-7b](#), which shows the profiles of LPC exit pressure, **Fan FOD**, **LPC fouling**, and **LPT Blade Broken** fault modes have pressures lower than the healthy pressure profile. The reason behind the difference is the low pressure shaft that connects fan, LPC, and LPT. Loss of efficiency in these subsystems, because of fouling or the loss of material, results in lower intake of air resulting in lower compression developed by the LPC.

The opposite is true for **HPC contamination** and **HPT Blade Broken** fault modes. Here, the loss of efficiency in the HPC results in lower compression by the HPC, and loss of efficiency in the HPT results in the HPC being driven with a lower speed. Hence, there is slight pressure build up at the entrance of the HPC, as the HPC cannot compress to its fullest in both the cases.

In [Figure 7-7c](#), it can be observed that loss of efficiency of any subsystem in engine results in higher turbine exhaust gas temperature because of the extra workload on the engine to achieve the demanded thrust. While [Figure 7-7a,b,c](#) showed a clear distinction between fault modes affecting the main subsystems of the engine, the fault modes **Bleed Valve Stuck** and **Customer Discharge Pressure (CDP) leakage** are not seen clearly, as they have profiles similar to the healthy one. These two fault modes are distinguished only with the help of pressure measured at the exit of the bleed air duct, which in this instance shows a distinguishing profile for all the fault modes, as seen in [Figure 7-7d](#).

Thus, with the help of these four features chosen by the mRMR algorithm, the current health state of the engine is monitored, and the degradation in any of the subsystems can be easily detected by comparing their performance profile with that of the healthy profile. Therefore, the four features of the engine, viz, the core speed, Pt_S24, Tt_S5, and Pt_AfCDP together form the symptom vector for the engine diagnosis.

7.3.2 The Environmental Control System (ECS)

The Environmental Control System (ECS) of an aircraft maintains the in-flight conditions suitable for the passengers as well as for the effective functioning of the onboard equipment. The bleed air system (BAS) in the ECS receives bleed air extracted from the APU (on the ground) and the engine (during flight) and passes it to the anti-icing system and the PACK (passenger air conditioner). The PACK is the subsystem in the ECS responsible for conditioning of the bleed air to be supplied to the cabin. The PACK consists of a series of components such as: the primary heat exchanger (PHX), the

secondary heat exchanger (SHX), air cycle machine (ACM), high pressure water separator (HPWS), temperature control valve (TCV), and a flow control valve called the Pack Valve (PV). The PACK conditions the pressure (P), temperature (T), and specific humidity (SH) of the bleed air to match the flight deck requirements. The conditioned air then passes through a mixed manifold system and air distribution system to the cabin, and the cabin pressure control system regulates the air pressure inside the cabin as required by the flight deck [26].

7.3.2.1 Digital Twin Setup

In order to simulate the operation of the ECS, a simulation package called SESAC (Simscape Environmental control system Simulation under All Conditions) is chosen [26]. SESAC provides a library of components to perform simulation of the ECS under a wide range of operating conditions for both healthy and faulty scenarios. The SESAC component library has been configured to produce a detailed model of the B737-800 PACK. This model has been validated against actual data from the aircraft [26].

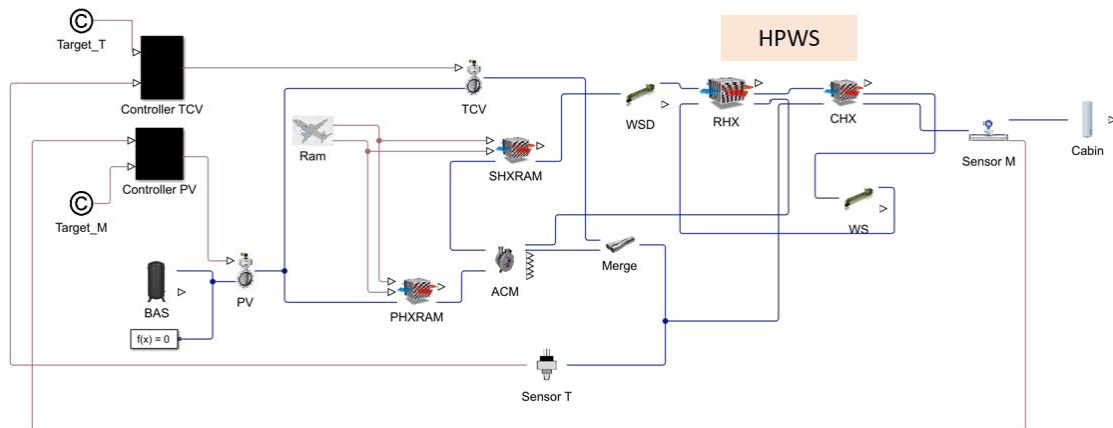


Figure 7-8 B737-800 ECS PACK model emulated in SESAC [26]

Figure 7-8 shows a schematic of the PACK model in SESAC. The bleed air from the engine or the APU is received at high temperature and pressure by the ECS. The PACK uses the PV to control the flow of this air, a portion of which is passed through the PHX for heat to be removed. This air is then cooled further by the compressor (as part of the ACM) and the SHX. Ram air is used by both heat exchangers (PHX and SHX) as a heat sink. The HPWS consists of a Reheater (RHX), condenser (CHX), and water separator (WS), which together remove any condensation from the air before it enters the turbine (the other ACM component). The cooled air is then merged with the bypassed hot air, the

TCV regulating the flow as demanded. Further details on the PACK model in SESAC can be found in this reference [26]. For the work here, the digital twin is fed with input conditions of 28000 ft altitude and 0.761 Mach number to meet the target cabin pressure of 79.1kPa, cabin temperature of 291.24K, and mass flow rate of 0.445 kg/s. In order to develop diagnostics for the ECS, the simulation of a 100% healthy scenario is run in SESAC, followed by injection of six faults which are listed below, combining local and interacting faults as for the engine case.

Local faults:

- i) **ACM 0.6:** only 60% mechanical efficiency of the ACM is simulated.
- ii) **PHX Fouling:** Fouling in the PHX, leading to its efficiency reduced by 50%.
- iii) **SHX Fouling:** Fouling in SHX, leading to 50% reduction in efficiency.

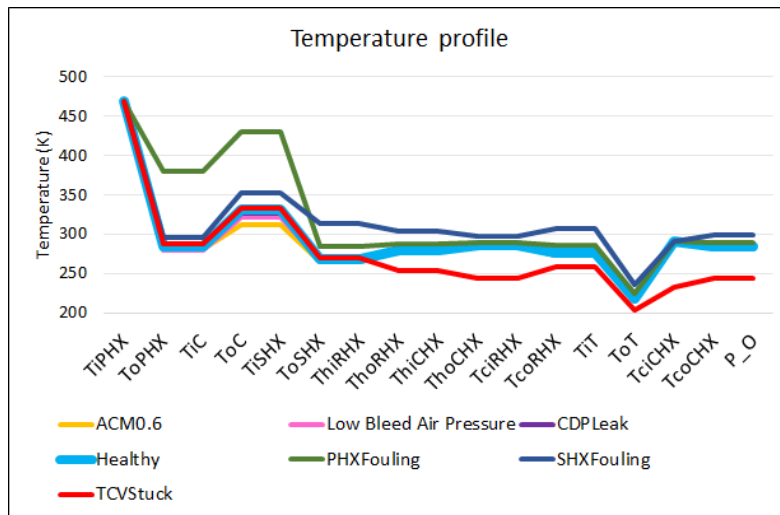
Interacting faults:

- iv) **CDP Leak:** this fault is due to 0.5% leakage in Customer Discharge Pressure (CDP) from the engine.
- v) **Low Bleed air Pressure:** this fault could be due to the HPSOV being stuck in the bleed air system, reducing the input bleed air to 75% of the original demand.
- vi) **TCV Stuck:** TCV is stuck at 10 degrees. This fault could be due to an intermittent supply from the EPS to the TCV.

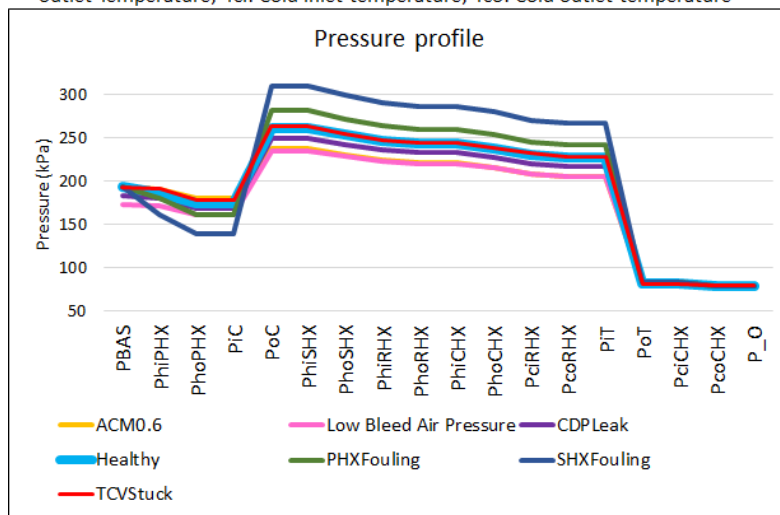
As in the case of the engine, the percentage loss of efficiency and leakage are chosen randomly, and hence these faults can be compared for trend, but cannot be quantitatively compared with each other. Temperature (T) and pressure (P) are measured at the inlet and outlet of every subsystem along the path of the bleed air.

Figure 7-9a shows the temperature profile for the healthy and faulty simulation scenarios. The monitored parameters are laid out in the way that represents the path followed by bleed airflow in the PACK. In the healthy scenario (sky blue color), the temperature of the bleed air is first reduced by passing it through PHX. It then increases slightly when compressed by the ACM compressor, after which the temperature decreases as the compressed air passes through the SHX. The air is reheated within the RHX and condensed in the CHX; hence the temperature increases in the hot sides of both RHX and CHX. The air is sent through the water separator and again through the cold side of RHX, where the temperature remains constant. The air is expanded by the turbine, with the temperature reducing dramatically. This expansion work directly powers the compressor

(on the same spool as the turbine). After this, the temperature of the expanded air increases when mixed with the hot airflow, to meet the cabin target temperature, before leaving the PACK outlet. Figure 7-9b shows the pressure profile, where, in the healthy scenario, the pressure of the bleed air decreases towards the PHX exit, followed by an increase in pressure when the air is compressed in the ACM. When the compressed air passes through the HPWS components (RHX and CHX), the pressure reduces gradually, followed by a dramatic reduction when the air is expanded through the turbine. This pressure is maintained as the air passes towards the PACK outlet.



(a) Temperature profile of PACK for healthy and faulty scenarios
(Ti: Inlet temperature; To: Outlet temperature; Thi: Hot inlet temperature; Tho: Hot outlet Temperature; Tci: Cold inlet temperature; Tco: Cold outlet temperature)



(b) Pressure profile of PACK for healthy and faulty scenarios
(Pi: Inlet pressure; Po: Outlet pressure; Phi: Hot inlet pressure; Pho: Hot outlet pressure; Pci: Cold inlet pressure; Pco: Cold outlet pressure)

Figure 7-9 Temperature and Pressure profiles for healthy and faulty scenarios for ECS PACK simulation in SESAC

Fault mode **ACM 0.6**, reducing the mechanical efficiency of the ACM, gives a lower temperature (Figure 7-9a) and pressure (Figure 7-9b) output from the compressor relative to the healthy case. Fault modes **PHX Fouling** and **SHX Fouling** lead to less heat exchanged when air passes through the corresponding heat exchangers, resulting in higher temperatures at their outlets, as seen in Figure 7-9a. However, the effect of degradation in PHX is negated by SHX, but the effect of degradation in SHX is carried forward into the HPWS, resulting in a higher turbine inlet temperature. Therefore, the pressure ratio increases between the turbine and the compressor, as seen in the pressure profile for **SHX Fouling** (Figure 7-9b). Fault modes **CDP Leak** and **Low Bleed air Pressure** allow air at lower pressure inside the PACK; hence, they are characterized by lower temperature and pressure profiles, similar to **ACM0.6**. In the **TCV Stuck** fault mode, when the temperature control valve is stuck at 10 degrees, it only allows air at a lower mass flow rate to pass. Hence the remaining air is cooled down much more than it should be, as seen in Figure 7-9a, and pressure profile is slightly higher than the healthy case.

While all six fault modes have individualistic profiles when compared to the healthy profile, monitoring all 32 parameters to detect the current state of the ECS would be tedious, time-consuming, and consume more computational power. Hence, feature selection is used, similar to the case of the engine, to choose only the most relevant parameters to the response variable.

7.3.2.2 Data Acquisition and Data Manipulation

The ECS digital twin is run in MATLAB R2019b for both healthy and faulty scenarios. For the ECS model in SESAC, temperature and pressure parameters are monitored at the inlet and exit of all ECS subsystems along its path. Using the mRMR algorithm explained in Section 7.3.1.2, the 34 monitored features are ranked and scored. The features with the first 15 scores are shown in Figure 7-10. It can be seen that the first two features have higher scores, followed by a large drop in the scores for the other features. This drop signifies the confidence level of the scores of the first two features. The features with the top two scores do not have better ranks, implying that they do not possess exclusive information, but they are better predictors of ECS health. Hence, the final list of four features chosen from Figure 7-10 for the condition monitoring and diagnosis of the ECS are i) Outlet temperature of the compressor (ToC), ii) pressure from the bleed air system

(PBAS), iii) Reheater cold side outlet temperature (TcoRHX), and iv) Compressor inlet pressure (PiC).

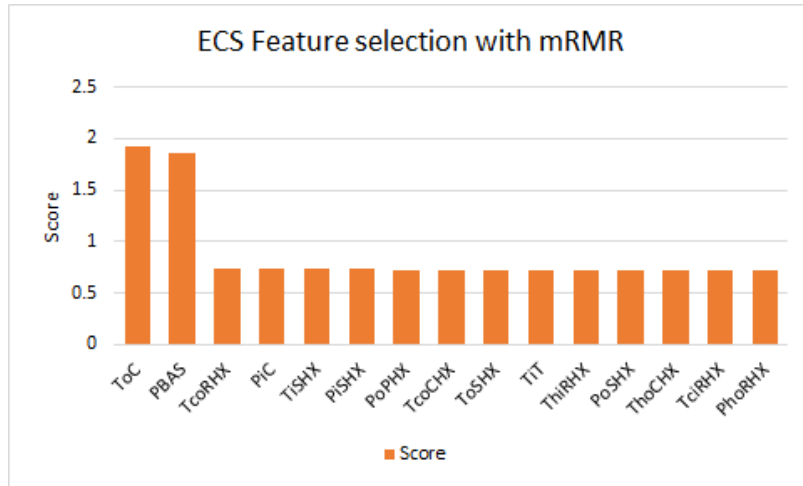


Figure 7-10 mRMR algorithm for the ECS features selection

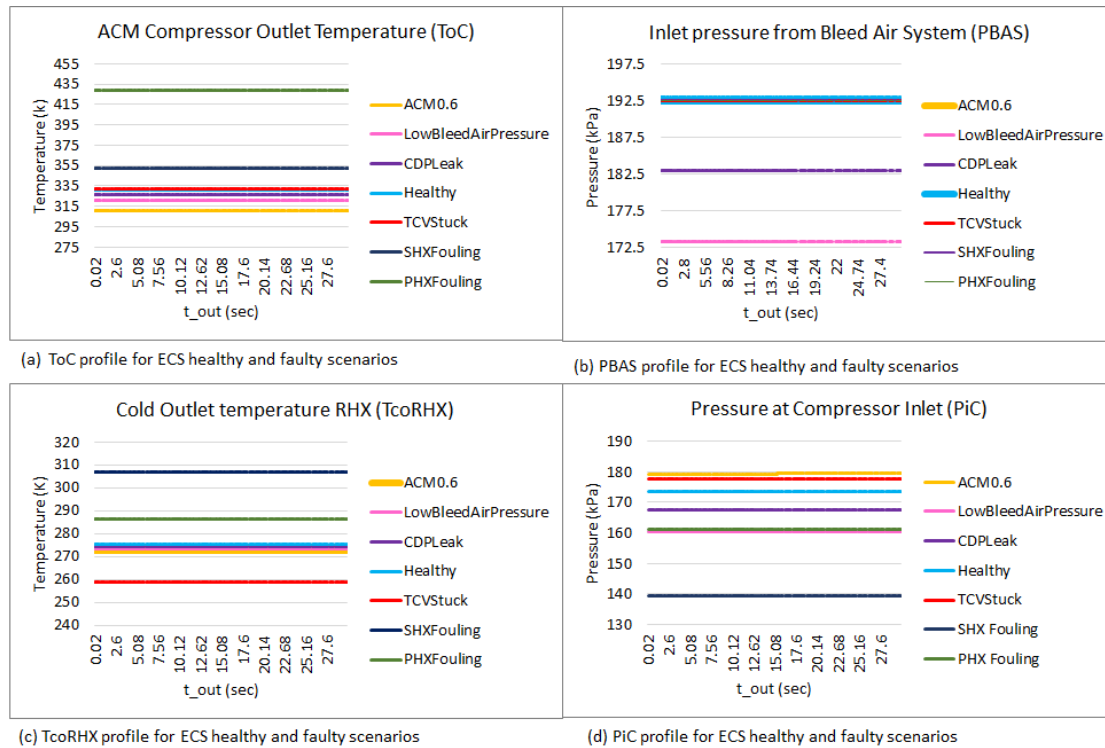


Figure 7-11 Healthy and faulty case profiles for ECS features selected by mRMR algorithm

7.3.2.3 State Detection

Instead of monitoring all 34 parameters listed in Figure 7-10, the current condition of the ECS will be detected by monitoring only the four parameters chosen by the mRMR algorithm, as shown in Figure 7-11. It can be seen from Figure 7-11 that each chosen

feature is able to distinguish more than one faulty scenario from the healthy profile. For example, from [Figure 7-11a](#), ToC has a distinguished profile for ***PHX Fouling***, ***SHX Fouling***, and ***ACM 0.6***. [Figure 7-11b](#) shows the differences between ***Low Bleed air Pressure*** and ***CDP Leak***. [Figure 7-11c](#) has a clear difference between ***PHX Fouling***, ***SHX Fouling***, ***TCV Stuck*** and the healthy scenario, and [Figure 7-11d](#) shows a distinction between most of the faulty scenarios. Using a combination of these four features alone will be sufficient to detect the presence of a fault in the ECS. Hence, the features ToC, PBAS, TcoRHX, and PiC form the symptom vector for the ECS to carry out the required diagnosis

7.3.3 The Fuel System

The primary function of the fuel system is to supply fuel to the engine and APU at the required pressure and temperature. In a typical twin-engine aircraft, the fuel system has three tanks, two in the wings, and one at the center of the fuselage. Both the engines consume fuel, at pressure, from the center tank until it reaches a low level, and then they use fuel from their respective main tanks. The fuel system transfers fuel from the tank, through the first stage fuel pump, the fuel oil heat exchanger, fuel filter, and then to the second stage boost pump. Fuel flow is then measured, after which the fuel enters the manifold and is supplied to the combustion chamber in the engine. Since the primary function of the fuel system depends upon the fuel transfer lines, any failure in these components could result in safety issues. The possibility of faults like pump failures, leakage in the fuel pipes, clogging in the filters, sticking valves, and blockages are taken care of during maintenance, in order to prevent their occurrence in flight. There are several diagnostic methods developed to isolate faults to the LRU level to help with troubleshooting activities [27]–[30].

7.3.3.1 Experimental Setup

The fuel system is represented by an experimental setup in the IVHM Center's laboratory and is therefore treated as HIL [31]. A simulation could have been written, and used as in the previous two examples, but the use of a rig for this example serves to show the flexibility of the overall approach. The fuel rig shown in [Figure 7-12](#) is closed loop, where fuel (water) is pumped from a reservoir R by a motor driven pump P that has an internal

relief valve, passing through a shut off valve S, followed by another valve V, a filter F, flowmeters FM1, FM2, a nozzle N and back to the reservoir R.

