IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

IEEE RELIABILITY SOCIETY SECTION

Received June 29, 2021, accepted July 26, 2021, date of publication July 28, 2021, date of current version August 9, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3100865

Development and Implementation of a Framework for Aerospace Vehicle Reasoning (FAVER)

CORDELIA MATTUVARKUZHALI EZHILARASU AND IAN K. JENNIONS
Integrated Vehicle Health Management (IVHM) Centre, Cranfield University, Bedfordshire MK43 0AL, U.K.
Corresponding author: Cordelia Mattuvarkuzhali Ezhilarasu (c.m.ezhilarasu@cranfield.ac.uk)

This work was supported by the Boeing Company, as part of their collaboration with Cranfield University IVHM Centre.

ABSTRACT This paper discusses the development and implementation of the architecture of a Framework for Aerospace Vehicle Reasoning, 'FAVER'. Integrated Vehicle Health Management systems require a holistic view of the aircraft to isolate faults cascading between aircraft systems. FAVER is a system-agnostic framework developed to isolate such propagating faults by incorporating Digital Twins (DTs) and reasoning techniques. The flexibility of FAVER to work with different types and scales of DTs and diagnostics, and its ability to adapt and expand for previously unknown faults and new systems are demonstrated in this paper. The paper also shows the novel combination of relationship matrix and fault attributes database used to structure the knowledge of FAVER's expert system. The paper provides the working mechanism of FAVER's reasoning and its ability to isolate faults at the system level, identify their root causes, and predict the cascading effects at the vehicle level. Four aircraft systems are used for demonstration purposes: i) the Electrical Power System, ii) the Fuel System, iii) the Engine, and iv) the Environmental Control System, and the use case scenarios are adapted from real aircraft incidents. The paper also discusses the pros and cons of FAVER's reasoning via demonstrations and evaluates the performance of FAVER's reasoning through a comparative study with a supervised neural network model.

INDEX TERMS Reasoning, health monitoring, aircraft systems, cascading faults, aircraft accidents, OSA-CBM.

I. INTRODUCTION

Integrated Vehicle Health Management (IVHM) is an evolving capability that enables the Condition Based Maintenance (CBM) of complex vehicles like aircraft. It uses data from various sources such as sensors, maintenance records, and design documents to condition monitoring, diagnosing faults and degradations, and evaluating the remaining useful life of the concerned systems [1]. For an aircraft, the main objective of IVHM is to enable the vehicle to function as intended, increasing its reliability and availability, and saving time and cost associated with unexpected downtime and the eventual consequence of prolonged maintenance activities. IVHM aims to be a part of the aircraft'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 mature process [1].

The associate editor coordinating the review of this manuscript and approving it for publication was Zhaojun Li.

Across the aircraft industry, IVHM has focused mainly on health monitoring at the component, Line Replaceable Unit (LRU), and system levels. 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 [2]. 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 [3]. Similarly, other original equipment manufacturers such as Safran and Meggitt provide Engine Monitoring Units for aircraft and HUMS for helicopters [4]–[7], while consultancies like Infosys provide landing gear health monitoring solutions [8]. TEAMS-Remote Diagnostic Server (RDS), a COTS product provided by Qualtech System Inc., focuses on systems level health monitoring and was developed to remotely monitor and diagnose International Space Station. It uses sensor data, equipment and field data and diagnoses multiple faults present in the systems. TEAMS-RDS uses a cause-effect approach to diagnose faults, and it helps in real time and proactive monitoring of the systems, along with troubleshooting. Its real-time embedded reasoner named TEAMS-RT, uses matrices of onboard tests and equipment failures to isolate root causes of the faults in target systems. These results are ideally suited for Integrated Health Management applications. TEAMS-RDS can be applied in aircraft, ships, and automobiles [9], [10]. On the other hand, methods like distributed diagnosis and timed failure propagation graphs are used to demonstrate failure propagation into surrounding areas in HVAC systems [11] and power distribution systems [12] and in fuel systems [13].

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 [2]. Furthermore, fault propagation from one aircraft system to another has become more frequent due to the increased interactions between the aircraft systems. While it is usually addressed during maintenance, if a fault takes an unexpected propagation path, its cascading effects are difficult to troubleshoot. 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 [14]. 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 [15]. 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 [16]. Such unexpected fault propagation paths lead to unplanned downtime costing the airlines time and reputation. 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.