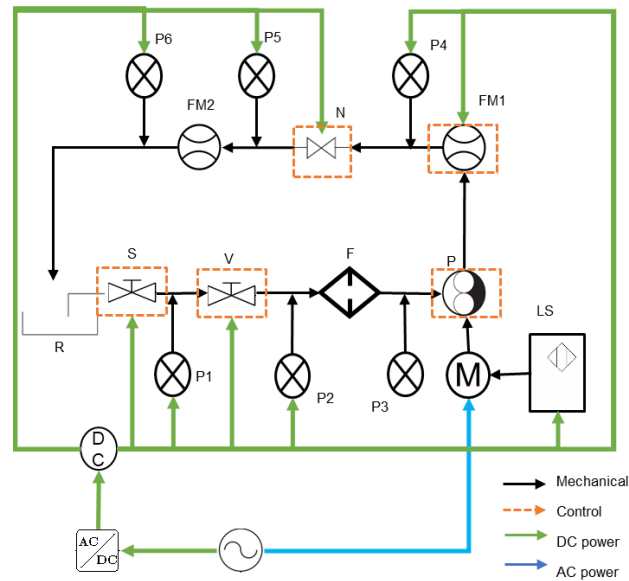


Figure 7-12 Layout of the fuel rig setup in IVHM Centre

The control system is programmed using LabView software, and the experiments are run at constant flow rate or constant pump speed. The speed of the motor driven pump on the test rig is controlled by LabView to meet with the required flow rate or a selected pump speed. Filters, valves, and nozzles on the test rig are emulated by Direct acting Proportional Valves (DPV), which are also controlled with LabView. Faults are injected, at different severity levels, by controlling the opening rates these DPVs. Pressure sensors P1, P2..., P6 are installed to monitor the pressure along the line, and they are powered by 12.8 VDC power supply. The laser sensor L which is used for measuring the pump speed is powered by 10-30VDC. The pump motor is powered by three phase 230/400 VAC at 50 Hz [31].

With the above-mentioned arrangement, the following five faults are injected into the fuel rig.

Local Faults:

- i) **Sticking Valve (SV):** Fault injected by manipulating the valve V by changing the opening rate of DPV from 100% to 90%, 80%, 70%, and 60%, emulating 0% severity of degradation to 10%, 20%, 30%, and 40% respectively.
- ii) **Clogged Filter (CF):** Fault injected by manipulating the filter F by introducing severity in increments of 10%, upto 40% degradation.

- iii) **Clogged Nozzle (CN)**: Fault injected by manipulating the Nozzle N by introducing severity in increments of 10% upto 40% degradation.
- iv) **Blocked Flow Meter (BFM)**: Fault injected by blocking the flow meter FM1 by introducing severity in increments of 10% up to 40% degradation.

Interacting fault:

- v) **Reduced flow (RF)**: Fault simulated by reducing the pump speed, because of low voltage supplied to the pump motor M. This is considered as an interacting fault, due to the involvement of the EPS in supplying reduced voltage to the pump motor.

With the pressure sensors, along with pump speeds, being monitored, the fuel rig is first run at the 100% healthy condition at a constant flow rate of 0.5 liter per minute (lpm), at the beginning of every experiment. The results from running at this condition are, as might be expected, highly repeatable. The health of component is reduced gradually from 100% to 90%, 80%, 70%, and 60% subsequently by manipulating the DPVS, giving a block of experiments for each failure mode.

7.3.3.2 Data Acquisition and Data Manipulation

For the fuel system, all experiments are initially run at the healthy condition, followed by fault injection. Readings from the sensors in the fuel rig are collected at a frequency of 1 kHz. [31]. The data is collected from the LABVIEW environment as .lvm files, which are then converted to csv files and processed, into MS EXCEL for further analysis.

In the fuel rig experiment, there are only six pressure sensors and a laser sensor for measuring pump motor speed. Hence, due to the low number of possible features, the mRMR algorithm is not applied here and all seven parameters are chosen for state detection and diagnosis.

7.3.3.3 State Detection

Figure 7-13 shows the pressure profiles and pump speed variation during each fault mode, while trying to maintain the constant fuel flow objective. Figure 7-13a shows the pressure profile for fault mode **Sticking Valve (SV)**, the pressure values at P2 and P3 decrease with increase in the fault severity. This is consistent with the fault being injected in the valve (V) that lies between pressure sensors P1 and P2. The fault results in an increased pressure drop across the valve, and since the fuel rig is controlled to run for constant flow, the pressure drop continues to be observed up to P3. Since the motor pump is controlled to meet the

constant flow of 0.5 lpm, the pump speed increases with increase in severity of the fault, as seen in Figure 7-13b. The values of P4, P5, and P6 do not change since the pump, delivering a constant mass flow, shields them.

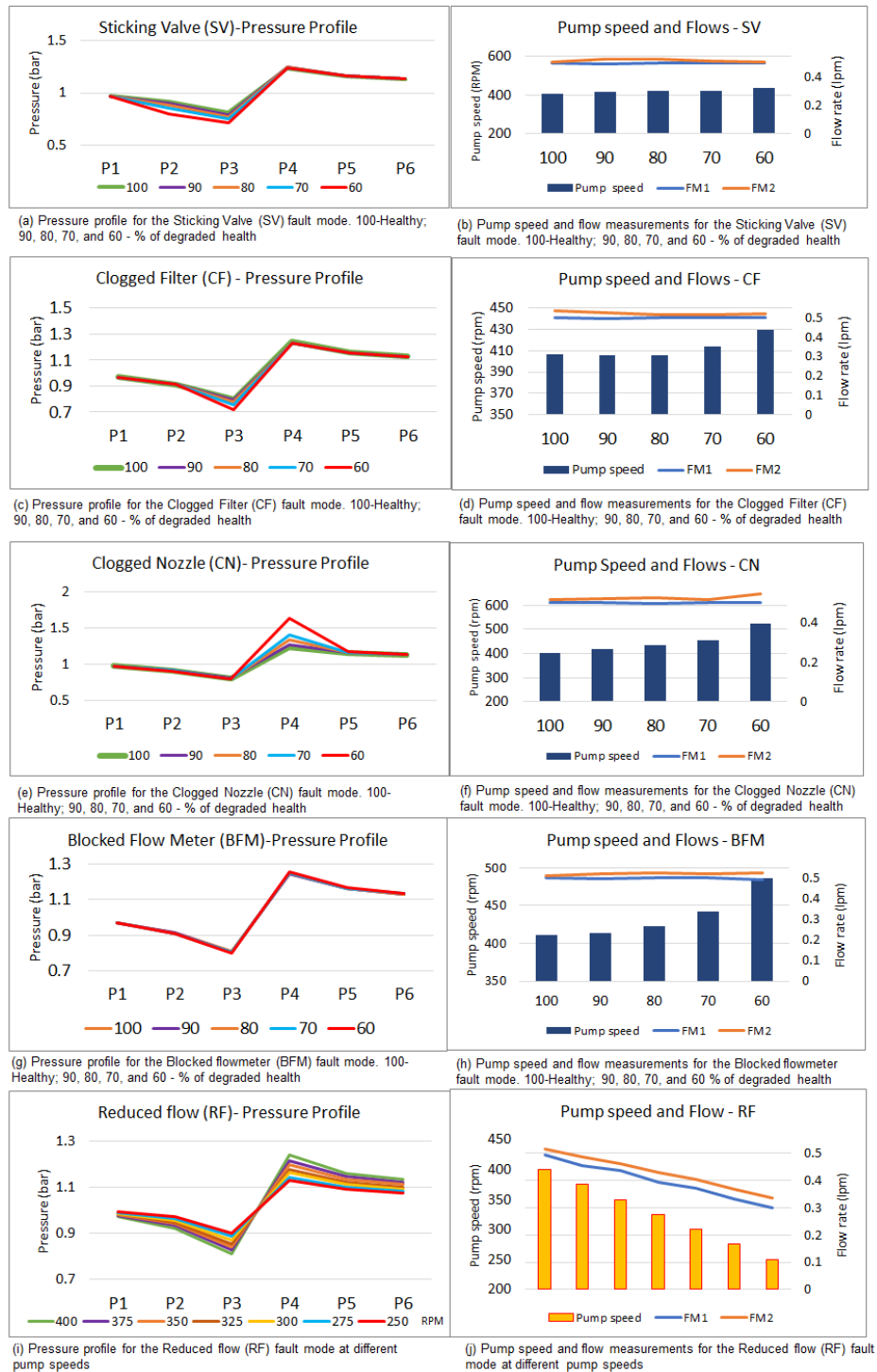


Figure 7-13 Pressure profiles, pump speeds and flows for various faulty scenarios from fuel rig experiment

Figure 7-13c shows the pressure profile and Figure 7-13d the pump speed profile, for a constant flow rate during **Clogged Filter (CF)**. There is an increase in pressure drop between P2 and P3 as the filter is located between these two sensors. The pressure values at other locations remain the same. The pump speed increases with the severity of degradation to meet with the constant flow demand, as observed in Figure 7-13d.

Figure 7-13e shows the pressure profile and pump speeds for **Clogged Nozzle (CN)**. The nozzle N is located after the pump between P4 and P5 as shown in Figure 7-12. When the **CN** fault is injected pressure increases at P4, just before the nozzle location. The rest of the pressure sensor readings remain the same. It can be seen from Figure 7-13f that the pump speed increases rapidly to meet the constant flow rate demand.

When the **Blocked flowmeter (BFM)** fault is injected (Figure 7-13g), it generates a back pressure to the motor pump, and the pump speed increases to maintain the flow at 0.5 lpm as observed in Figure 7-13h. There is no pressure sensor between the pump and the flowmeter, and hence the pressure profile does not indicate any change when this fault is injected.

In the case of the **Reduced Flow (RF)** fault mode the pump speed was reduced by 50 rpm per experiment, starting from 400 rpm to 250 rpm, to emulate the reduced flow in the line. The response of all pressure sensors is seen through the gradual change, as observed in Figure 7-13i, with respect to change in the pump speed. The change in flow rate with respect to the pump speed, as shown in Figure 7-13j.

Thus, the seven parameters (P1, P2,..., P6, Pump Speed) from the fuel rig experiment form the features for the symptom vector for diagnosis of faults in the fuel rig.

7.4 Diagnosis

Once the features have been short listed for being candidates for the symptom vector for each system, the next step in the OSA-CBM methodology is diagnosis.

In this paper, diagnosis is treated as a multi-class classification problem, in which, the symptom vector is classified as belonging to any of the fault class or labelled as healthy. In order to devise the diagnostics for all three aircraft systems, the data collected from the experiments are trained with machine learning classification algorithms and tested for their accuracy of classification. It is to be noted that the model based diagnosis can also be implemented using the OSA-CBM methodology. However, they are not tested in this paper,

and only data-driven diagnosis is implemented for creating uniformity in developing similar diagnostic functions for different aircraft systems.

Data from the chosen features from each use case is collected and split into training and testing datasets by Monte-Carlo simulation at 60%-40%, respectively. The datasets for training are uploaded into the MATLAB Classification Learner application and trained using various classification algorithms.

The training uses various supervised machine learning algorithms in this multi-class classification problem and the best performer is chosen for the next stage. The algorithms chosen for comparison are Decision trees, k-nearest neighbor (kNN), Linear Discriminant analysis (LDA), and Support Vector Machines (SVM). While the decision tree algorithm classifies the symptom vector based on rules and conditions developed using the training sets, the kNN algorithm assigns value to the symptom vector and classifies based on how its points are when compared to the training set. The LDA algorithm calculates the means and covariance matrix for different classes from the training set and classifies the symptom vector based on its computed values. In the SVM method, hyperplanes are created based on the classes from the training set. The symptom vectors are assigned co-ordinates and are classified based on their distances from the hyperplanes. The mathematics behind these algorithms can be found in these references [32], [33], and are not discussed in this paper. Datasets from each aircraft system are trained using all three algorithms and are tested for their accuracy of classification.

Table 7-1 shows the accuracy in classification, i.e., in diagnosing fault modes, using the machine learning algorithms, for each of the aircraft system use cases outlined in this paper. It appears from the table, and judicious choice of features, that any of these algorithms could be used to diagnose faults to quite high levels.

Table 7-1 Accuracy of classification by machine learning algorithms for diagnosis of faults in three aircraft systems

	Engine	Environmental Control System	Fuel System
Decision Tree	93.2%	100%	99%
k-Nearest Neighbour	99.1%	100%	88.9%
Linear Discriminant Analysis	99.9%	100%	89.2%
Support Vector Machines	99.8%	100%	95.1%

The LDA algorithm is chosen for the engine diagnostics, as it shows 99.9% classification accuracy. For the ECS, all four algorithms show 100% classification accuracy, and the kNN algorithm is chosen for the ECS diagnostics for its ease of computation compared to the other algorithms. The decision tree algorithm is chosen for the FS diagnostics for having 99% accuracy from table 1. [Figure 7-14](#) shows the confusion matrix for the best methods for each of these aircraft systems.

The better performer in each use case is chosen for the next stage, depending upon the application in which the diagnostic results are used. For example, if the diagnostic results are used in aircraft maintenance directly, they can be helpful in troubleshooting and fault isolation. On the other hand, these diagnostic results could also be used for remaining useful life calculations in the prognostic step of OSA-CBM methodology, followed by the advisory generation step, which is to develop suitable maintenance plans for the particular aircraft system. In FAVER, these diagnostic results will be used in the next stage of this research work for further reasoning at the vehicle level to identify cascading faults and root causes.

7.5 Summary and Future Work

In this paper, the OSA-CBM methodology has been used to demonstrate the process of developing system level diagnostics for multiple aircraft systems. [Table 7-2](#) presents a summary of the use cases demonstrated in this paper. In order to emulate a typical industrial scenario, the digital twin of each aircraft system has been developed separately and is isolated from the others. They have various faults injected, some affecting the local system and some interacting with other systems. They have a different number of features monitored for assessing the health of the systems. Developing diagnostics for such independent aircraft systems is generally time-consuming. However, the application of the OSA-CBM methodology has helped in establishing uniformity in building the system level diagnostics into the framework. The developed diagnostic functions in MATLAB can isolate a certain number of faults, with just a few chosen monitored features. These features form a symptom vector for each of the targeted aircraft systems.

While this paper demonstrated how the bottom three layers from FAVER's schematic in [Figure 7-2](#) (starting from the digital twin layer to diagnosis layer) are brought together, the future work will present how FAVER uses the symptom vectors generated from these aircraft systems to isolate faults and their root causes. These papers will also discuss the

architecture of FAVER, the reasoning mechanism involved, and demonstration of isolation of faults and identification of the root causes and their cascading effects on interacting aircraft systems.

Linear Discriminant Analysis for Engine										
True class	Bleed Valve Stuck	100%								
	CDP Leakage	1%	99%							
	Fan FOD			100%						
	HPC Contamination				100%					
	HPT BladeBroken					100%				
	Healthy						100%			
	LPC Fouling							100%		
	LPT BladeBroken								100%	
	Reduced Fuelflow									100%
Predicted Class										
	Bleed Valve Stuck	CDP Leakage	Fan FOD	HPC Contamination	HPT Blade Broken	Healthy	LPC Fouling	LPT Blade Broken	Reduced Fuelflow	
										True Postive Rate
										False Negative Rate

(a) Confusion Matrix for Engine diagnosis with Linear Discriminant Analysis

Fine k-Nearest Neighbour for Environmental Control System										
True class	ACM0.6	100%								
	CDPLeak		100%							
	Healthy			100%						
	LowBleed AirPressure				100%					
	PHXFouling					100%				
	SHXFouling						100%			
	TCVStuck							100%		
Predicted Class										
	ACM0.6	CDPLeak	Healthy	LowBleed AirPressure	PHXFouling	SHXFouling	TCVStuck			
								True Postive Rate	False Negative Rate	

(b) Confusion Matrix for ECS diagnosis with Fine k-Nearest Neighbor Algorithm

Fine Tree Classification for Fuel system										
True class	Blocked FlowMeter	>99%	<1%					<1%		
	Clogged Filter	<1%	99%		<1%					
	Clogged Nozzle			100%						
	Healthy		<1%		>99%					
	Reduced Flow					100%				
	Sticking Valve	<1%						>99%		
Predicted Class										
	Blocked FlowMeter	Clogged Filter	Clogged Nozzle	Healthy	Reduced Flow	Sticking Valve				
							True Postive Rate	False Negative Rate		

(c) Confusion Matrix for Fuel System diagnosis with Fine Decision Tree Algorithm

Figure 7-14 Confusion Matrices for best classification method for three aircraft systems

Table 7-2 Summary of the use cases demonstrated using OSA-CBM methodology⁶

OSA-CBM Steps	Engine	Environmental Control System	Fuel System
Digital twin Simulation Platform	T-MATS	SESAC	Test rig in IVHM Centre
No. of Fault modes	8	6	4 fault modes with 4 degradation levels + 1 fault mode with 7 degradation levels
Data Acquisition and Manipulation	85 features monitored	34 features monitored	7 features monitored
State detection	With 4 chosen features	With 4 chosen features	With 7 chosen features
Health Assessment (Diagnosis algorithm)	Linear Discriminant Analysis	k-Nearest Neighbour	Decision Tree

References

1. Mike Monroney Aeronautical Center. Federal Aviation Administration Joint Aircraft System/Component Code Table And Definitions. Federal Aviation Administration. Oklahoma City, Oklahoma 73125; 2008. Available at: https://av-info.faa.gov/sdrx/documents/JASC_Code.pdf (Accessed: 12 September 2020)
2. Ezhilarasu CM., Skaf Z., Jennions IK. Progress towards a Framework for Aerospace Vehicle Reasoning (FAVER). Proceedings of the Annual Conference of the PHM Society. 2019; 11(1): 1–9. Available at: DOI:<https://doi.org/10.36001/phmconf.2019.v11i1.887>
3. Ezhilarasu CM., Skaf Z., Jennions IK. Understanding the role of a Digital Twin in Integrated Vehicle Health Management (IVHM). IEEE International Conference on Systems, Man and Cybernetics (SMC). Bari, Italy: IEEE; 2019. pp. 1484–1491.

⁶ For FS, data was collected for every scenario at the frequency of 1khz from the rig and representative population of 3 seconds was considered for analysis. Hence, 3000 rows for each level of degradation for every fault and healthy state in FS (22 scenarios) = 66000 rows. Total data set = 66000 rows * 7 columns (features).

For engine, each simulation ran for t=350sec and data was collected at 25Hz. Hence 8750 rows*9 scenarios = 78750 rows. Dataset = 78750 rows * 4 columns (features)

For ECS, each simulation ran for t=30sec and data was collected at 50Hz. Hence 150 rows * 7 scenarios = 1050 rows. Total dataset = 1050 rows * 4 columns (features)

Available at: DOI:10.1109/SMC.2019.8914244

4. Vohnout S., Kim B., Kunst N., Gleeson B., Wagoner R., Balaban E., et al. A Model-based Avionic Prognostic Reasoner (MAPR). Infotech@Aerospace 2012. Reston, Virginia: American Institute of Aeronautics and Astronautics; 2012. pp. 1–17. Available at: DOI:10.2514/6.2012-2421
5. Watson M., Sheldon J., Amin S., Lee H., Byington C., Begin M. A Comprehensive High Frequency Vibration Monitoring System for Incipient Fault Detection and Isolation of Gears, Bearings and Shafts/Couplings in Turbine Engines and Accessories. Volume 5: Turbo Expo 2007. ASMEDC; 2007. pp. 885–894. Available at: DOI:10.1115/GT2007-27660
6. Sarkar S., Singh DS., Srivastav A., Ray A. Semantic sensor fusion for fault diagnosis in aircraft gas turbine engines. Proceedings of the 2011 American Control Conference. IEEE; 2011. pp. 220–225. Available at: DOI:10.1109/ACC.2011.5991168
7. Ezhilarasu CM., Skaf Z., Jennions IK. The application of reasoning to aerospace Integrated Vehicle Health Management (IVHM): Challenges and opportunities. Progress in Aerospace Sciences. Elsevier Ltd; 2019; 105(January): 60–73. Available at: DOI:10.1016/j.paerosci.2019.01.001
8. Mimosa. OSA-CBM UML Specification 3.3.1 Release. Machine Information Management Open Systems Alliance. 2010. Available at: <https://www.mimosa.org/specifications/osa-cbm-3-3-1/> (Accessed: 12 September 2020)
9. International Organisation for Standardisation. ISO 13374, Condition Monitoring and Diagnostics of Machines, International Organization for Standardization. 2015. Available at: <https://www.iso.org/standard/54933.html>
10. Sreenuch T., Tsourdos A., Jennions IK. Software Framework for Prototyping Embedded Integrated Vehicle Health Management Applications. Journal of Aerospace Information Systems. February 2014; 11(2): 82–97. Available at:

DOI:10.2514/1.I010046

11. Dunsdon J., Dunsdongge J., Harrington M., Haffington M. The Application of Open System Architecture for Condition Based Maintenance to Complete IVHM. IEEE Aerospace Conference. IEEE; 2008. pp. 1–9. Available at: DOI:10.1109/AERO.2008.4526639
12. Keller K., Amo A Del., Jordan B. Aircraft Electrical Power Systems Prognostics and Health Management (AEPHM). SAE Technical Papers Series. 2004; Available at: DOI:2004-01-3162
13. Bastard G., Lacaille J., Coupard J., Stouky Y. Engine Health Management in Safran Aircraft Engines. Proceedings of the Annual Conference of the Prognostics and Health Management Society, PHM. 2016. pp. 101–108. Available at: <https://pdfs.semanticscholar.org/a1c9/204a386439f2b22dc4f780305d5dda91121e.pdf>
14. Divakaran VN., Subrahmanya RM., Ravikumar GVV. WHITE PAPER INTEGRATED VEHICLE HEALTH MANAGEMENT OF A TRANSPORT. Infosys; 2018. Available at: <https://www.infosys.com/engineering-services/white-papers/Documents/aircraft-landing-gear-system.pdf>
15. Bense W. Prognosis and Health Monitoring Systems for Aircraft Engines. SAE International. 2013; Available at: DOI:10.4271/2013-01-2146
16. Du J., Wang S., Han L., Zhao S., Guo C. Prognostic management verification system of aircraft hydraulic power supply system. IEEE International Conference on Industrial Informatics (INDIN). IEEE; 2012. pp. 693–699. Available at: DOI:10.1109/INDIN.2012.6301133
17. Ezhilarasu CM., Jennions IK. A System-Level Failure Propagation Detectability Using ANFIS for an Aircraft Electrical Power System. Applied Sciences. 2020; 10(8): 2854. Available at: <https://doi.org/10.3390/app10082854>
18. NASA Glenn Research Center. How does a jet engine work? Available at: <https://www.grc.nasa.gov/www/k-12/UEET/StudentSite/engines.html> (Accessed:

16 January 2018)

19. Chapman, J. W., Lavelle, T. M., May, R. D., Litt, J. S., & Guo T. T-MATS: Toolbox for the Modeling and Analysis of Thermodynamic Systems. 2014. Available at: <https://doi.org/10.2514/6.2014-3932>
20. Chapman JW., Lavelle TM., Litt JS., Guo T. A Process for the Creation of T-MATS Propulsion System Models From NPSS Data. 2014. Available at: <https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20150000147.pdf>
21. Peng H., Long F., Ding C. Feature Selection Based on Mutual Information: Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy. IEEE Transactions on Pattern Analysis and Machine Intelligence. 27: 1226–1238. Available at: DOI:10.1109/TPAMI.2005.159
22. Mathworks. fscmr. Available at: https://www.mathworks.com/help/stats/fscmr.html#mw_bacedbf4-b0ee-4a43-97ee-5de3f6696e6a (Accessed: 8 July 2020)
23. Garg S. Fundamentals of Aircraft Turbine Engine Control. Lewis Field; 2015. Available at: https://www.grc.nasa.gov/WWW/cdtb/aboutus/Fundamentals_of_Engine_Control.pdf
24. Yilmaz I. Evaluation of the relationship between exhaust gas temperature and operational parameters in CFM56-7B engines. Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering. 2009; 223(4): 433–440. Available at: DOI:10.1243/09544100JAERO474
25. Yildirim MT., Kurt B. Aircraft Gas Turbine Engine Health Monitoring System by Real Flight Data. International Journal of Aerospace Engineering. 2018; 2018(i): 12. Available at: DOI:<https://doi.org/10.1155/2018/9570873>
26. Jennions I., Ali F., Esperon Miguez M., Camacho Escobar I. Simulation of an Aircraft Environmental Control System. Applied Thermal Engineering. Elsevier; 2020; 172(November 2019): 114925. Available at:

DOI:10.1016/j.applthermaleng.2020.114925

27. Kelly EM., Bartlett LM. Aircraft fuel rig system fault diagnostics based on the application of digraphs. Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability. 26 December 2007; 221(4): 275–284. Available at: DOI:10.1243/1748006XJRR88
28. Ofsthun SC., Abdelwahed S. Practical applications of timed failure propagation graphs for vehicle diagnosis. 2007 IEEE Autotestcon. IEEE; 2007. pp. 250–259. Available at: DOI:10.1109/AUTEST.2007.4374226
29. Snooke NA., Lee MH. Qualitative Order of Magnitude Energy-Flow-Based Failure Modes and Effects Analysis. Journal of Artificial Intelligence Research. 29 March 2013; 46: 413–447. Available at: DOI:10.1613/jair.3898
30. Lamoureux B., Massé J., Mechbal N. An approach to the health monitoring of a pumping unit in an aircraft engine fuel system. Proceedings of the European Conference of the Prognostics and Health Management Society. 2012; : 1–7. Available at: https://www.phmsociety.org/sites/phmsociety.org/files/phm_submission/2012/phmce_12_017.pdf
31. Yufei Lin. System Diagnosis using a Bayesian Method. PhD Thesis, Cranfield University; 2017.
32. Jung M., Niculita O., Skaf Z. Comparison of Different Classification Algorithms for Fault Detection and Fault Isolation in Complex Systems. Procedia Manufacturing. Elsevier B.V.; 19(2018): 111–118. Available at: DOI:10.1016/j.promfg.2018.01.016
33. MathWorks. Discriminant Analysis Classification. 2020. Available at: <https://www.mathworks.com/help/stats/discriminant-analysis.html> (Accessed: 1 August 2020)

8 FAVER - A Framework for Aerospace Vehicle Reasoning

A Framework for Aerospace Vehicle Reasoning (FAVER) is a robust framework developed to implement Integrated Vehicle Health Management (IVHM) with the aim of isolating faults at the aircraft level, along with their cascading effects and root causes. It encapsulates the Digital Twins (DTs) of the aircraft systems, allowing interaction and fault propagation due to the use of a modular architecture. FAVER also employs a novel combination of a vehicle level relationship matrix and a fault attributes database in its expert system. This paper expands FAVER to its intended potential and, in doing so, enables demonstration of scalability and flexibility. For the demonstrations, four aircraft systems: i) the Electrical Power System, ii) the Fuel System, iii) the Engine, and iv) the Environmental Control System, are taken into consideration. A couple of real aircraft accident scenarios with propagating faults are adapted for defining the use case scenarios. These test FAVER's ability to isolate faults at the system level, identify their root causes, and warn about the cascading effects. The paper evaluates the performance of FAVER's reasoning through a comparative study with a supervised neural network model, and discusses the pros and cons of FAVER's through the demonstrations.

8.1	Introduction
8.2	Building the framework
8.3	Scalability and Flexibility of FAVER
8.4	Demonstration Use cases
8.5	Evaluation of FAVER
8.6	Summary and Future work

Figure 8-1 Overview of Chapter 8

8.1 Introduction

Across the aircraft industry, Integrated Vehicle Health Management (IVHM) has focused mainly on health monitoring at the system level. Examples include: the engine health monitoring system, the landing gear health monitoring system, and the avionics health monitoring system. These IVHM systems contribute to Condition Based Maintenance (CBM), which consists of condition monitoring, diagnosing faults and degradations, and evaluating the remaining useful life of the concerned aircraft systems. However, there is a lack of focus on developing IVHM systems that consider the interactions between the aircraft systems. A Framework for Aerospace Vehicle Reasoning (FAVER) has been constructed to address this issue, isolating faults in an aircraft and identifying their root cause [1]. The effect of these faults may propagate from one system, affecting other systems in the interacting network.

With the trend of increasingly integrated systems in the aircraft industry, like more-electric and all-electric aircraft, the complexity of interaction between the aircraft systems is rising. This has resulted in the necessity for monitoring aircraft systems' health, not just 'within the systems' but also 'across the systems', i.e., at the vehicle level [1]. Developing an IVHM system by implementing FAVER addresses this issue by reasoning through health information from several interacting aircraft systems and identifying cascading faults, their root causes, and propagating effects [2], [3].

Fault propagation from one aircraft system to another is not uncommon and is usually addressed during maintenance. However, if the fault takes an unexpected propagation path, its cascading effects are difficult to troubleshoot. A report on General Aviation Safety 2018 shows that 20% of fatal accidents between 2001 and 2016 were caused due to Systems failures [4]. Several aircraft accidents have occurred due to faults from one aircraft system affecting other interacting systems. One such example is the engine rollback incident of a British Airways B777-200 ER in 2008, which resulted in the aircraft touching down 300m short of the runway at London's Heathrow airport. The rollback was due to restricted fuel flow to the engines, caused by ice blocking the fuel-oil heat exchanger [5]. A similar incident had taken place in Montana, with Delta Airlines B777-200 ER in 2008, resulting in an emergency descent from 39000ft to 31000ft [6]. Another example is the emergency evacuation of a Fokker F28 in 2002, caused by a small crack in the compressor blade of the Auxiliary Power Unit (APU). The debris from the

compressor blade was ingested in the gas path, cracking an oil seal and allowing oil spray to be released in the bleed valve, and eventually generating smoke in the cabin [7]. Such unexpected fault propagation paths lead to unplanned downtime consuming time and money. Troubleshooting and isolating these cascading faults requires a holistic view of the aircraft considering the interactions between its various systems, i.e. vehicle level health monitoring.

In the industry and in academia, most IVHM systems focus on components / Line Replaceable Units (LRUs) or at system level health monitoring. For example, the state-of-the-art IVHM system in the Boeing 777, Aircraft Diagnostics and Maintenance Systems (ADMS) is capable of isolating cascading faults caused due to interaction between the components [1]. Honeywell's Health and Usage Monitoring System (HUMS) uses sensors to monitor mission-critical components like the engine compressor, the engine accessory gearbox, and the drive trains in order to isolate mechanical faults and prevent catastrophic failures [8]. Similarly, other original equipment manufacturers such as Safran and Meggitt provide Engine Monitoring Units (EMUs) for aircraft and HUMS for helicopters [9]–[12], while consultancies like Infosys provide landing gear health monitoring solutions [13]. TEAMS-RT, a COTS product provided by Qualtech System Inc., is a real-time embedded diagnostic reasoner that focuses on systems level health monitoring [14]. On the other hand, failure propagation among surrounding areas are demonstrated using distributed diagnosis in HVAC systems [15] and power distribution systems [16] and through timed failure propagation graphs in fuel systems [17].

Only a few research works consider subsystems/systems interactions at the vehicle level [1]. A vehicle health monitoring system architecture patented by Honeywell employed a decision support system which would ignore a fault from an aircraft system, like the Environmental Control System (ECS), if there was a possibility that the said fault could have been caused by the system it interacts with, like the engine [18]. Similarly, Airbus Defense and Space demonstrated the effect of a fault in a fuel-cooled oil cooler on other independent systems using a modular framework that employed Bayesian Networks for reasoning [19]. These examples of IVHM systems show that the area of vehicle level health monitoring is still underexplored. It is also observed that the application of reasoning is significant for IVHM systems [1]. Hence, FAVER is proposed

to fill this significant gap in the field of IVHM [2].

This paper explains how FAVER employs reasoning combined with digital twins (DTs) to enable vehicle level health monitoring for the aircraft. To that effect, section 8.1 showcased the need for vehicle level health monitoring in IVHM systems by presenting the aircraft accident scenarios and state-of-the-art IVHM systems in industry. Section 8.2 provides FAVER's general schematic, objectives, architecture, and related works to set up this framework. Section 8.3 presents two important properties of FAVER: its scalability and flexibility, through demonstrations. In Section 8.4, the objectives of FAVER to isolate root causes of single and interaction faults in new added systems, and to resolve ambiguities are demonstrated with examples of fault propagation involving four aircraft systems akin to real aircraft incidents. Section 8.5 presents the pros and cons of FAVER's reasoning using a few use cases and evaluates the performance of FAVER's reasoning by carrying out a comparative study with neural networks. Section 8.6 summarizes the paper and provides some ideas for the future development of FAVER.

8.2 Building the Framework

8.2.1 Overview

FAVER has been developed through a System-of-Systems methodology to isolate faults in aircraft systems and identify their root causes and cascading effects across the systems that are in the interaction network [3]. FAVER is influenced by the OSA-CBM methodology and is based around two principal components: i) the Digital Twins, and ii) Reasoning. A DT is a virtual representation of any physical asset. In an IVHM framework, the DTs can be used as an alternate representation of the aircraft systems and help validate the outputs from the IVHM system [20]. In FAVER, the DTs of the aircraft systems are used to generate data for different 'what-if' scenarios, which could be used to anticipate these systems' health states and to explore how these aircraft systems interact with each other, thus creating the interaction network. Using this health information, FAVER incorporates various reasoning techniques to satisfy its objectives for fault isolation at a system level [2], [3].

Figure 8-2 shows the general working schematic of FAVER [3], consisting of four layers. The DT layer, the bottom row, consists of the aircraft systems considered in the framework. The next two rows cover the system level diagnostics and fault modes for

each of the aircraft system DTs. Considering these layers vertically for each system would be the classical way of conducting aircraft system diagnostics, with each system being stove-piped. The final layer, the top row, is the reasoning layer that connects all the verticals, using the symptom vector as input to enable fault isolation, including interacting faults. [Figure 8-2](#) shows four aircraft systems: the Electrical Power System (EPS), the Fuel System (FS), the engine, and the Environmental Control System (ECS). Each of these aircraft systems has its own representative DT, simulating its system level operations along with the (macro) vehicle level interactions shown in [Figure 8-2](#). The FS provides fuel to the engine, the engine supplies bleed air to the ECS and shaft power to the EPS, and the EPS provides electricity to all the other systems in the framework. These high level interactions are treated as input/output for each DT. The system level diagnostics can isolate a certain number of fault modes, which consist of faults with local effects (micro, within the system), as well as interacting effects (macro, propagating to other systems) [3]. The sensor data from each system is combined to form the symptom vector, which, along with the results from the four system level diagnostics, is input to the top level for further reasoning.

With this arrangement in [Figure 8-2](#), the objectives FAVER are to:

- a) Isolate single and interacting faults that affect local systems as well as have an interacting macro effect on other systems in the interaction network.
- b) Find and resolve ambiguity among the faults flagged.
- c) Expand the framework to accommodate other aircraft systems.

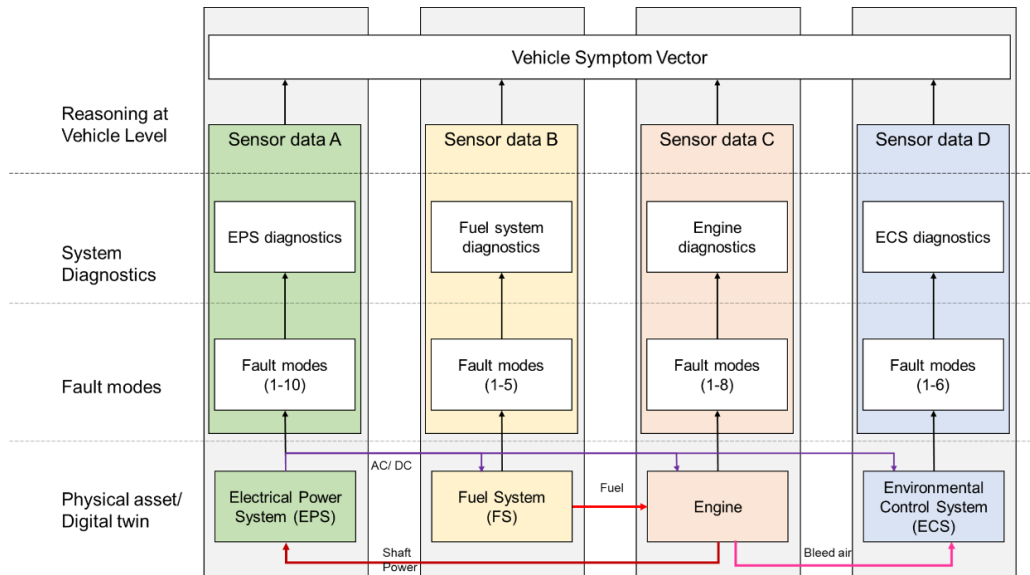


Figure 8-2 Working schematic of FAVER

8.2.2 Previous Work and Current Contribution

This paper is the final part of a three-year project to build the FAVER framework. Several parts of the framework have previously been documented and are shown in Figure 8-3. A brief recap of previous work is given here in order to explain the overall scale of the work and the current

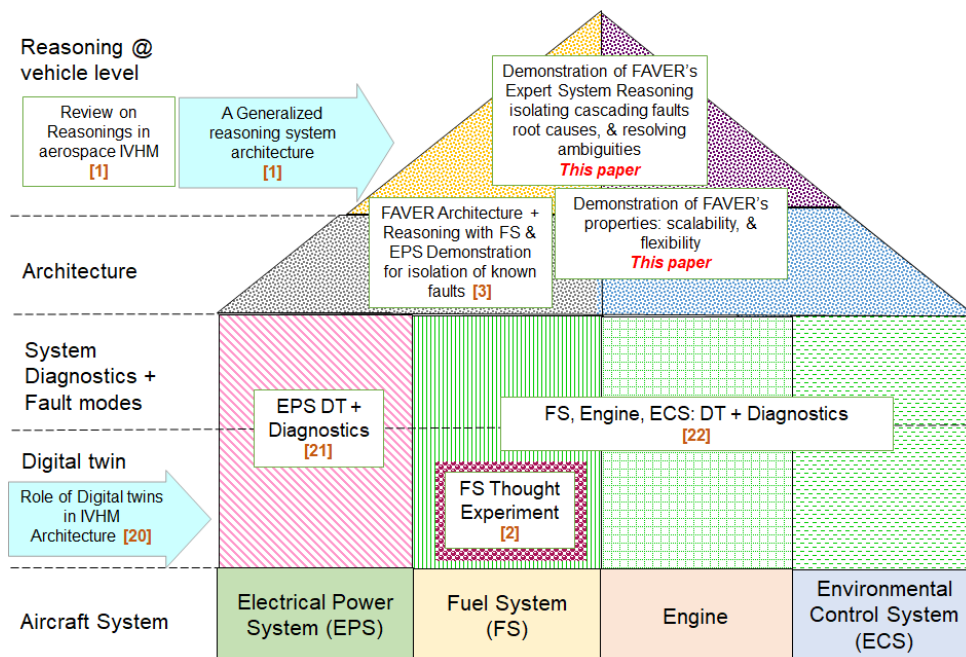


Figure 8-3 Previous work and current contribution

contribution:

- a) A thought experiment was conducted using the fuel rig [2]. It examined the complexity of such a system and helped define the assumptions necessary to avoid undue complication in developing this reasoning approach. The assumptions are given in the next subsection.
- b) A review of the reasoning literature [1], including many fields apart from aerospace, enabled the selection of reasoning approaches to be demonstrated in the project. A generalized reasoning system was derived from comparing various reasoning systems which is applied in developing FAVER's architecture. Given the framework that has now been formed, there is scope for additional approaches to be considered; some are given at the end of this paper.
- c) A review of the roles of a DT in the overall engineering process, and IVHM in particular [20], showed how a DT would work in an IVHM system, and allowed standards such as OSA-CBM to be introduced into the development process of FAVER.
- d) An initial architecture paper demonstrated how the selected architecture supported reasoning to isolate interacting faults and their root causes using the FS and EPS systems [3]. The modular architecture of FAVER is developed using a system-of-systems methodology; the paper presented the requirements for the platform and a step-by-step construction of its architecture, implementing the generalized reasoning system at the vehicle level. The FS used for demonstration is represented by the fuel test rig in the IVHM lab and is included as hardware-in-the-loop (HIL) to show the framework's flexibility. The selection of these systems also allowed ambiguity to be studied.
- e) As no DT existed for an EPS, one was created via a MATLAB-SIMULINK model along with its own diagnostics [21]. The diagnostics used ANFIS (Adaptive Neuro-Fuzzy Inference System) combined with causal reasoning and is capable of isolating ten fault modes.
- f) Subsequently, a generalized methodology, guided by OSA-CBM, was devised for system diagnostics and demonstrated on the fuel, ECS, and engine systems [22]. The diagnostic for the fuel system, represented by HIL as mentioned above, is a decision tree-based function which can isolate five fault modes. The engine was developed using T-MATS in MATLAB-SIMULINK [23]; its diagnostics function

is made up of machine learning algorithm k-Nearest Neighbour (kNN), and it can isolate eight fault modes. The ECS model is a MATLAB SIMSCAPE model called SESAC [24], and its system level diagnostic function uses a Linear Discriminant Analysis (LDA) algorithm and can isolate six fault modes.