Only a few research works consider subsystems/systems interactions at the vehicle level [2]. A vehicle health monitoring system architecture patented by Honeywell employed a decision support system that 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 [17]. 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 [18]. These examples of IVHM systems show that the area of vehicle level health monitoring is still underexplored. Moreover, many studies carried out at the vehicle level have a centralised architecture [8], [27], which might 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 Digital Twin (DT) and reasoning has been proposed [28].

This paper explains how FAVER's architecture, influenced by the Open Standard Architecture for Condition Based Maintenance (OSA-CBM), is developed and how it explores vehicle level health monitoring for the aircraft by employing reasoning and digital twins. To that effect, Section I shows the need for vehicle level health monitoring in IVHM systems by presenting the aircraft accident scenarios and state-of-the-art IVHM systems in the industry. Section II provides FAVER's general schematic, objectives, and related works to set up the framework. Section III presents the architecture of FAVER and demonstrates the two important properties of FAVER: its scalability and flexibility. In section IV, the objectives of FAVER to isolate root causes of single and interaction faults in newly added systems, and to resolve ambiguities, are demonstrated with examples of fault propagation involving four aircraft systems akin to real aircraft incidents. Section V 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 VI summarises the paper and provides some ideas for the future development of FAVER.

II. 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 [19]. FAVER consists of two essential parts: 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. Fig.1 shows the general working schematic of FAVER [20], consisting of four layers, influenced by OSA-CBM. OSA-CBM implements ISO 13374 Condition Monitoring and Diagnostics of Machines [21] to define an open standard for distributed CBM. It is currently maintained by Machine Information Management Open Systems Alliance (MIMOSA). OSA-CBM is defined using Unified Modelling Language (UML) and is platform independent. It enables

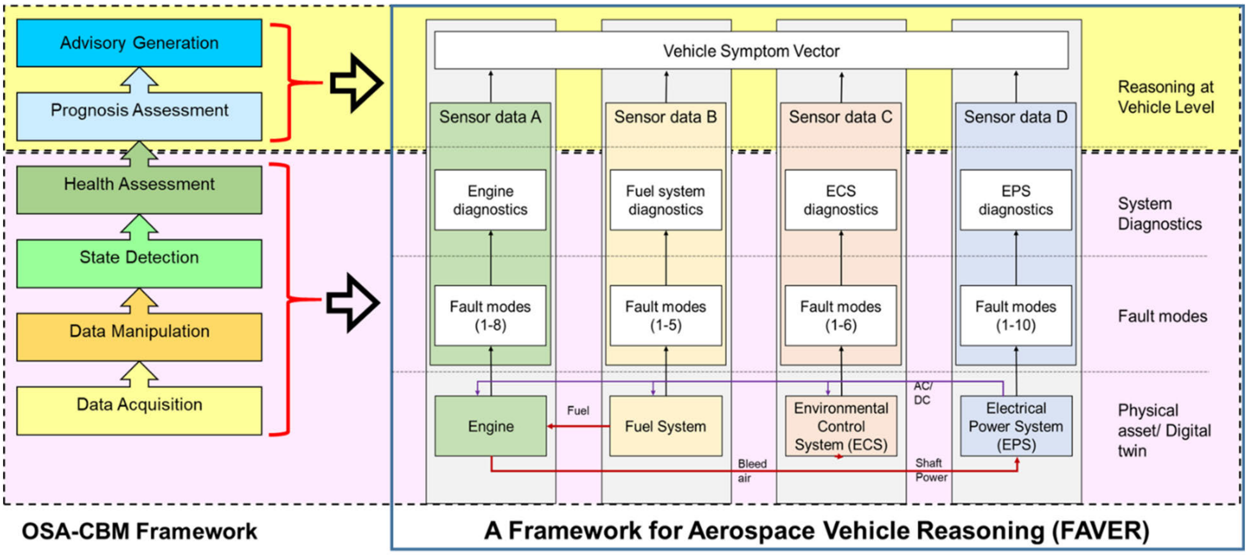


FIGURE 1. Working schematic of FAVER influenced by OSA-CBM framework.