The current contribution is also shown in [Figure 8-3](#) and comprises a demonstration of FAVER's reasoning and significant features through real-world examples. It contains the following:

- a) The vital properties of FAVER – scalability and flexibility are demonstrated in this paper.
 - i) The flexibility of FAVER is demonstrated by using different types of system DTs and system diagnostics in the framework.
 - ii) For scalability, the demonstration is done by expanding the framework (that already has the EPS and the FS from previous work) in two steps:
 - i. Including two more complicated aircraft systems: the engine and the ECS to the framework.
 - ii. Including new faults to every system diagnostics in the framework.
- The demonstration details through activity diagrams, the requirements and steps necessary to include a new system into the framework, and new faults to the systems and show how easily FAVER can be adapted for such a change.
- b) FAVER's reasoning uses a novel knowledge base for its expert system and a mixture of reasoning techniques. This paper demonstrates the strength of FAVER's reasoning in isolating the root causes and cascading effects of interacting faults. For this purpose, a variety of examples are adapted from real aircraft accident cases. Various combinations of faults, such as simultaneous single faults, multiple interaction faults, single as well as interaction faults, are also examined to investigate the capability of FAVER's reasoning to resolve ambiguities arising from such scenarios.
- c) The overall framework is critically evaluated in two steps:
 - i. This paper presents the cost of using FAVER's reasoning, such as the need to assign priorities and misclassification costs.

- ii. FAVER's reasoning is compared with a data-driven reasoning using a neural network, and the pros and cons of using FAVER's reasoning are explained.

8.2.3 Assumptions

Following a thought experiment [2] a number of assumptions were made in order for the framework to be developed in a three-year time period while incorporating sufficient complexity to enable realistic demonstration of capability. The assumptions made in this work are:

- i) The control system is not considered and is assumed healthy. Introducing the control system to the framework would have significantly increased the complexity of the interactions. There are two systems – electrical and controls – that span most of the systems on an aircraft. By considering the EPS in this work, one of these far-reaching systems was explored.
- ii) Sensors are always healthy.
- iii) Only steady state is considered. Given what is now known about the reaction of most systems to time-dependent input, this would be a relatively easy upgrade.
- iv) Only single faults are considered as the emphasis here was to look at interaction faults.
- v) There are no false alarms.

These assumptions could obviously be removed or conservatively updated in future projects.

8.2.4 Architecture

The schematic of FAVER is implemented through the architecture shown in [Figure 8-4](#). This brings together three distinct areas: i) DT, ii) Reasoning, iii) Communications; these will now be described in this order.

Each DT is encapsulated and performs I/O with the DT Facilitator. It does not correspond directly with other DTs, as this would lead to a high degree of complexity when a new DT was introduced to the system. Instead, the DT Facilitator orchestrates running each DT and the interactions between them. Using this approach enables systems on different scales to work together. For example, the engine DT is for a Pratt and Whitney JT9D engine, while the fuel system is a benchtop fuel system in the laboratory.

The DT Facilitator handles the interaction between the two systems on a 0-100% basis rather than in engineering units. This modularity feature in the DT layer prevents a complex, time-consuming, Verification, and Validation (V&V) process. This would otherwise be required for every update in the DT, or all the DTs in the interaction network when a new system is added to the framework.

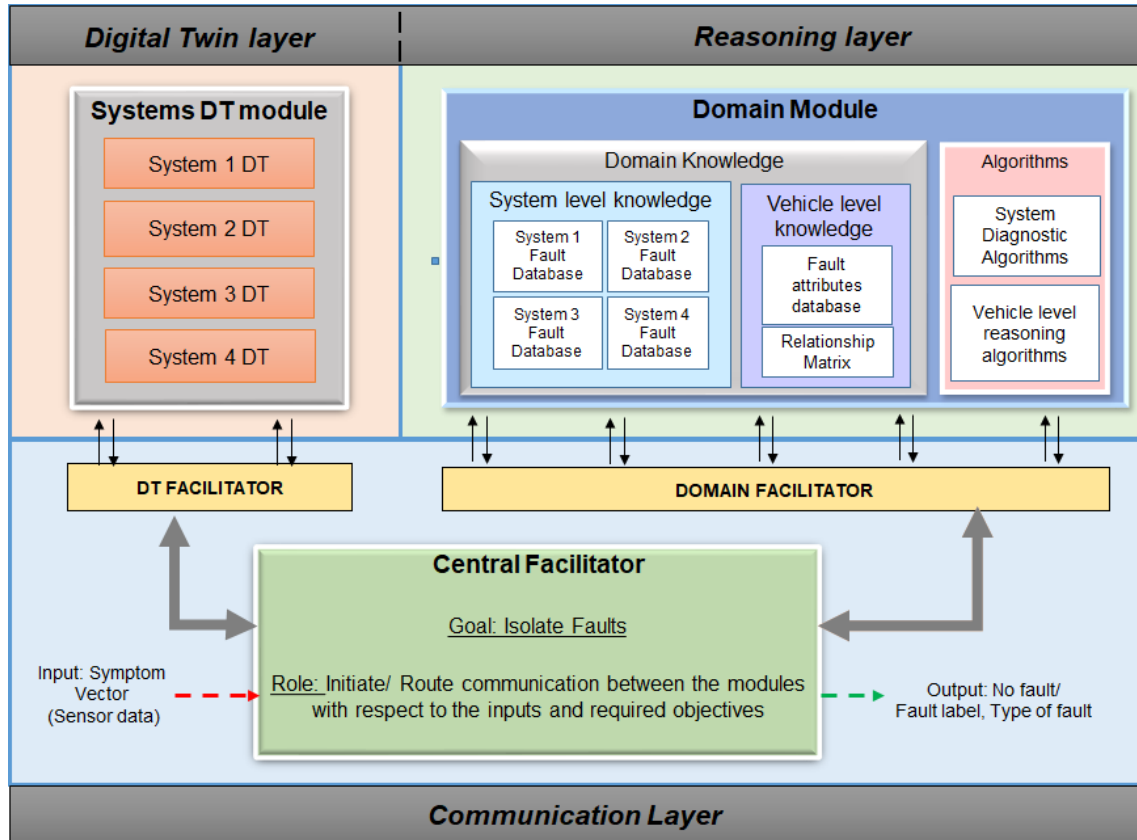


Figure 8-4 The architecture of FAVER

The Reasoning layer, acting through the Domain Facilitator, is responsible for retaining knowledge at the vehicle level. It simultaneously processes the symptom vectors aligned with multiple system diagnostics to carry out reasoning at the vehicle level, to isolate faults and their root causes. The Reasoning layer is adapted from a generalized reasoning system derived as part of the background work for setting up this framework [1]. It consists of a domain module acting as the brain of FAVER. The domain module comprises of diagnostic functions at the systems level and an expert system at the vehicle level. FAVER's expert system at the vehicle level consists of a knowledge base that is made up of a novel combination of a relationship matrix and a fault attributes database. This combination enables a simpler expansion of knowledge, contributing to the

scalability of FAVER. Section 8.3 shows examples demonstrating this particular feature of FAVER. The Reasoning layer also has a rule based approximate reasoning algorithm to isolate root causes of faults, and predict their cascading effects based on results from system level diagnostics [3].

The Communications layer contains the Central facilitator, which forms the bridge between the DT layer and the Reasoning layer. The Central facilitator fetches the input symptom vector from the user and passes it on to the system level diagnostics in the reasoning layer through the Domain facilitator. The working mechanism of the Central facilitator is based on a case based reasoning methodology. In this the results of rule based approximate reasoning from the reasoning layer are retrieved, interpreted, and revised through abductive reasoning, and the new knowledge is returned back to the Reasoning layer [3]. The results on the type of faults, their root causes, and cascading effects are then output back to the user, through the Central Facilitator. The activities carried out through these modules from architecture can also be seen in (Figure 8-6 and Figure 8-7 in this paper, for better understanding.

Using this architecture, fully populated, as shown in Figure 8-3, FAVER's objectives (Section 8.2.1) can be realized and demonstrated. Five test cases have been selected to show attainment of FAVER's fault isolation capability:

- i) **Known fault isolation – Single:** Demonstrate isolation of a single fault occurring either in the FS or the EPS.
- ii) **Known fault isolation – Interaction:** Demonstrate isolation of a fault originating in the EPS and affecting the FS due to their interaction.
- iii) **Ambiguity:** Ambiguous scenarios, where two system level diagnostics indicate faults, were simulated and FAVER's reasoning was able to identify which system is the origin point and which system was flagged as a result of fault propagation. *More complicated scenarios are demonstrated in this paper.*
- iv) **New fault (previously unknown) isolation – Single:** *To be demonstrated in this paper*
- v) **New fault (previously unknown) isolation – Interaction:** *To be demonstrated in this paper*

The first three of these five test cases were defined to meet the functional requirements of FAVER, which is to isolate faults at the vehicle level and identifying their root causes

and cascading effects. These cases were demonstrated successfully in the previous work using two systems, the EPS and the FS [3]. The last two test cases are designed to satisfy the non-functional requirements of FAVER: scalability and flexibility, which this paper demonstrates, in addition to the functional requirements.

8.3 Scalability and Flexibility of FAVER

8.3.1 Scalability: Expansion of FAVER

The term scalability has different definitions based on the industries where it is applied. From a system's point of view, scalability is defined as an attribute that defines a system's capability to adapt to increasing demand or workload. This property is significant in FAVER for two reasons:

- i) Health reasoning at the vehicle level requires consideration of the behaviours of multiple aircraft systems. Implementation of a vehicle level health reasoner like FAVER requires a lot of background preparations - setting up the aircraft system's digital twin and its system level diagnostics. Hence FAVER has to be implemented in stages. While FAVER aggregated two aircraft systems (the FS and the EPS) for its initial sets of demonstrations, scaling up the framework to accommodate more complicated aircraft systems is necessary to achieve health assessment with a holistic view of the aircraft.
- ii) Aircraft systems are ever-evolving, adapting to achieve different objectives such as more electric or all-electric aircraft and sustainable aviation. This leads to constant updates to the system level diagnostics, either to include new complicated faults that are to be diagnosed, or to remove the obsolete ones. Since the vehicle level reasoning in FAVER is built upon the system level diagnostics, it has to adapt to process new sets of faults from these aircraft systems, without much changes to its structure, in order to isolate their root causes and predict cascading effects.

In this section, the scalability of FAVER is demonstrated by expanding the framework to accommodate new aircraft systems, as well as adding new faults to the system diagnostics which are already integrated to the framework.

8.3.1.1 Adding New Systems to the framework

In order for FAVER to be judged expandable and scalable it needs to be able to accommodate new systems, and their respective diagnostics, with minimal change. In accepting new systems, it also needs to be robust, with potential problems being easily isolated. It accomplishes this by using a modular design where functions are carefully separated, and interactions minimized, enabling rapid troubleshooting within FAVER, should it be needed. New systems can be added to the framework through the following steps.

Requirements to be met by the systems:

To be included in the architecture shown in [Figure 8-4](#), the following requirements should be met by the new aircraft systems.

- i) *For the DT layer in FAVER:* The representation of the system DT is required and will be incorporated in the DT layer of FAVER.
- ii) *For the Diagnostics layer in FAVER:* The aircraft system models should have their individual system level diagnostic modules in place for isolating a certain number of fault modes. This will be integrated with the system level knowledge in the Reasoning layer of FAVER.

Changes to be made in FAVER:

To expand the framework and include systems into it, the following changes must be made in FAVER.

- i) *In the Reasoning layer:* As mentioned in the previous section, the vehicle level reasoning in FAVER is built upon the diagnostic capabilities of the aircraft systems in the framework, bolstered by an expert system consisting of domain knowledge at the vehicle level. Hence, the domain knowledge at the vehicle level must be updated to reflect any new system added into the framework. Unlike the traditional relationship matrix, FAVER has a novel combination of a relationship matrix with a fault attributes database for its expert system knowledge base, which requires only simple updates to the domain knowledge and is easily scalable [3]. In this expert system, for every new fault added to the framework, the fault attributes database need only two attributes to be added to the system: i) *macro_cause* (Possibility of fault caused by an interacting system), and *macro_effect* (Possibility of fault affecting an

interacting system). These changes to the knowledge base are sufficient to accommodate new systems and their faults modes into the framework.

- ii) *In the Central facilitator:* Along with the updates in FAVER’s domain knowledge, the Central facilitator has to be upgraded to fetch symptom vector as input to the engine diagnostics integrated to it. This involves providing the monitored engine parameter details to the Central facilitator’s program, to be integrated with the user interface.

Demonstration of expansion of FAVER

The scalability of FAVER is demonstrated in this subsection by adding new systems, the engine, and the ECS, to the framework. Being one of the principal systems of the aircraft, the engine interacts with multi-physical systems across the aircraft, inducing complicated interaction scenarios with both micro (within its own system) and macro (with other systems) effects, which can be considered inside FAVER. For example, consider the interaction between the four aircraft systems: the FS, the engine, the EPS and the ECS, shown in Figure 8-5. The initial demonstrations of FAVER had considered only two systems: the EPS and the FS [3]. The EPS provides electricity to the pump and the valves in the FS, as indicated by the black colored arrow in Figure 8-5. When the engine is added to the framework, the number of interactions increases. The FS provides fuel to the engine, the engine provides shaft power required for the EPS, and the EPS provides power to the valves in the engine, as shown by the red colored arrows in Figure 8-5. Similarly, adding the ECS to the framework results in even further interactions as the EPS provides power to the ECS valves, and the engine provides the bleed air, as indicated by the green colored arrows in Figure 8-5. Thus, by adding the engine and the ECS to the framework, complicated interaction networks such as this can be explored using FAVER’s architecture.

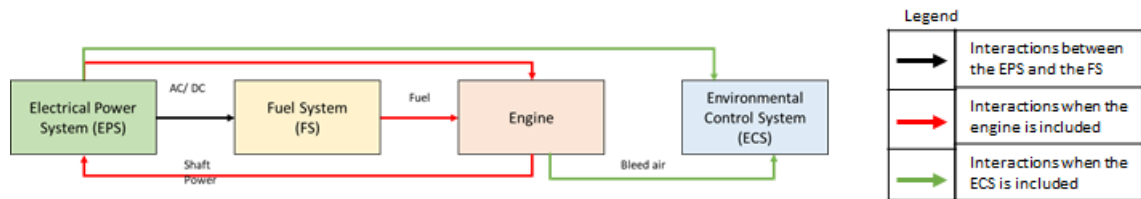


Figure 8-5 Main interactions between four aircraft systems

For this demonstration, the DTs for four aircraft systems and their system level diagnostics are developed as a part of background work [21], [22], as mentioned in section

8.2. [Figure 8-6](#) shows the activities carried out for the expansion of FAVER. This activity diagram shows the inclusion of the engine and the ECS into the framework, which already has the EPS and the FS from the previous work [3]. The generalized modules in the architecture diagram of FAVER in [Figure 8-4](#) are shown with more specifics in [Figure 8-6](#) relating to the four aircraft systems, such as their DTs, system diagnostics, and some of their fault modes.

The middle section of [Figure 8-6](#) shows the existing structure of FAVER with the FS and the EPS DTs, and their diagnostics set up, each isolating three fault modes and four fault modes respectively (as demonstrated in previous work [3]). The EPS and the FS DTs from the DT layer share their data for the system level diagnostics, which are integrated to FAVER's Reasoning layer through Domain facilitator, which makes the system level knowledge for the FAVER's expert system. It is to be noted that the DTs are not connected with the Reasoning layer; their data is simply used to develop the system level diagnostics by the Reasoning layer. For the expert system knowledge, the relationship matrix at vehicle level, and the fault attributes database are populated by experts based on the faults diagnosed by the system diagnostics. Along with this knowledge, a rule based approximate reasoning algorithm for fault isolation at the vehicle level is present in FAVER's Reasoning layer.

The left section of the activity diagram in [Figure 8-6](#) shows the activities carried out by the Central facilitator through the CBR methodology. The Central facilitator fetches the symptom vectors from the user, dispatches the information to the Reasoning layer, retrieves results of system level diagnostics and vehicle level rule based approximate reasoning. The results are then output to the user/ engineer for abductive reasoning, and further updates and the new knowledge are retained in FAVER's expert system.

In expanding FAVER to include two more systems, the engine, and the ECS, the DTs for both these systems and their system diagnostics are required. The right section of [Figure 8-6](#) shows how FAVER accommodates the addition of the engine and the ECS into the framework, whose diagnostics can isolate four and three fault modes respectively. The knowledge of FAVER's expert system will have the relationship matrix grow to the size 4x4, and the fault attributes database with seven new rows, two attributes in each row. This results in a total of only 12 new cells to be updated in the vehicle level relationship matrix and 14 new cells to be updated in the fault attributes database by the

experts, as per (5-3). On the other hand, a traditional matrix would have grown in size to 14x14, experts having to update 147 new cells as per (5-2), hence not used in FAVER for its time-consuming process. The rule based approximate reasoning at the vehicle level does not need any changes for expansion of the framework; it can be applied ‘as-is’ to isolate the root causes and cascading effects of the interacting faults. In the top left portion of Figure 8-6, the Central facilitator is updated to take additional symptom vectors as inputs from the engine and the ECS system diagnostics. These are the only changes to be made to FAVER to accommodate new systems in its framework.

8.3.1.2 Adding new faults to existing systems diagnostics

It is not always possible to include all the faults occurring at the systems level. The systems DTs and diagnostics are updated frequently based on the newly gained knowledge. This subsection presents the steps to update the framework with newly added faults. Consider each of the four aircraft system diagnostics in the currently updated framework, and the requirement to add more faults. The activity diagram in Figure 8-7 reflects this scenario with new fault modes shown in the white background under the list of faults diagnosed in each system⁷. Figure 8-7 shows the transformation required by FAVER to expand and accommodate new faults. To add the new faults to the systems diagnostics in the framework, considering the systems diagnostics are updated with the functions to isolate the latest fault modes, the only change that is required in FAVER is the addition of new faults in the fault attributes database, with their *macro_cause* and *macro_effect* values updated. It should be noted that in the activity diagrams (Figure 8-6 and Figure 8-7) the activities between the Central facilitator and the Reasoning layer do not change. This is because FAVER’s structure is robust due to modularity, hence these expansions do not require any further changes.

8.3.2 Flexibility in FAVER

The modular architecture of FAVER ensures flexibility for the integration of the DTs. Every system DT is encapsulated and connected to the DT Facilitator, which will control its operation and connectivity with the other DTs and the broader framework. The DT

⁷ When new faults are added to the system diagnostics, they are independent of the changes to other systems diagnostic functions or algorithms used. They are simply integrated to FAVER's Fault attributes database.

Facilitator transfers engineering data between DTs by non-dimensionalising the macro interactions shown in [Figure 8-5](#), allowing differently scaled systems to co-exist. As a small example of this, the engine used in this paper for demonstration is a Pratt & Whitney JT9D, but the fuel system is a benchtop experimental rig [18]. The system itself could be represented by a physics-based or function-based model, or a data driven DT, or it could also be a Hardware-in-the-loop (HIL). There are no changes required to be made in the representation model for its integration into FAVER since it is not directly hardwired with any other DTs. For example, the FS used for demonstration is HIL, whereas the EPS and the engine are developed in MATLAB SIMULINK, and the ECS DT is developed in MATLAB SIMSCAPE. The behaviours of these aircraft systems during simulations are translated into knowledge for vehicle level health assessment in FAVER by the DT's system level diagnostics. This feature of FAVER promotes simplicity and prevents redundant V&V of the newly established interaction models, which would have been the case if the DTs were hardwired [3].

The diagnostic modules of the system DTs can also be of any type (e.g. model based, data-driven or hybrid). The number of fault modes or the types of faults, like those with micro or macro effects, do not matter to FAVER. Any type of system level diagnostic can be integrated into FAVER by developing a MATLAB function on top of the diagnostic module, which can be accessed by FAVER's Domain facilitator, ensuring flexibility in the framework. The diagnostics of the FS, the engine, and the ECS are made of data-driven machine learning algorithms [22]. In contrast, the diagnostics of the EPS is made of hybrid reasoning, comprising of ANFIS and causal reasoning [21].

Across the industry, the types of DTs and diagnostics developed for health monitoring of aircraft systems are not uniform. Hence, with a rigid framework, it would be difficult to carry out vehicle level reasoning requiring rigorous background work of converting every system DT and its diagnostics into a uniform format. Thus, FAVER's flexibility ensures vehicle level health reasoning of aircraft systems with their representative DTs and system diagnostics in 'as-is' conditions, irrespective of their scales and formats.