processing of multiple types of information without involving technical interfaces. OSA-CBM consists of six layers: i) Data Acquisition, ii) Data Manipulation, iii) State Detection, iv) Health Assessment, v) Prognosis Assessment, and vi) Advisory Generation [22]. The FAVER framework is developed in parallel to the layers defined in OSA-CBM. The DT layer, the bottom row of FAVER in fig. 1, consists of the aircraft systems considered in the framework. This layer is used to acquire health data from different representations of aircraft systems. The next two rows cover the system level diagnostics and fault modes for each of the aircraft system DTs. Data collected from the DT layer for both healthy and fault modes is manipulated and health information is extracted. These are used to develop diagnostics for each aircraft systems, in parallel to the State Detection and Health Assessment layers of OSA-CBM. Considering the bottom three layers of FAVER 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 (sensor readings of selected parameters representing the health of each system) as input. This layer enables fault isolation, including interacting faults at the vehicle level, and carries out qualitative prognosis to predict the effects of these faults over the aircraft systems in the interaction network, providing advisories regarding the potential root causes and cascading effects at the vehicle level. Fig.1 shows four aircraft systems comprising the interaction network: 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 fig.1. 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). 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 fig.1, the objectives FAVER are to:

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

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 fig.2. A brief recap of previous work is given here in order to explain the overall scale of the work and the current contribution:

A. THOUGHT EXPERIMENT AND ASSUMPTIONS

A thought experiment was conducted using the fuel rig [19]. It examined the complexity of such a system and helped define the assumptions necessary 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

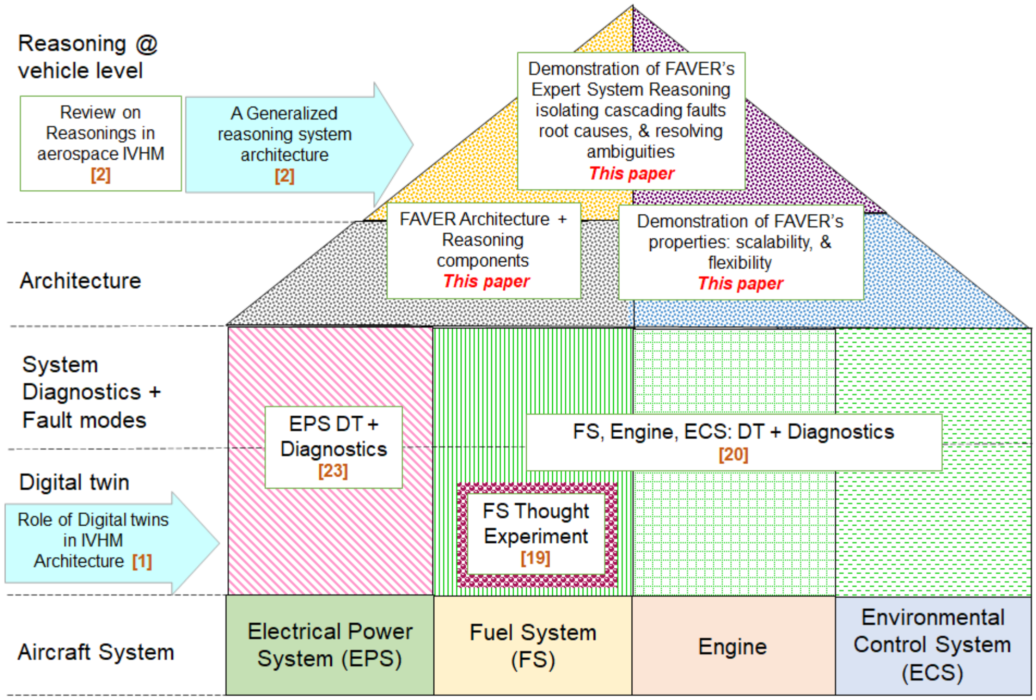


FIGURE 2. Previous work related to FAVER and contributions in the current paper.

systems on an aircraft. By considering the EPS in this work, one of these far-reaching systems is 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 within the systems, as the emphasis here is to look at interaction faults.

v. There are no false alarms.

These assumptions are planned to be removed or conservatively updated in future projects.

B. LITERATURE REVIEWS

i) A review of the reasoning literature [2], including many fields apart from aerospace, enabled the selection of reasoning approaches to be demonstrated in the project. A generalised reasoning system was derived from comparing various reasoning systems; it 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.

ii) A review of the roles of a DT in the overall engineering process, and IVHM in particular 1], 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.

C. DIGITAL TWINS AND SYSTEM DIAGNOSTICS

i) A DT for the EPS was created via a MATLAB-SIMULINK model along with its own diagnostics [23]. The diagnostics used ANFIS (Adaptive Neuro-Fuzzy Inference System) combined with causal reasoning and is capable of isolating ten fault modes.

ii) Subsequently, a generalised methodology, guided by OSA-CBM, was devised for system diagnostics and demonstrated on the fuel system, ECS, and engine systems [20]. The diagnostic for the FS, represented by HIL as mentioned above, is a decision tree-based function that can isolate five fault modes. The engine was developed using T-MATS in MATLAB-SIMULINK [24]; 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 [25], and its system level diagnostic function uses a Linear Discriminant Analysis (LDA) algorithm and can isolate six fault modes.

D. CURRENT CONTRIBUTION

The current contribution is also shown in fig.2 and comprises a demonstration of the strength of FAVER's architecture through its significant features and its reasoning through real-world examples. It contains the following:

i) The architecture of FAVER: its building blocks and working mechanism, and demonstrations on its vital properties– scalability and flexibility.

a) The flexibility of FAVER is demonstrated by using different types of system DTs and system diagnostics in the framework.

b) For scalability, the demonstration is done by expanding the framework in two steps by including new faults to every system's diagnostics in the