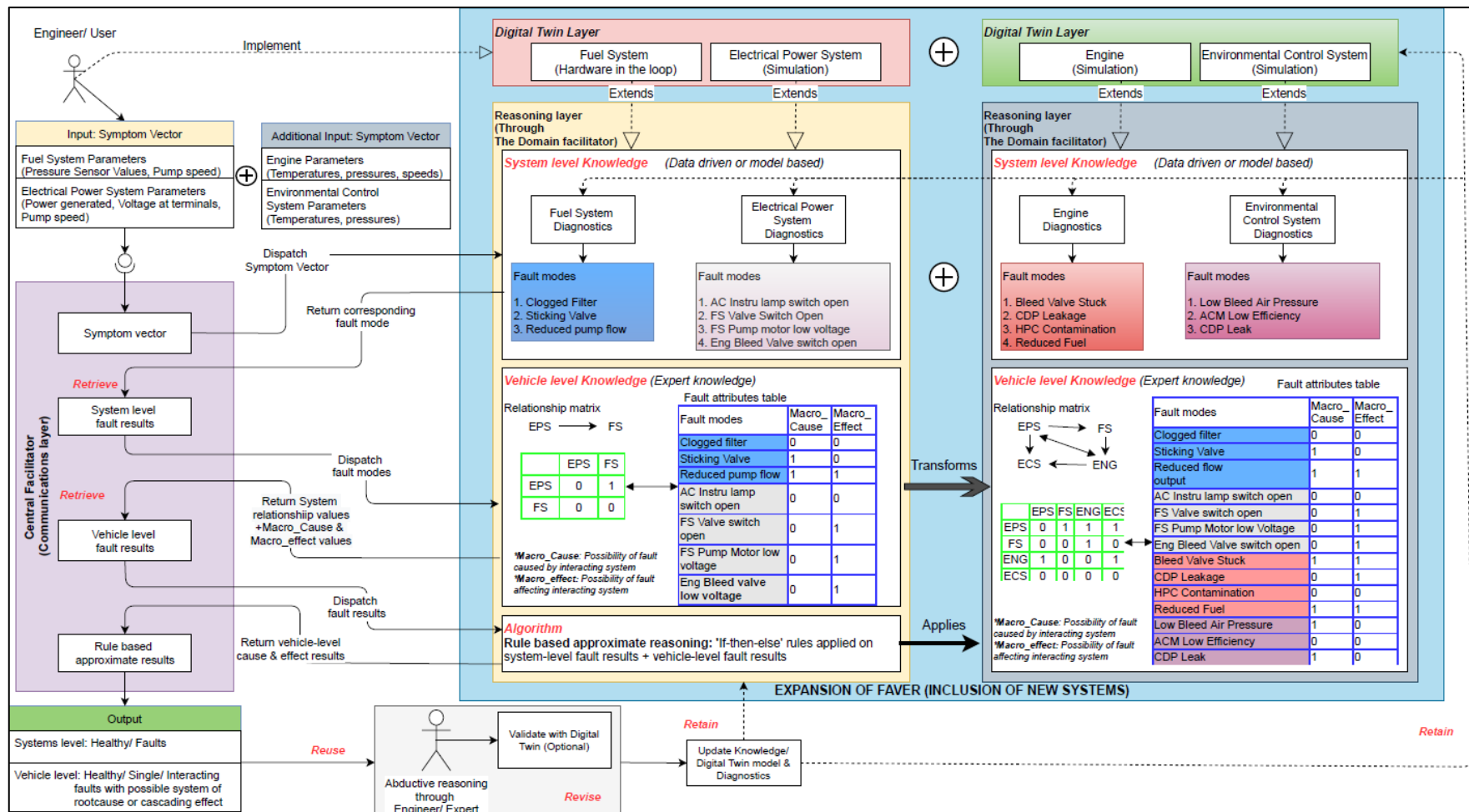


Figure 8-6 Activity diagram for expansion of FAVER to include two new aircraft systems

8.4 Demonstration Use cases

Section 8.3 presented the requirements and steps to be followed to expand the framework and include a new system (previously unknown) to FAVER. Once a new system is integrated into the framework, the changes made to FAVER have to be tested for integration loopholes and verified for its functionality. Adding a new system to the framework increases the number of use cases that can be generated through various combinations of healthy and faulty scenarios. In this case, having included the engine and the ECS in the framework several combination of use cases such as, all four systems healthy, all four systems with single faults, interaction faults involving four systems, interaction faults in two systems and two systems with single faults, and so on, can be generated. For the sake of brevity in this paper, only a few combinations of use cases are demonstrated to show FAVER's ability to isolate single and interacting faults in a newly added system and solve rising ambiguities.

The interactions between the EPS, the FS, the engine, and the ECS in the aircraft are substantial from the operations, maintenance, and safety perspective [25]. Accident events, like the engine rollback on a B777 as a result of fuel starvation [5], [6], the engine inflight shutdown in an experimental aircraft due to a burnt electric fuel boost pump [26], and an emergency descent due to cabin pressurization resulting from engine bleed air system failure [27], show that despite the rarity, such worst-case scenarios do occur. Moreover, with the current trend moving towards the more-electric engine and more-electric aircraft, the influence of the EPS over the FS, the engine, and the ECS is growing significantly [21], [25]. Hence, use cases similar to these accident scenarios are used for demonstration. While each of the aircraft system diagnostics developed as background work for FAVER can isolate several fault modes, only a few of them are used for demonstration in this paper. The expanded version of FAVER with four aircraft systems and system diagnostics isolating five faults each, as shown in [Figure 8-7](#) is used for demonstrations in this section. The details of the faults modes used here for demonstration and the corresponding behaviours of the respective system DTs can be found in the previous chapters 6 and 7 [21], [22]. They are not detailed here for the sake of brevity. The step-by-step Central facilitator activities along with the rule based approximate reasoning algorithm is provided in [Appendix B](#).

8.4.1 New system (previously unknown): Single and Interaction faults isolation

8.4.1.1 Use Case Scenario 1

In line with the incidents of engine rollback caused by a blockage in the fuel system [5], [6], and engine inflight shutdown due to a burnt fuel boost pump resulting in fuel starvation [26], consider a scenario where a low voltage supply fault arises in the EPS and affects the FS pump motor, which in turn could reduce fuel supply to the engine.

Figure 8-8 shows the development of the use case scenario as four independent simulation runs as listed below. Every stage of fault propagation is one simulation run with the vehicle symptom vector given by the user, which are generated separately through the DTs simulations. The procedure of developing vehicle symptom vector and the screenshots of the steps carried out by the Central facilitator program can be seen in Appendix B.

Simulation run 1: All systems are healthy

Simulation run 2: Single fault (*FS Pump Motor Low Voltage*) arising in the EPS and rest of the systems are healthy

Simulation run 3: *FS Pump Motor Low Voltage* from the EPS affecting the FS triggering *Reduced Pump flow* fault, and the other two systems are healthy, and

Simulation run 4: *Reduced Pump flow* fault in the FS cascading to the engine triggering *Reduced input from the FS* fault, and the ECS remains healthy.

The results of above-mentioned runs are aggregated as one figure (Figure 8-8) to show how FAVER's reasoning will produce results during each stage of the fault propagation. The left side of each run shows the current state of systems through diagrams. In these diagrams, the healthy interactions in simulation runs are indicated by green lines, interactions having the potential to cascade faults are indicated by blue lines, and the interactions that have cascaded the effects are shown as red lines, for clarity. The use case scenario starts with the simulation run 1, where four systems are healthy. In the next runs, the simulations are run with fault injected in the systems.

Simulation run 1 of Figure 8-8 shows the first stage where all the systems are healthy and diagnosed as such. In simulation run 2, the vehicle symptom vector consists of *FS Pump Motor Low Voltage* fault in the EPS, and other systems at healthy states. The vehicle symptom vector is input to the Central facilitator and the fault *FS Pump Motor*

Low Voltage detected through the Domain Facilitator's System diagnostic algorithms (Figure 8-4). The fault is detected as being in the EPS, all other systems being judged healthy by the Central facilitator. This is shown by the first six lines of the Central facilitator result in Figure 8-8. Physically, this low voltage input to the FS AC motor pump could result in reduced rotor speed and torque of the pump (simulation details can be found in previous work [21]). This fault would essentially affect the fuel pump as indicated by the blue colored line in Figure 8-8. FAVER detects this potential propagation by passing the detected fault into the Domain Knowledge part of the Reasoning layer. Here FAVER's rule based approximate reasoning uses the relationship matrix to identify the related systems to the EPS and the fault attributes database to identify if the fault *FS Pump Motor Low Voltage* has macro_cause or macro_effect. It then warns of a possible cascading effect to the FS through the Central facilitator, last three lines in Figure 8-8. Having stepped through this case in detail, the other cases are described more briefly.

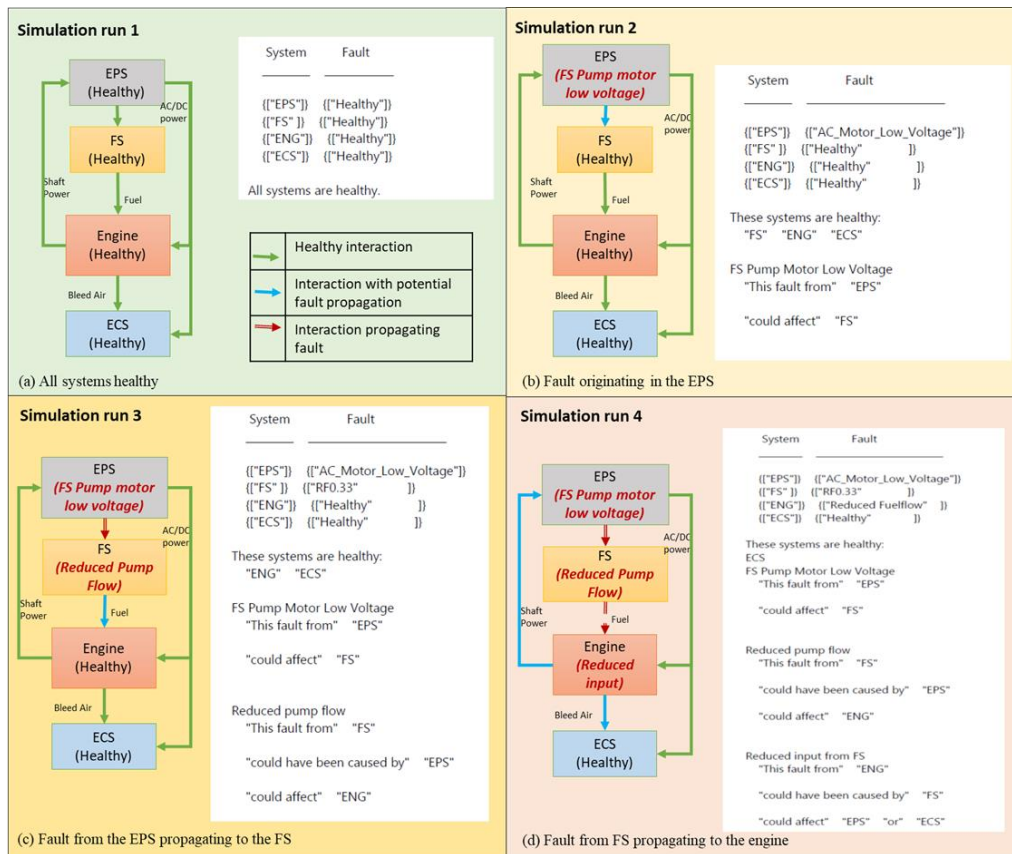


Figure 8-8 Use case scenario of fault originating in the EPS propagating to the FS and the engine

Simulation run 3 shows the next potential stage of fault propagation, where the fault

FS Pump Motor Low Voltage affects the FS. Here, the vehicle symptom vector input consists of *FS Pump Motor Low Voltage fault* in the EPS, and *Reduced Pump flow* in the FS, and the other two systems are healthy. [22]. The symptom vector is input into the Central facilitator and results in both the EPS and the FS being flagged as faulty (RHS of simulation run 3). Further, the possibility of *Reduced Pump flow* from the FS affecting the engine is identified by the rule based approximate reasoning with the expert systems knowledge in Domain Knowledge. This also leads to the tracking of the potential root cause to the EPS, as well as predicting its effect over the engine, which is reported back to the Central facilitator.

In the final stage of the propagation of this fault, the *Reduced Pump flow* fault from the FS provides a lower fuel flow to the engine; this behaviour is translated to the engine DT by the *Reduced input* fault mode [22]. The resulting symptom vector is input to the Central facilitator, and simulation run 4 shows the result. The Diagnostic algorithms return a fault in 3 systems to the Central facilitator, with only the ECS being healthy. FAVER's reasoning, in the Domain knowledge, tracks the cause of the engine's fault to the FS, and then maintains the root cause with the EPS. It should also be noted that the engine's cascading effect warning is given for the EPS and the ECS, taking into consideration the shaft power provided by the engine to the EPS and the bleed air supply to the ECS respectively (highlighted by a blue line in the left side diagram of simulation run 4 of [Figure 8-8](#)).

8.4.1.2 Use Case Scenario 2:

In 2008, an Airbus A319 aircraft made an emergency descent of 11000 ft, which resulted from the wrong engine bleed air system being chosen for fault isolation by the pilot. The faulty engine bleed air system chosen failed during flight and, because the pilots inadvertently switched off the other bleed air system, an emergency descent had to be made because of cabin depressurization, and oxygen masks were deployed for the passengers [27]. A use case is developed similar to this accident scenario. A *Bleed Valve Stuck* fault is triggered in the engine DT, and corresponding behaviour in the ECS DT is observed by triggering the *Low Bleed Air Pressure* fault through simulations. The results from the system level diagnostics are used for vehicle level reasoning in FAVER, by providing their symptom vectors to the Central facilitator, to check if FAVER's reasoning can identify the root cause of these faults.

Figure 8-9 shows the development of the bleed valve fault through stages in the same way as for the previous use cases. The Central facilitator output (white box) expands as the fault scenario develops. In simulation run 1, the *Bleed Valve Stuck* fault is found only in the engine, and the ECS is healthy, and so are the other systems. Here, FAVER's reasoning shows a warning of the possibility of a fault cascading from the engine to the ECS at the earliest sign of a fault, thus triggering a watch on the ECS for potential fault propagation. In simulation run 2, the *Bleed Valve Stuck* fault is propagated to the ECS by translating its effect of reduced bleed air output to the ECS DT, leading to *Low Bleed Air Pressure* fault. In this case, the Central facilitator identifies the root cause of the *Low Bleed Air Pressure* fault in the ECS to be originated from the engine.

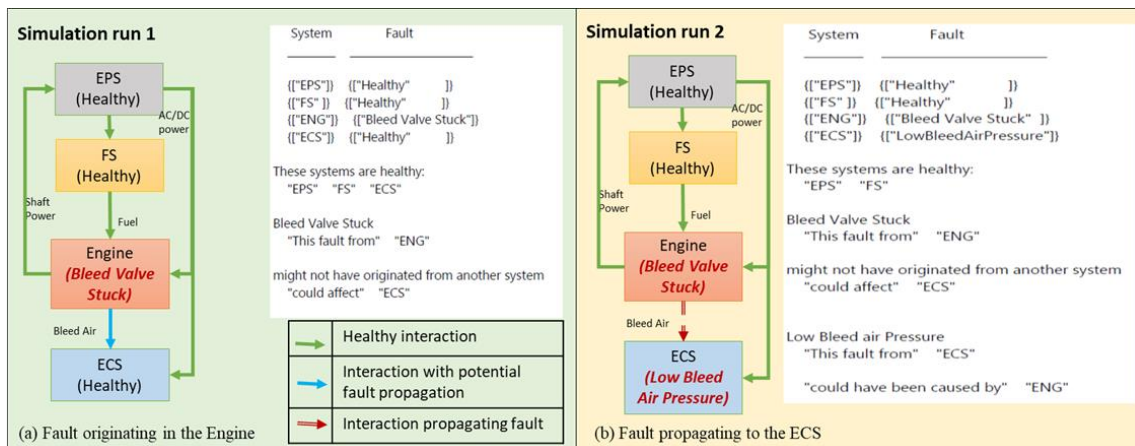


Figure 8-9 Use case scenario of fault originating from the engine and cascading to the ECS

Thus, in both the use case scenarios, at every stage of fault propagation, FAVER's high level reasoning is able to isolate the root cause of the interacting faults and warn about the cascading effects at the same time. This, when implemented for industrial applications, can be used for preventing faults propagating and for isolating their root causes during troubleshooting in maintenance.

8.4.2 Cases of multiple faults

It is quite common for faults from multiple systems to be flagged simultaneously by systems diagnostics in a complex vehicle like an aircraft. These faults could be independent of each other or may occur due to their interactions with other systems, and they give rise to ambiguity. One of the objectives of reasoning is to resolve this ambiguity by clearing these ambiguous scenarios by identifying whether they are simultaneous

single faults or by identifying which fault is the root cause in case of interaction faults. Figure 8-10 shows four scenarios where FAVER can differentiate interacting faults from simultaneous single faults, clearing ambiguity. Unlike the previous subsection, where Figure 8-8 and Figure 8-9 show the stages of fault propagation of one use case scenario, the cases shown in Figure 8-10 are four different cases.

Case 1 in Figure 8-10 shows the scenario of an interaction fault with three of the four systems involved while the FS has a single fault simultaneously. The interaction fault in Case 1 is an extension of use case scenario 2, where the *Bleed Valve Stuck* fault in the engine results in supplying bleed air at lower pressure to the ECS (*Low Bleed Air Pressure*). In this case, the fault in the engine is assumed to have been caused by *Eng Bleed Valve switch stuck open* fault mode from the EPS. These fault modes are generated through simulations from the engine DT, ECS DT, and EPS DT [21]. Meanwhile, fault mode *Clogged Filter* is chosen for the FS and it is assumed not to affect any other system. The vehicle symptom vector containing all these health states are given to the Central facilitator, and all four system diagnostics are flagged for faults as a result, giving rise to an ambiguous scenario. The result of FAVER's reasoning, in this case, can be seen in case 1 of Figure 8-10. The rule based approximate reasoning is able to distinguish the *Clogged Filter* fault in the FS as a single fault from the interactive fault involving the other three systems, identifying the root cause to have originated from the EPS and warning of the cascading effects to the ECS from the engine. It can be noticed from case 1 of Figure 8-10, that the *Low Bleed Air Pressure* fault in the ECS is suspected of having originated either from the engine or the EPS. The reason behind this result is discussed in the next subsection.

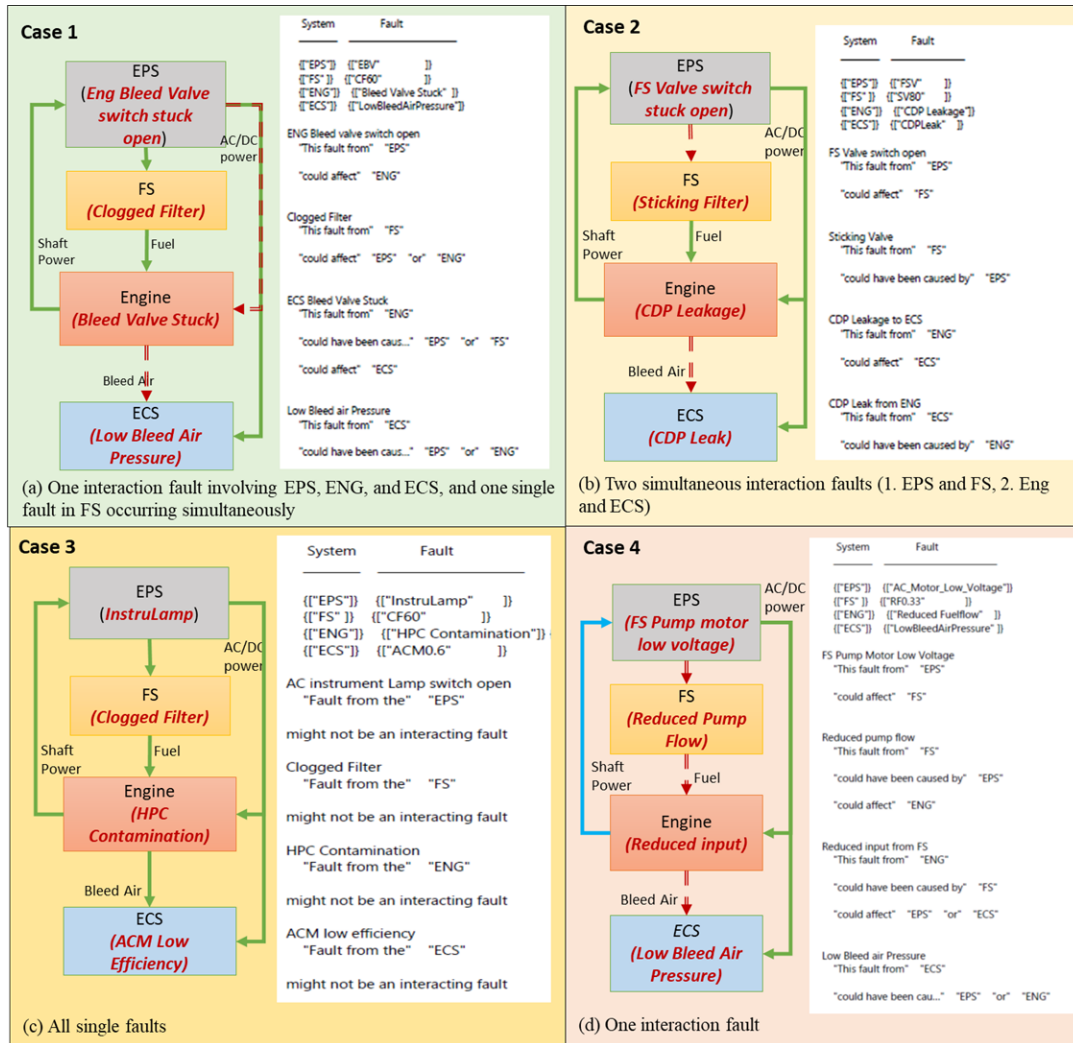


Figure 8-10 Different multiple fault scenarios (some resulting in ambiguity) reasoned by FAVER

Another scenario is tested, as shown in case 2 in Figure 8-10, where two simultaneous, interacting faults are injected in the system DTs. The fault mode, *FS Valve switch stuck open* in the EPS is assumed to cause *Sticking Valve* fault mode in the FS. Simultaneously, a leak in the Customer Discharge Pressure (CDP) of the bleed air duct in the engine (*CDP Leakage*) provides less bleed air to the ECS. This behaviour is translated to the ECS DT and is detected by the ECS diagnostics as *CDP Leak*. Case 2 in Figure 8-10 shows that FAVER's reasoning can differentiate both of these interacting faults and suggest their possible root causes and cascading effects.

In case 3 of Figure 8-10, all four systems have faults that are assumed to have only local effects and do not affect any other systems. *InstruLamp* fault mode affects the lighting in the cockpit instrument panel [21]. *Clogged Filter* fault mode in the FS results in increased pump speed to meet the fuel flow demand but does not affect any other

connected system. *HPC Contamination* is the fault mode as a result of contamination in the high pressure compressor in the engine, reducing its efficiency. The *ACM low efficiency* is the reduced mechanical efficiency of the air cycle machine in the ECS [22]. Case 3 in Figure 8-10 shows that FAVER's reasoning is capable of identifying all four single faults that occurred simultaneously, not confusing with interaction faults.

In case 4 of Figure 8-10, one interaction fault starting from the EPS affects all the other systems. This use case is a projection of the use case scenario 1 demonstrated in the previous subsection and seen in Figure 8-8. While in the previous subsection, the ECS is shown as healthy, with the potential of *Reduced input from FS* fault in the engine affecting the ECS, in this scenario, it is assumed that the fault from the engine results in *Low Bleed Air Pressure* fault in the ECS. In this case, FAVER's reasoning is able to identify the root causes of fault for every system and its cascading effect. Similar to simulation run 1 in Figure 8-8, the fault in the ECS in case 4 is suspected to be originated from either the EPS or the engine. The reason behind this is discussed in the following subsection.

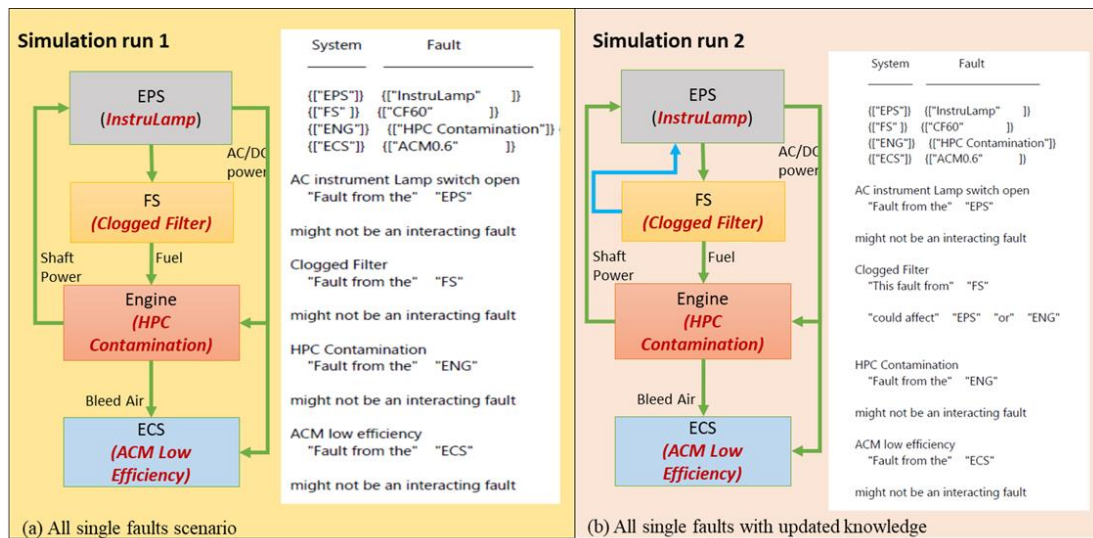
8.5 Evaluation of FAVER

8.5.1 Pros and Cons of using FAVER's reasoning

This subsection discusses the pros and cons of using FAVER's reasoning through a few use cases.

8.5.1.1 Updating FAVER's knowledge

One of the strengths of FAVER is the ease of updating the expert systems knowledge to retain and reflect the new learnings regarding the fault modes at the systems level and



their propagation behaviour at the vehicle level. For example, the *Clogged Filter* fault mode in the FS is assumed not to affect any connected system other than the increased pump speed to meet the fuel flow demand in the previous demonstrations. However, in practice, the resultant increased pump speed affects the power demanded from the EPS. This knowledge could be reflected by making only two changes to FAVER's expert system: i) to update the relationship between the FS and the EPS in the relationship matrix, and ii) to update the *macro_effect* attribute of the *Clogged Filter* mode from '0' to '1' in the fault attributes database. Once these changes are carried out, FAVER's reasoning is able to update the cause and effect of this fault mode automatically. This can be seen in [Figure 8-11](#), where the all single fault scenario from simulation run 1 is updated with new knowledge concerning the relationship between the FS and the EPS in simulation run 2. The difference can be seen in the results generated by the Central facilitator (white boxes), wherein the *Clogged Filter* mode is suggested as not an interacting fault in simulation run 1 but is suggested to affect the EPS or the engine in simulation run 2.

Figure 8-11 Updated knowledge scenario in FAVER

8.5.1.2 Assigning priority in FAVER's reasoning

In a fault case like the *FS Pump Motor Low voltage* from the EPS triggering a series of cascading faults across multiple systems, the EPS triggers the fault that eventually affects the engine, and the engine's reduced power affects the shaft power provided to the EPS. Such cyclic events cannot have clear root cause predictions without time stamps of the fault occurrences. Simulation run 1 in [Figure 8-12](#) shows such a scenario where FAVER's reasoning shows that the engine affects both the ECS and the EPS, while the EPS is the original trigger. To break this cycle, priority can be assigned either for the root cause or for the cascading effects in reasoning. While simulation run 1 shows all possible systems that could be affected by the engine because of reduced output from the engine, simulation run 2 in [Figure 8-12](#) removes the EPS from the cascading effect since it is also a possible root cause of the fault in the engine. This case is an example of the ability of FAVER's reasoning to adapt to special cases such as the cyclic events found in this interaction fault.

It can be observed from [Figure 8-12](#) that the root cause of *Low Bleed Air pressure* in the ECS is listed as the EPS or the engine in the simulation run 1, and as only the engine

in the simulation run 2. The approximate reasoning in FAVER can be finetuned to provide correct root cause by adding ‘clues’ in the description of the fault (*Low Bleed air pressure from ENG*), which could be text mined to provide contextual reasoning.

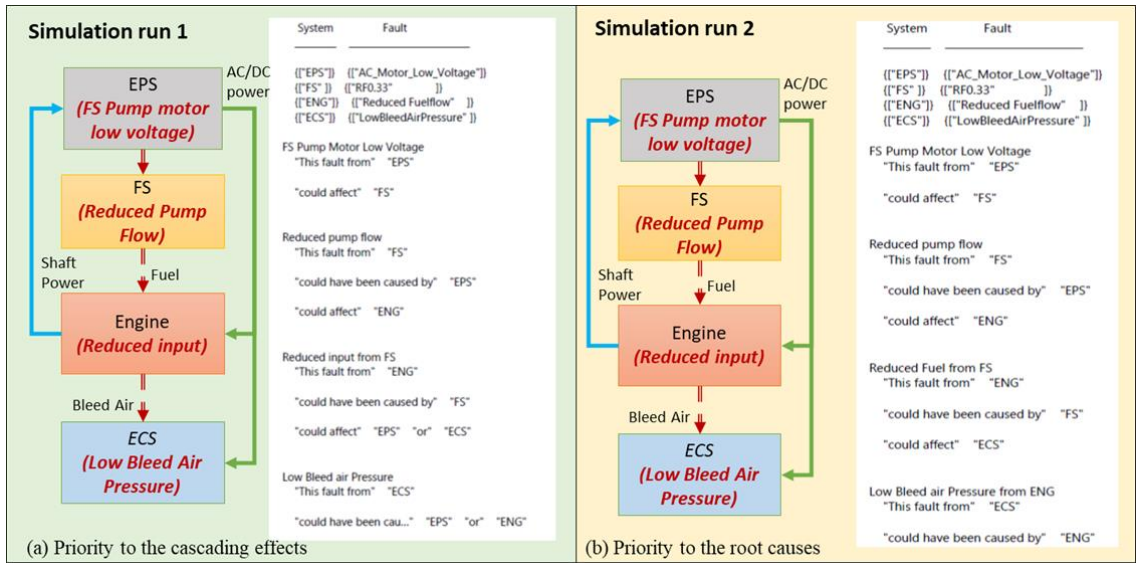


Figure 8-12 Assigning priorities in FAVER's reasoning

8.5.1.3 Cost of using the novel combination of knowledge

In cases 1 and 2 of Figure 8-13, while the *Low bleed air pressure* fault is cascading from the engine to the ECS, FAVER's reasoning suggests it could either be from the EPS or the engine. The root cause of fault in the ECS is from the engine, and that of the engine is from the ECS. But, the root cause of the fault in the ECS is predicted to have been from the engine or the EPS. This is the cost of using the combination of fault attributes database with the vehicle level relationship matrix in place of the traditional relationship matrix in the expert system. The accurate root causes in these cases are not predicted, in favour of less complex and less time-consuming knowledge format, and are compensated by abductive reasoning, where the experts infer the actual root cause from the choices provided in the Central facilitator. This could also be corrected by adding text mining or contextual reasoning to FAVER's reasoning. However, this method does have shortcomings, since text mining could confuse the reasoning when texts of multiple systems are present in the fault description.

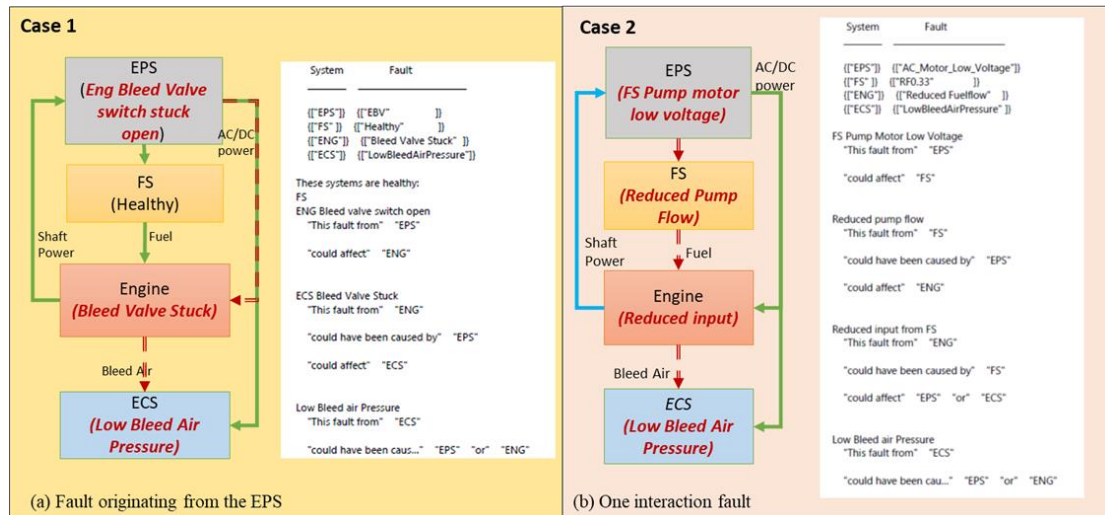


Figure 8-13 Cost of replacing traditional relationship matrix in FAVER

8.5.1.4 Cost of misclassification in system level diagnosis

As FAVER's vehicle level reasoning is built upon the system level diagnostic functions, the misclassification rates at the system level are inherited to the reasoning at the vehicle level. Hence, the misclassification rates are additive for single faults and multiplicative for interaction faults for FAVER's reasoning at the vehicle level.

8.5.2 Use cases demonstration with Neural Networks

To evaluate FAVER's expert system knowledge with the combination of abductive and rule based approximate reasoning, a comparative study with data-driven reasoning is performed, using a few use case scenarios trained using a Neural Network. The DT layer in FAVER is used for the neural network for this comparison study. A synthetic dataset is prepared for supervised learning by simulating three DTs (EPS, FS, and engine) for the nine scenarios listed in Table 8-1. The details on fault modes used can be found in these references ([21], [22]).

Table 8-1 Use case scenarios for neural network training

Scenarios	Electrical System	Power	Fuel system	Engine
All Healthy	Healthy		Healthy	Healthy
Single Fault scenario	Healthy		Clogged Filter @40% degradation	Healthy
Single Fault scenario	AC Instrument Panel Lighting fault		Healthy	Healthy
Single Fault scenario	Healthy		Healthy	LPC contamination @ 7% degradation

Interaction fault with 2 systems	FS_Nozzle_Stuck	Clogged Nozzle @ 20% degradation	Healthy
Interaction fault with 2 systems	Bleed_Valve_stuck	Healthy	Bleed_Valve_stuck @ 60 deg
2 simultaneous single faults	AC Fluorescent Lighting fault	Blocked Flow Meter @ 60%	Healthy
2 simultaneous single faults	AC Fluorescent Lighting fault	Healthy	HPT Blade broken with 1% loss of efficiency
Interaction fault with 3 systems	EPS @ 95% input shaft speed	Reduced flow @ 375 rpm	Reduced flow with 80% input

The neural network was trained using the Levenberg-Marquardt algorithm [28] in MATLAB. The input symptom vectors are made up of 61 input parameters directly combined from the three DTs (6 parameters from FS fuel rig simulation, 39 parameters from the engine T-MATS simulation, and 16 parameters from EPS MATLAB simulation). The neural network is trained to flag faults at the system level, along with flags for single and interaction faults and the presence of ambiguity at the vehicle level, resulting in a total of seven output classes as shown in Figure 8-14. The cases are formulated as a classification problem, with the binary output results showing if a symptom vector belonged to an output class or not. The trained neural network provided the results with 100% accuracy on the test dataset. Further customization of the neural network is deemed unnecessary at this point.

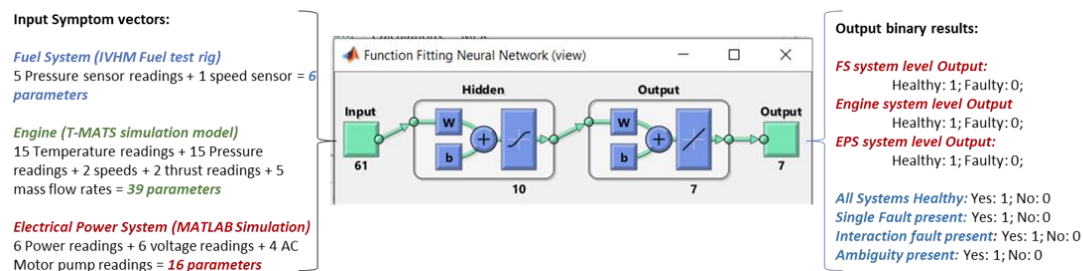


Figure 8-14 Neural network fitting to isolate faults at the vehicle level

Advantages of using data-driven reasoning:

The data-driven reasoning to identify if a symptom vector belongs to an output class using neural networks has certain advantages over FAVER's reasoning. The important one is its independence of system diagnostic functions. The output classes trained in this demonstration are only at the high level, but the in-depth supervised learning would have made the neural network isolate the fault modes of the systems as well, without developing diagnostics for each system. Also, the results produced by the data-driven reasoning are more accurate than FAVER's reasoning, due to the approximate reasoning

and expert system knowledge components of the latter. With the availability of more data and time on curating the datasets, the neural networks could be trained to even isolate root causes and cascading effects of interacting faults. While the neural network can process the results for multiple sets of symptom vector at a given time, it does consume more time and much heavier computational capacity; hence it is not compared with FAVER's computational time directly.

Advantages of using FAVER:

One of the main advantages of FAVER against the neural network is the clarity in the outputs produced. While the neural network depends upon engineering knowledge and DTs only for curating its datasets and produces results based on the learning the patterns from its data, FAVER's reasoning depends upon engineering knowledge from its datasets for system level diagnostics to its expert system knowledge at the vehicle level. This makes FAVER's process more transparent and trustworthy as against data-driven reasoning. Another advantage of using FAVER is its flexibility. FAVER's modular structure enables the integration of any type of simulation model and diagnostic functions into the framework. In contrast, the neural network can accept only data-driven inputs for its reasoning process. Besides, the process of adding a new fault or new system into the database is more straightforward in FAVER when compared to the data-driven reasoning, where every change requires careful consideration for readjusting its datasets.

To summarize, while FAVER's reasoning is less accurate and performs slower when compared to the neural network, its significant advantage lies in its modularity. Encapsulation of the DTs and the novel knowledge base of FAVER's expert system enable FAVER to be flexible and scalable. While the neural network has provided very accurate results and has better performance, it requires carefully curated datasets representing the fault scenarios, which consumes more time and effort for every new addition to its structure. In practice, FAVER has the advantage due to its transparency over the neural network's black box approach. The results produced by FAVER can be used directly by the maintainers, as they are clearly explainable and show the routes for troubleshooting. Whereas, the neural network can produce only the results, from which, the maintainers will not have the knowledge of how the fault occurred or why the action is needed, and the interpreting process creates more time-consuming work during maintenance.

8.6 Summary and Future work

To summarize, the following contributions were made in this paper.

- i) The working concept behind FAVER's framework, its architecture, and the previous work done to set up the framework for the demonstrations have been described.
- ii) FAVER's scalability is demonstrated on two fronts:
 - a. The ability to expand the framework to include a new system with minimal change to the framework is demonstrated using four aircraft systems: i) the EPS, ii) the FS, iii) the Engine, and iv) the ECS.
 - b. The ability to expand to include new faults to the systems with minimal change to the framework by adding additional faults to all four systems.
- iii) Real aircraft incidents were adapted for use case scenarios, and FAVER's ability to isolate faults for single faults, interaction faults and to resolve ambiguity are tested.
- iv) Evaluation of FAVER was carried out in two stages:
 - a. The advantages and disadvantages of using FAVER's reasoning were demonstrated with examples.
 - b. FAVER's reasoning and data-driven reasoning (neural networks) were compared, and the pros and cons were discussed.

The framework is currently in its initial stage of isolating faults, identifying root causes, and cascading effects with a basic set of assumptions. FAVER can be further expanded by adding multi-physical systems like avionics, landing gear, and controls. False alarms and incorrect sensor readings, as well as unsteady state scenarios, can also be investigated. While FAVER's expert system helps maintain its trustworthiness and transparency, a hybrid of expert systems with data-driven methods can increase the dependency of its knowledge base and derive more information from existing data, rather than to rely only on experts. With its unique features and robust framework, FAVER opens many avenues for further exploration in vehicle level health reasoning for IVHM systems.

References

- [1] C. M. Ezhilarasu, Z. Skaf, and I. K. Jennions, "The application of reasoning to

- aerospace Integrated Vehicle Health Management (IVHM): Challenges and opportunities,” *Prog. Aerosp. Sci.*, vol. 105, no. January, pp. 60–73, 2019.
- [2] C. M. Ezhilarasu, Z. Skaf, and I. K. Jennions, “Progress towards a Framework for Aerospace Vehicle Reasoning (FAVER),” *Proc. Annu. Conf. PHM Soc.*, vol. 11, no. 1, pp. 1–9, 2019.
 - [3] C. M. Ezhilarasu and I. K. Jennions, “Developing the architecture of FAVER (a Framework for Aerospace Vehicle Reasoning) using a System-of-Systems methodology,” 2020.
 - [4] (Federal Aviation Administration), “Fact Sheet – General Aviation Safety,” 2018. [Online]. Available: https://www.faa.gov/news/fact_sheets/news_story.cfm?newsId=21274. [Accessed: 26-Nov-2018].
 - [5] P. A. Sleight and R. D. G. Carter, “AIRCRAFT ACCIDENT REPORT 1/2010,” 2014.
 - [6] Aero News Network, “NTSB Investigates B777 Uncommanded Engine Rollback,” 2008. [Online]. Available: <http://www.aero-news.net/index.cfm?do=main.textpost&id=7cba7571-d0e2-4736-a08b-2c24fcd7f225>. [Accessed: 29-Sep-2020].
 - [7] K. Conradi, “AAIB Annual Safety Report 2015,” 2015.
 - [8] Honeywell, “Health and Usage Monitoring System (HUMS),” 2018. [Online]. Available: <https://aerospace.honeywell.com/en/learn/products/health-and-usage-monitoring>. [Accessed: 21-Aug-2020].
 - [9] Safran, “Health Monitoring for military aircrafts.” [Online]. Available: <https://www.safran-electronics-defense.com/aerospace/military-aircraft/propulsion-system-solutions/health-monitoring-military-aircraft>. [Accessed: 21-Aug-2020].
 - [10] G. Bastard, J. Lacaille, J. Coupard, and Y. Stouky, “Engine Health Management in Safran Aircraft Engines,” in *Proceedings of the Annual Conference of the*

- Prognostics and Health Management Society, PHM*, 2016, vol. 2016-Octob, pp. 101–108.
- [11] Meggitt, “Engine health and condition monitoring,” 2018. [Online]. Available: <https://meggittsensing.com/aerospace/product/engine-health-and-condition-monitoring/>. [Accessed: 21-Aug-2020].
 - [12] Meggitt, “Health and Usage Monitoring (HUMS).” [Online]. Available: <https://meggittsensing.com/aerospace/product/helicopter-hums-to-prevent-failures-increase-safety-and-reduce-costs/>. [Accessed: 21-Aug-2020].
 - [13] V. N. Divakaran, R. M. Subrahmanya, and G. V. V. Ravikumar, “WHITE PAPER INTEGRATED VEHICLE HEALTH MANAGEMENT OF A TRANSPORT.” Infosys, 2018.
 - [14] Qualtech Systems Inc, “TEAMS-RT,” 2020. [Online]. Available: <https://www.teamqsi.com/products/teams-rt/>. [Accessed: 21-Aug-2020].
 - [15] P. M. Papadopoulos, V. Reppa, M. M. Polycarpou, and C. G. Panayiotou, “Distributed Diagnosis of Actuator and Sensor Faults in HVAC Systems,” *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 4209–4215, Jul. 2017.
 - [16] M. Daigle, A. Bregon, and I. Roychoudhury, “Qualitative Event-based Diagnosis with Possible Conflicts Applied to Spacecraft Power Distribution Systems*,” *IFAC Proc. Vol.*, vol. 45, no. 20, pp. 265–270, Jan. 2012.
 - [17] S. C. Ofsthun and S. Abdelwahed, “Practical applications of timed failure propagation graphs for vehicle diagnosis,” in *2007 IEEE Autotestcon*, 2007, pp. 250–259.
 - [18] E. O. Nwadiogbu, D. Mylaraswamy, S. Menon, H. C. Voges, and G. Hadden, “Vehicle Health Monitoring System Architecture for Diagnostics and Prognostics Disclosure,” US008346429B2, 2013.
 - [19] B. S. López, A. M. Siddiolo, P. P. Adhikari, and M. Buderath, “A Bayesian paradigm for aircraft operational capability assessment and improved fault diagnostics,” in *European Conference of the Prognostics and Health Management*

Society, 2016, pp. 1–14.

- [20] C. M. Ezhilarasu, Z. Skaf, and I. K. Jennions, “Understanding the role of a Digital Twin in Integrated Vehicle Health Management (IVHM),” in *IEEE International Conference on Systems, Man and Cybernetics (SMC)*, 2019, pp. 1484–1491.
- [21] C. M. Ezhilarasu and I. K. Jennions, “A System-Level Failure Propagation Detectability Using ANFIS for an Aircraft Electrical Power System,” *Appl. Sci.*, vol. 10, no. 8, p. 2854, 2020.
- [22] C. M. Ezhilarasu, Z. Skaf, and I. K. Jennions, “A Generalized Methodology for Diagnosis of Multiple Aircraft Systems,” 2020.
- [23] J. W. Chapman, T. M. Lavelle, J. S. Litt, and T. Guo, “A Process for the Creation of T-MATS Propulsion System Models From NPSS Data,” 2014.
- [24] I. Jennions, F. Ali, M. Esperon Miguez, and I. Camacho Escobar, “Simulation of an Aircraft Environmental Control System,” *Appl. Therm. Eng.*, vol. 172, no. November 2019, p. 114925, 2020.
- [25] T. Tokarski and A. Gębura, “Diagnostics and Technical Status Monitoring of Electric Fuel Pumps for Aircrafts,” *Res. Work. Air Force Inst. Technol.*, vol. 33, no. 1, pp. 237–252, 2013.
- [26] National Transportation Safety Board, “NTSB Identification: CEN16LA219,” 2016. [Online]. Available: https://www.nts.gov/_layouts/nts.aviation/brief.aspx?ev_id=20160614X54845&key=1. [Accessed: 01-Sep-2020].
- [27] South African Civil Aviation Authority, “Aircraft Incident Report,” 2008.
- [28] Mathworks, “trainlm,” 2020. [Online]. Available: <https://www.mathworks.com/help/deeplearning/ref/trainlm.html;jsessionid=3ad5f684b2f37dcc87231f5aa45a>. [Accessed: 06-Sep-2020].

9 Summary & Future works

Fault propagation between the systems in an aircraft leads to complicated troubleshooting processes, leading to unexpected downtime, prolonged maintenance, and sometimes, aircraft accidents. This research focused on developing a framework, FAVER, in order to isolate faults propagating between various aircraft systems and to identify their root causes well in advance. The IVHM systems in the industry are mostly implemented at the systems level of the aircraft, considering the interactions between the subsystems within. The lack of emphasis on IVHM systems in the domain of vehicle level health monitoring is addressed in this research through this framework. FAVER simplifies high level reasoning in the IVHM systems, that will aid troubleshooting activities, especially in the presence of cascading faults and associated ambiguities, thus contributing to an efficient and safe flight.

This chapter summarizes this thesis and lists the salient contributions made during this three-year research period. Section 9.1 summarizes how this thesis has achieved the research objectives, followed by section 9.2, which highlights the contributions made in the thesis, in addition to those achieving the objectives of this research. Finally, section 9.3 presents some of the research areas to be focused on in the future. Appendix C has a Strength, Weakness, Opportunities, and Threat (SWOT) analysis for the proof of concept of FAVER.

9.1 Addressing the research aim and objectives

The overall aim of this research is to develop a framework, FAVER incorporating Digital Twins (DT), to demonstrate the interaction of multiple aircraft systems and a reasoning mechanism to investigate the scenarios of fault propagation between them. This research achieves this aim by fulfilling a number of objectives that are given below.

Objective 1:

A literature review to find the existing work and gaps in the following areas:

Evidence:

a. Examples for fault propagation across aircraft systems

A number of aircraft accidents that occurred due to fault propagation involving multiple systems were cited in Chapters 2, 4, 5, and 8. A couple of these accident scenarios were adapted in chapter 8, for the demonstration of FAVER's capability to isolate cascading faults and their root causes.

b. Health monitoring techniques developed at systems and vehicle level

c. Reasoning systems developed in academia and industries

Chapter 2 discusses in detail the various health monitoring techniques and systems developed at the systems level and vehicle level. In addition, based on the focus of the particular chapters, the literature review sections of chapters 4, 5, 6, 7, and 8 present various techniques used in both industry and academia. Achieving this objective helped in identifying the research gap in the IVHM systems, as well as, in choosing the reasoning systems and techniques to be used throughout this research.

d. Digital Twin (DT) applications in the field of IVHM

Chapter 3 discusses in detail the various applications of a Digital Twin (DT) in the field of IVHM. This objective helped in understanding the role that DT plays in the industry and how it can be used in FAVER. The integration of DT in an Open Standard Architecture for Condition Based Maintenance (OSA-CBM) influenced IVHM framework from this chapter helped the conceptualization of integration of system DTs in FAVER.

Objective 2:

To choose multiple interacting aircraft systems from a legacy aircraft, define boundary conditions with sufficient complexity.

Evidence: In order to implement the framework within the prescribed time limit, yet with enough sophistication, the use cases for demonstrations were framed in a way that could make use of existing aircraft system DTs without reinventing the wheel. Hence, there was a lookout for available sources of aircraft system DTs. Finally, the open source engine model provided by T-MATS, the copyrighted model SESAC for the ECS, and the IVHM lab fuel rig for the FS are adapted for their integration into FAVER. This exercise also enabled FAVER to be defined in such a way to include any representation of a system DT, at any scale, into the framework to match with the reality.

Objective 3:

To produce simulations of the chosen aircraft system DTs and to develop diagnostics for these systems to isolate a certain number of fault modes.

Evidence: Chapters 6 and 7 present the simulations of four aircraft system DTs, along with the process of developing standalone system level diagnostics. The EPS DT is developed in MATLAB SIMULINK with connections to the other aircraft systems. An ANFIS monitor combined with causal reasoning is developed for isolating both continuous and intermittent faults and is demonstrated to have classified all the faults accurately. For the other three aircraft systems, viz. the engine, the ECS, and the FS, the existing sources are used to produce simulations of healthy and faulty conditions. The OSA-CBM methodology is applied to develop data-driven diagnostics for each of these aircraft systems. These diagnostic functions were able to isolate faults that could be having both local and macro effects, with more than 95% accuracy. The results from the system level diagnostics are used for vehicle level reasoning in FAVER.

Objective 4:

To develop a reasoning module at the vehicle level that

- a. **contains the knowledge of both the system level diagnostics and the relationship between the systems,**
- b. **has a knowledge system that is easily scalable and updatable, and**
- c. **possesses reasoning methodologies and algorithms to isolate single and interacting faults at the vehicle level**

Evidence: FAVER consists of an expert system whose knowledge is made up of a combination of a vehicle level relationship matrix and a fault attributes database. Chapter 5 discusses the advantages of using such a combination for the expert system knowledge, as against a traditional relationship matrix. Also, chapters 5 and 8 show how FAVER's expert system can be easily updated and scaled to include new faults and systems into the framework. The activity chart in chapter 5 explains how the Case Based Reasoning (CBR) methodology is used for the activities between different modules in FAVER and how the rule based approximate algorithm is applied along with abductive reasoning to isolate cascading faults and identify their root causes.

Objective 5:

To demonstrate the ability of the FAVER that combines the power of reasoning with the capabilities of a Digital Twin (DT) to deduce and isolate the system interaction faults and their root causes in the following cases:

- d. isolate faults that affect both local systems as well as have macro effects on the interacting systems,**
- e. find ambiguity among the faults and resolve, and**
- f. expand the framework to accommodate systems that are previously unknown**

Evidence: Chapter 8 provides demonstrations for use cases adapted from accident scenarios. These demonstrations show how FAVER's blend of reasoning techniques could isolate interacting faults, identify their root causes, and warn of cascading effects well in advance. Few more examples also show how FAVER is able to resolve the ambiguity arising from faults being flagged simultaneously by distinguishing the interaction faults from simultaneously occurring single faults that do not affect other systems or two different interacting faults. Chapter 8 also provides demonstrations of expansion of FAVER by including the engine and the ECS into the framework that already had the EPS and the FS in place.

9.2 Contributions to knowledge

In the journey of this research, beginning from gap analysis, conceptualization, up to the successful demonstration of use cases adapted from real accidents, a number of significant contributions in this research, alongside fulfilling the aim and objectives.

- i) **A Framework for Aerospace Vehicle Reasoning:** The main contribution to knowledge from this thesis is FAVER, which is a system-agnostic framework and is capable of isolating cascading faults and their root causes at the vehicle level. FAVER also has the ability to accommodate a new system DT with any type of representation, scale, and system level diagnostics. Since the DTs are encapsulated in FAVER, the system DTs can be maintained and updated to recent models without affecting the other DTs in its interaction network. Due to these features, FAVER can be implemented in any IVHM system by aggregating/ updating recent system level

models and data and can be used for vehicle level health monitoring without much changes to the existing systems.

- ii) **FAVER's expert system:** The expert system in FAVER consists of a combination of vehicle level relationship matrices with fault attributes database. This form of knowledge can be easily updated, edited, and maintained, as against the traditional relationship matrix, which requires much more effort for the same process. This expert system, along with rule based approximate reasoning and abductive reasoning, provides transparent trustworthy results with human-in-the-loop, unlike the black box approach found in AI.
- iii) **Requirements of FAVER using System-of-Systems (SoS) methodology:** This research develops the architecture of FAVER in a transparent method using SoS methodology starting from capturing the requirements of FAVER until the verification process to ensure the functioning of FAVER. This exercise is carried out to be in alignment with the current industrial practices, thus enabling the implementation of this framework in both industry and academia for further development of vehicle level health monitoring.
- iv) **The thought experiment in FAVER:** One of the significant academic contributions from this research is the thought experiment that was carried out at the beginning. This thought experiment helped to learn the underlying complexities of representing an aircraft system, which led to the formation of assumptions. This exercise helped to limit and guide the conceptualization of the framework that had to be realized within a short period of time.
- v) **Reasoning in aerospace IVHM:** The paper titled 'The application of reasoning to aerospace Integrated Vehicle Health Management (IVHM)' is one of the first academic contributions in reviewing various reasoning methods applied in the field of aerospace for IVHM, as well as exploring its relationship with respect to their timeline in the field of artificial intelligence. In this paper, a generalized reasoning system is also derived by comparing several other reasoning systems, which can be used to build any reasoning system similar to FAVER's. Due to the underexplored areas covered in this work, this paper has garnered 17000+ views, and 10+ citations in the span of 18 months, and is featured as the most downloaded paper in the journal homepage as on 2nd November 2020 [1].

- vi) **The role of Digital Twin in IVHM:** The paper titled ‘Understanding the role of a Digital Twin (DT) in Integrated Vehicle Health Management’ is again, one of the first positional paper which derives the role of a DT in IVHM, based on its applications in both industry and academia. Also, a Digital Twin ecosystem is derived based on its various roles along the lifecycle of products, together with the required input and output from the DTs at every stage and their enabling technologies. In this paper, the role of DT is represented in an OSA-CBM influenced IVHM system architecture, which is an important contribution, given the increasing applications of DTs in every field, including IVHM.
- vii) **The Electrical Power System Digital Twin and system level diagnostics with ANFIS monitor and causal reasoning:** The Electrical Power System (EPS) simulation model is developed with connections to its interaction network with other aircraft systems through LRUs like valves and pumps. The EPS diagnostics is made up of an Adaptive Neuro-Fuzzy Inference System (ANFIS) monitor, which, together with causal reasoning, is able to isolate both continuous and intermittent faults injected in the LRUs in the interaction network. This standalone simulation model and the diagnostic algorithm can be adapted for other EPS related researches.
- viii) **The OSA-CBM methodology for developing diagnostics:** While the EPS DT is developed from scratch, the DTs of the engine, the Fuel System (FS), and the Environmental Control System (ECS) are adapted from existing works both open source as well as copyrighted. In order to develop standalone diagnostics for each of these systems, the OSA-CBM methodology was adapted. The OSA-CBM standard has been used widely in IVHM systems and has been applied for diagnostics in industrial applications. However, in this thesis, the OSA-CBM methodology has been used to demonstrate the development of diagnostics for multiple aircraft systems, thus streamlining the process and saving time for setting up the platform for high level reasoner like FAVER.

9.3 Future work

To conceptualize, develop, implement, and demonstrate FAVER, a number of assumptions were made at the beginning of this research. Following the successful fulfilment of aims and objectives of this research, FAVER can now be explored in various

scenarios to make it more realistic of its physical counterpart. For example, this research brought together the interacting systems, the EPS, the engine, the FS, and the ECS. The EPS was brought in the framework as it adds a multi-physical dimension to the DT layer, as well as adds complications to the interaction networks. Hence, in the future, FAVER can be further expanded and tested using more integrated and interacting systems like avionics and control systems. The other assumptions made in the research, such as no false alarms, always true sensor readings, and only steady state scenarios can also be removed or conservatively updated to make FAVER more updated with the recent areas of research.

The system level diagnostics developed for FAVER in this research are only data-driven and consider only single faults. For future work, these diagnostics could include complicated scenarios like simultaneous degradations, false alarms, incorrect sensor readings, and cascading faults within the subsystems. The better system level diagnostics lead to better vehicle level reasoning by FAVER. Model based diagnostics can also be brought into FAVER to showcase its ability to integrate with any type of diagnostics.

The transparent and trustworthy reasoning through the expert system in FAVER can be strengthened by verifying the knowledge by extracting knowledge from different data sources using data-driven methods. Also, the strength of rule based reasoning can be increased by adding text mining and other related techniques to implement contextual reasoning for vehicle level health monitoring. The other breaking points of this expert system could be further explored and bolstered with related reasoning techniques to make FAVER's reasoning robust enough to face more complicated scenarios.

FAVER is built upon the system level diagnostics of the aircraft systems. Validation of this method can be carried out by reverse engineering from the symptom vectors in the simulation models to identify the parameters of the root causes. FAVER's abilities can also be validated by using an entirely different set of systems other than those used in this research, as well as developing data-driven models like deep learning to prove the reasoning ability of FAVER if different sets of data are available.

Since FAVER is system agnostic and is not contained within the boundaries of any particular characteristics of a vehicle, it will be interesting to see how FAVER can be implemented in other vehicles other than aircraft and if all its capabilities hold true for their vehicle level health monitoring.

Reference

1. Elsevier. Progress in Aerospace Sciences. 2020. Available at: <https://www.journals.elsevier.com/progress-in-aerospace-sciences> (Accessed: 2 November 2020)

APPENDICES

Appendix A

Algorithm for Method 2 in the EPS diagnostics: ANFIS monitor with fuzzy boundary approach

Run simulation

i1=1;

for every 5 sec

% customizable window %

Power_gen (i1) = mean (Power_Generated);

i1=i1+1;

end

%load trained ANFIS file%

fis=readfis ('Neurofis.fis');

test = evalfis(fis, Power_gen);

[n]= row(Power_Gen);

% Classification usingANFIS monitor %

for i2= 1:n

if (z(i2) <=5, and z (i2) >=4) result(i2) = 'Region 5'

elseif (z(i2) <4, and z (i2) >=3) result(i2) = 'Region 4'

elseif (z(i2) <3, and z (i2) >=2) result(i2) = 'Region 3'

elseif (z(i2) <2, and z (i2) >=1) result(i2) = 'Region 2'

elseif (z(i2) <1, and z (i2) >=0) result(i2) = 'Region 1'

elseif (z(i2) <0) result(i2) = 'Region 0'

elseif (z(i2) >5) result(i2) = 'Undefined'

end

end

if z(1)=z(2)=.....z(n),

monitor_res= z(1)

else

monitor_res = 'intermittent'

end

% Diagnosis with Fuzzy boundary approach and Causal reasoning with diagnostic matrix %

if monitor_res != 'intermittent'

carry tests from the diagnostic matrix (table 6.2) to isolate the root cause

end

Appendix B

This appendix shows the step-by-step demonstration of the FAVER's reasoning. [Figure B-1](#) shows the data architecture diagram for FAVER. While the data flow activities are already shown in activity diagrams such as [Figure 8-6](#) and [Figure 8-7](#), this figure shows what type of data is collected, where it is stored and how it is used in the different modules in the architecture of FAVER.

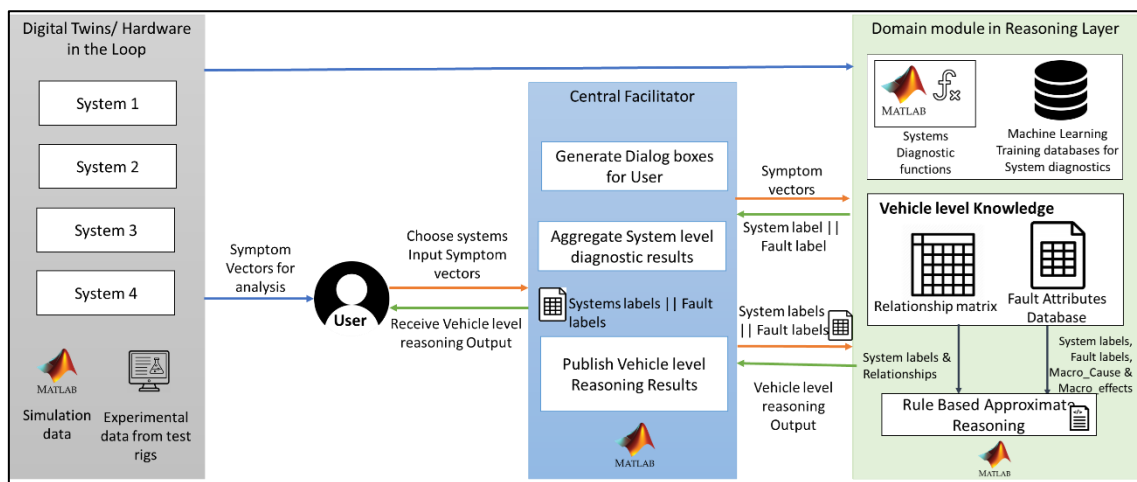


Figure B- 1 Data architecture diagram for FAVER

The different steps involved in FAVER's reasoning carried out at the system and vehicle levels are shown in [Figure B- 2](#). The first four steps are to be done by the user, which includes uploading all databases (training sets and diagnostic functions for system diagnostics) required for the analysis, choosing the systems for analysis and input the symptom vectors. The Central facilitator uses these data, runs the system level diagnostics followed by the Domain facilitator's program on vehicle level reasoning with rule based

approximate reasoning algorithm, as shown in activity diagrams (Figure 8-6 and Figure 8-7).

```

% Activities by the user %
% Activities by the Central facilitator %
% Activities by the Domain facilitator %
=====
Step 1a: Load Databases (into Domain knowledge)
Step 2a: Run the program
Step 3a: Choose systems for analysis
Step 4a: Input symptom vectors for chosen systems when dialog box appears
%% At systems level %%
Step 1b: Show the dialog box for the symptom vector input from the user for chosen systems
Step 2b: Access system diagnostic functions from domain knowledge
Step 3b: Run diagnostics with the input symptom vector
Step 4b: Fetch the generated system diagnostic results: Fault labels/ healthy
Step 5b: Repeat steps 1-3 until all chosen systems diagnostics are run
Step 6b: Aggregate the systems level results
%% At vehicle level %%
Step 7b: Load vehicle level knowledge bases (relationship matrix and fault attributes database)
Step 8b: Check if all systems healthy from step 5,
        if yes, show all systems are healthy, go to Step 10.
        else show the healthy systems, run analysis for remaining systems
Step 9b: For the remaining systems, compare fault labels from results in step 5 to the data in fault attributes database :
        Step 9.1: If no macro cause and no macro effect: Single fault, no interaction
        Step 9.2: If only macro effect: Check successor of system in vehicle level relationship matrix, generate
        cascading effect warning.
        Step 9.3: If only macro cause: Check predecessor of system in vehicle level relationship matrix, generate root
        cause possibility
        Step 9.4: If both macro cause and macro effect: Generate root cause and cascading effect warnings
Step 10b: Provide generated results to the Central facilitator program
Step 11b: Publish results
Step 12b: Stop

```

Figure B- 2 Steps involved in FAVER's reasoning

Symptom Vector Generation

The vehicle symptom vector that is to be input to FAVER by the user, is generated by a series of steps depending upon the number of systems chosen for the analysis. For example, consider a scenario as seen in Figure B- 3 where the engine is with the fault mode '*Bleed Valve stuck*' leading to '*Low Bleed Air pressure*' in the ECS, while the EPS and the FS are healthy.

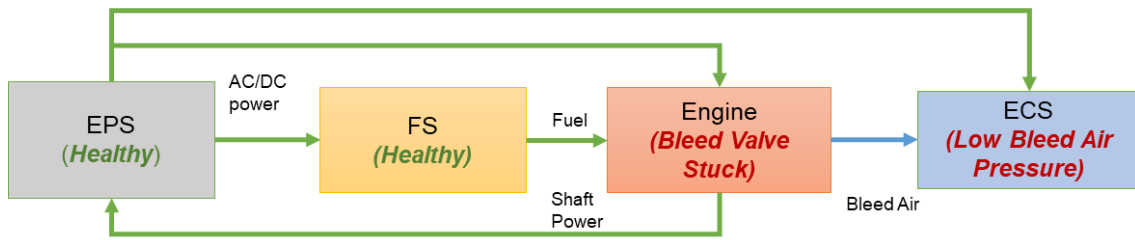


Figure B- 3 ‘Bleed valve stuck’ scenario

For this scenario, the vehicle symptom vector is generated through the following steps.

- i) The EPS DT is run at the ‘healthy’ conditions and the 9 parameters chosen to be the part of EPS symptom vector are aggregated.
- ii) The symptom vector with 7 parameters of the FS is chosen from the database of 100% healthy experimental run.
- iii) The engine DT is run with the fault mode ‘*Bleed valve stuck*’ and the four parameters chosen for its symptom vector are aggregated.
- iv) The ECS DT is run with the ‘*Low Bleed Air pressure*’ fault mode and the four parameters chosen for its symptom vector are aggregated.

Figure B- 4 shows the parameters of each system’s symptom vectors for the above-mentioned conditions. All these system level symptom vectors together make the vehicle symptom vector, which is then input to FAVER’s central facilitator by the user.

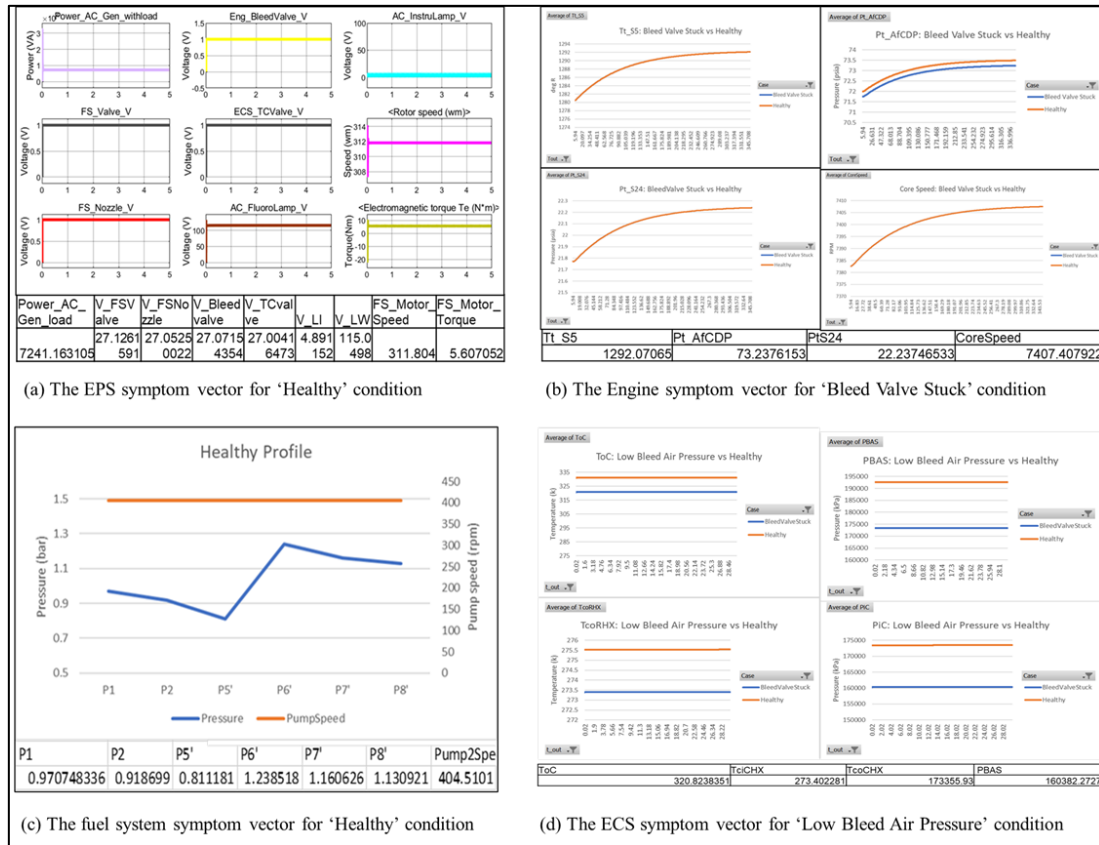


Figure B- 4 The vehicle symptom vector generation for 'Bleed Valve stuck' scenario

The snapshots during the runtime of the Central facilitator's and the domain facilitator's programs are shown in [Figure B- 5](#) and [Figure B- 6](#). These are in alignment with the steps shown in [Figure B- 2](#). The Central Facilitator runs the system diagnostics for the chosen systems with the input symptom vector, aggregates the result and provides it for the Domain facilitator. In the Domain facilitator, the vehicle level knowledge bases are uploaded, and the system level results are compared with the knowledge using rule based approximate reasoning program. The final vehicle level results are generated as a result of this reasoning and is fed back to the Central facilitator and is published to the user.

Step 1a-4a: Done by the user

- Step 1a : Load the training files required for system diagnostics
- Step 2a: Run the program
- Step 3a: Choose the systems for analysis (4 systems are chosen in this case)

Step 4a: Input the symptom vector for each system one by one in the appearing dialogs

Symptom
vector

	A	B	C	D	E	F	G	H	I
1	EPS Healthy	7244.219	27.03242	27.04972	27.03577	27.02533	4.891152	115.0498	311.804
2	FS Healthy	0.970448	0.912409	0.11758	0.23162	0.147731	410.0713		
3	Eng Bleed Valve fault	1292.07065033992	73.23761531074	22.237465330701	7407.40792150373				
4	ECS Low Bleed Air pressure	320.82383514912	273.402281448349	173355.93	160382.272700923				
5									

Dialog boxes of symptom vector input for each
system diagnostics

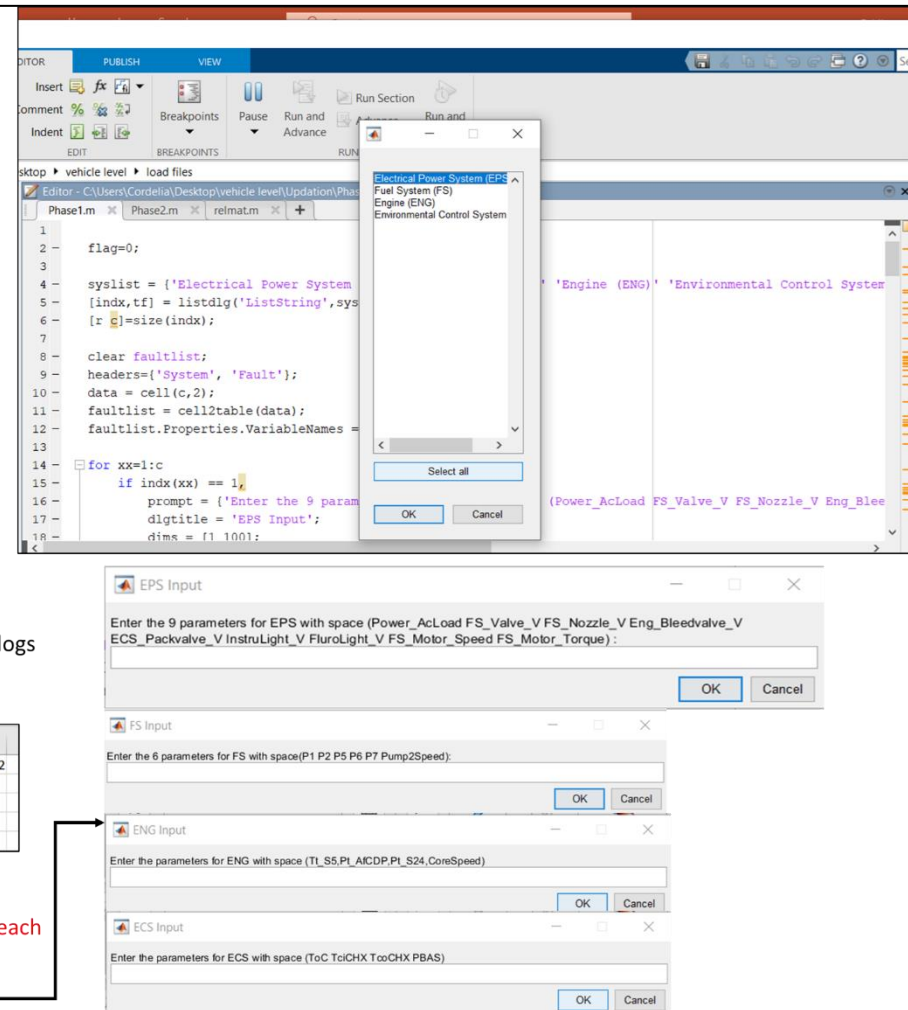
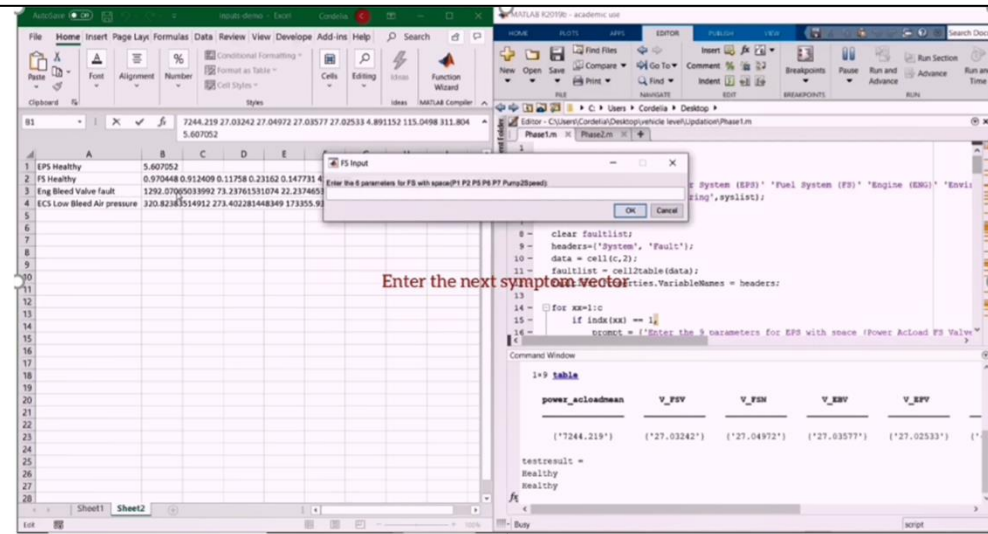


Figure B- 5 Activities by the user for FAVER's reasoning

Steps 1b-12b: By the Central Facilitator and Domain Facilitator

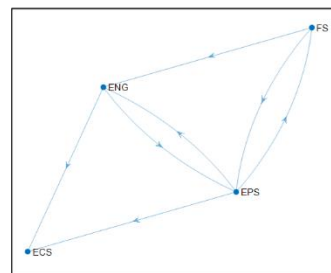
- Step 1b: Show dialog box to user for input
- Step 2b: Access and Run System diagnostics
- Step 3b: Fetch system diagnostics results
- Step 4b: The user is prompted to enter the next system's symptom vector as soon as the previous system diagnostics produces results
- Step 5b: Steps 1b-3b are repeated for all four systems
- Step 6b: Aggregate the system level results
- Step 7b: Load relationship matrix and fault attributes database
- Step 8b Check if systems are healthy
- Step 9b: For remaining systems, vehicle level reasoning is carried out using the relationship matrix and fault attributes database shown in figure 8-7.
- Step 10b: Results of vehicle level reasoning generated and sent to Central facilitator



Step 2b-4b: System diagnostics of one system is run and the dialog box for input appears for the next system

System	Fault
["EPS"]	["Healthy"]
["FS"]	["Healthy"]
["ENG"]	["Bleed Valve Stuck"]
["ECS"]	["LowBleedAirPressure"]

Step 6b: System level diagnostics results aggregated by the Central facilitator



Step 7b (background file): Relationship matrix representation of interaction between the systems at vehicle level

```

Command Window

These systems are healthy:
"EPS"    "FS"

ECS Bleed Valve Stuck
"This fault from"    "ENG"

might not have originated from another system
"could affect"    "ECS"

Low Bleed air Pressure
"This fault from"    "ECS"

"could have been caused by"    "ENG"
    
```

Step 10b: Generated vehicle level results

Figure B- 6 Activities by the Central facilitator and the Domain facilitator for FAVER's vehicle level reasoning

Appendix C

SWOT Analysis

A SWOT (Strength, Weakness, Opportunity, and Threat) analysis is carried out to the developed framework.

C.1 Strengths

FAVER combines the concept of reasoning and DTs to isolate faults considering interactions between multiple aircraft systems. FAVÉR possesses some exclusive features that help the framework to achieve the objectives.

Framework: FAVÉR is expandable yet robust; any aircraft system can be added to the framework and considered for health reasoning at the vehicle level with very few changes to the existing framework. It is system-agnostic and flexible; the framework is independent of the aircraft systems used for demonstration as well as their diagnostic algorithms. Existing DTs and diagnostics can be easily merged with the framework.

Architecture: The architecture of FAVÉR features modularity in its DT layer by encapsulating the digital twins and enabling interaction between the system DTs only through FAVÉR's Central facilitator. This helps to avoid the otherwise tedious V&V process that would happen when any small change is carried out with the aircraft system DTs affecting the interaction network with other DTs in the framework.

Reasoning: FAVÉR employs a generalized reasoning system that uses expert system knowledge at the vehicle level along with multiple reasoning strategies, such as abductive and rule based approximate reasoning, for processing results, and case based reasoning for its implementation. FAVÉR's reasoning can be easily customized based on users/engineers preferences for identifying root causes and/ or cascading effects or both. The expert system's knowledge base in the reasoning layer of FAVÉR uses a unique combination of a relationship matrix with a fault attributes database, and it avoids unnecessary expansion of vehicle level knowledge. This enables easy integration of new faults to an existing system level diagnostic function, as well as of the new system to the framework. The use of an expert system makes FAVÉR's reasoning transparent and trustworthy, unlike the black box approaches made by data-driven reasoning.

C.2 Weaknesses

Dependence on system level diagnostics: FAVÉR's vehicle level reasoning is built

upon system level diagnostics; these should be run for all the involved systems for every analysis at the vehicle level. Also, the vehicle level reasoning inherits the misclassification rates of the system level diagnostics.

Dependence on experts: FAVER is built against a background of engineering knowledge. The engineers who use and extend this framework should be knowledgeable in the domain: in the use of system level diagnostics, and in updating the relationship matrix and fault attributes database at the vehicle level. Also, as FAVER's vehicle level reasoning includes the abductive reasoning technique, some results from FAVER require expert interpretation.

C.3 Opportunities

Prospects in the industry: FAVER is an OSA-CBM influenced framework that can be adapted to many different applications such as vehicle level health monitoring software packages for ground testing and maintenance, and plug-in IVHM systems for onboard monitoring. The use of DTs will enable onboard monitoring with the help of infrastructures like Internet-of-Things.

Strengthening the architecture and reasoning: The Central facilitator can be automated to take multiple symptom vectors in and process results for several scenarios at a given time. FAVER's reasoning can be bolstered with text mining and other such techniques to provide accurate root causes and cascading effects. FAVER's expert system knowledge should withstand and scale appropriately to the number of systems, their fault modes, and complicated interactions between them in a complex vehicle like an aircraft. While this scalable feature has been demonstrated in this paper, the breaking point of this knowledge base should be tested and strengthened for further expandability.

Expanding the boundaries: FAVER's capabilities can be expanded by updating the assumptions made during the initial stages of the framework. For example, currently, only steady state conditions are considered for the simulation, and sensors are always considered as healthy. These assumptions could be changed and tested to adapt FAVER to more realistic scenarios. Simultaneous faults are considered only across the systems and could be extended to be within the systems. Also, the degree of degradation of faults at the system level is not considered while reasoning at the vehicle level. Due to time consideration, FAVER was tested only with data-driven diagnostics. The framework is structured in a system level diagnostic agnostic way. However, this needs to be tested and

proven. FAVER can be validated with a new set of DTs and system level diagnostics. Reverse engineering from the symptom vectors to isolate the root causes can also validate FAVER's reasoning.

C.4 Threats

Adapting to simplicity: The specification for FAVER makes the assumption that a user is an expert in the specific domain. FAVER has to be made adaptable to be used by a general engineer as well as those who are an expert in the field.

Adapting to complexity: With the rise in data generated for every flight, FAVER needs to be adapted to accommodate new algorithms and techniques to extract knowledge from this data and use it to establish vehicle level relationships, rather than depend only upon the experts.