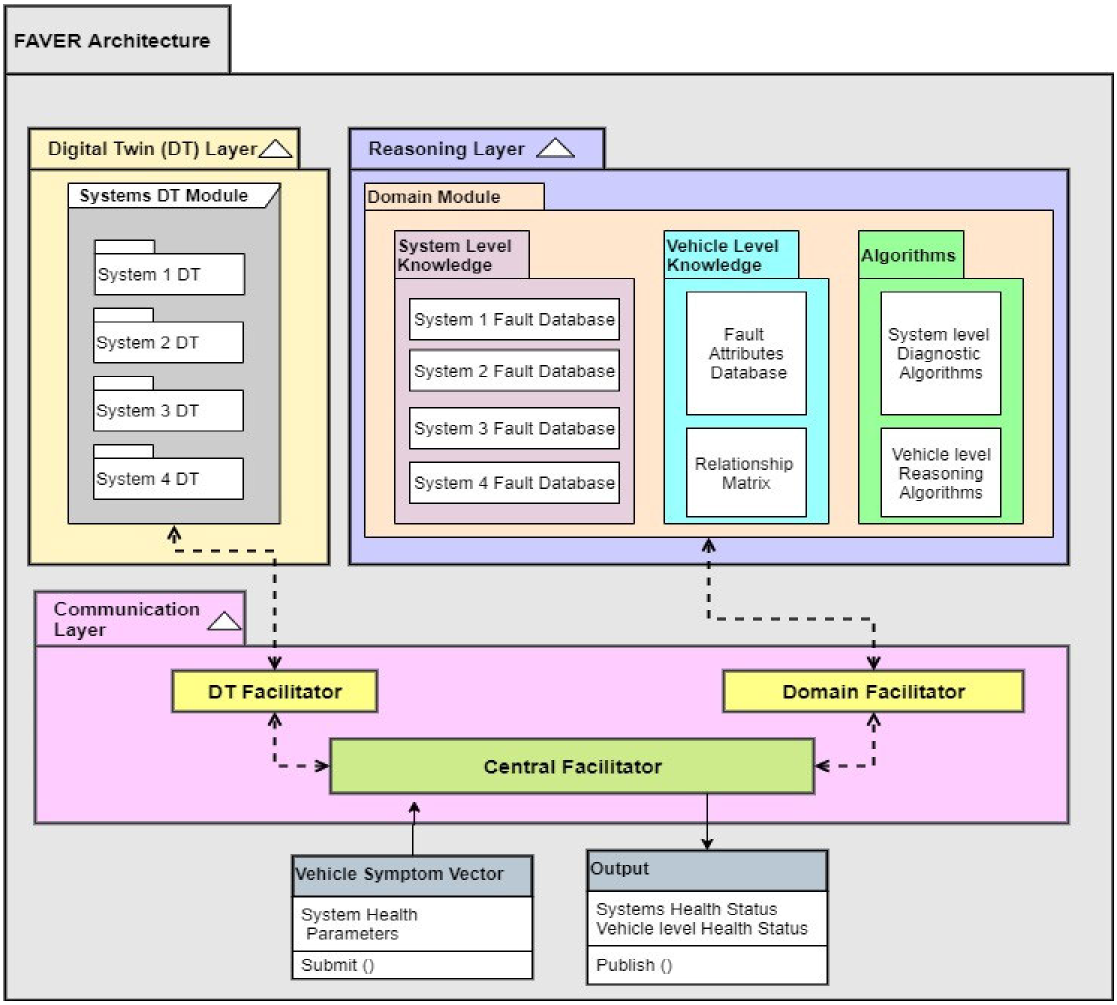


FIGURE 3. The architecture of FAVER.

framework, and by including two more complicated aircraft systems.

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.

ii) The reasoning part of FAVER uses a novel knowledge base for its expert system and a mixture of reasoning techniques. The paper demonstrates the strength of FAVER's reasoning in isolating the root causes and predicting the 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 resolve ambiguities arising from such scenarios.

iii) The overall framework is critically evaluated in two steps:

a) This paper presents the cost of using FAVER's reasoning, such as the need to assign priorities and misclassification costs.

b) The reasoning part of FAVER is compared with a data driven reasoning using a neural network, and the pros and cons of using FAVER's reasoning are explained.

III. FAVER ARCHITECTURE

To implement the schematic of FAVER shown in fig.1, the architecture of FAVER is developed by bringing together two distinct layers: i) DT and ii) Reasoning, through a mediating third layer, viz, iii) Communications layer. FAVER's architecture as shown in fig.3, has a layered structure comprising of federated modules. The architecture is developed by adapting a directed System-of-Systems (SoS) methodology [26]. Here, FAVER is characterised as a platform of SoS and its building blocks (modules) like the aircraft system DTs and their diagnostics, which are usually independent, work together as SoS for the central purpose of diagnosing faults at the vehicle level. This is accompanied by the other modules required for fault isolation at the vehicle level as well as communication between three layers.

A V-diagram model is adapted for the development of FAVER's architecture, as shown in fig.4. This is based

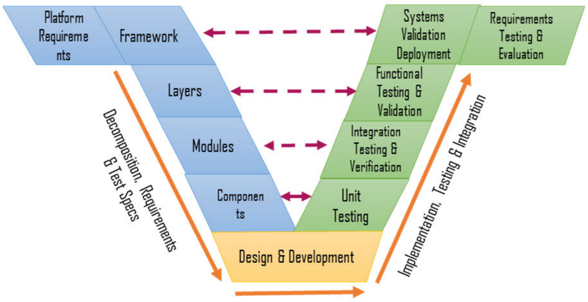


FIGURE 4. The V-diagram adapted for FAVER.

on the guidelines provided for capturing requirements for an IVHM system using a System Engineering methodology [27]. Through this V-diagram, the objectives of FAVER mentioned in section II are mapped as the general requirements of FAVER as a platform, and its architectural blocks are characterised to achieve the overall objectives. FAVER's architecture is structured in three levels: i) Level 1: FAVER (as a platform), ii) Level 2: DT layer, Reasoning layer and Communications layer, and iii) Level 3: DT modules and Domain modules. Since FAVER is considered as a SoS, most of its modules, especially the system DTs are supposed to be functioning independently, a Holistic Requirements model [28] is approached for collecting i) Operational Requirements, ii) Functional Requirements, and iii) Non-functional Requirements for all three layers of FAVER in level 2. For the lower-level modules and their components, only functional requirements are collected, as they collectively meet the operational and non-functional requirements defined for upper level layers. Table 1 shows the list of requirements captured for FAVER at each level and the ways through which they are addressed. The features and the functions of the building blocks of FAVER's architecture are described in the following subsections.

A. DIGITAL TWIN (DT) LAYER

The DT layer consists of aircraft system DTs as individual modules. The key characteristic features of the DT layer are as follows.

i) To ensure the *flexibility* of the framework, each DT of an aircraft system can be either physics-based, function-based models or data-driven representations for emulating the usual operations including the interactions of the aircraft systems. The DT layer can also contain an HiL representation of a system.

The DTs 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 health state of the system as the output in the form of a symptom vector required for further reasoning. These DTs are also used to generate data required for developing diagnostics for the concerned systems.

ii) Each DT has the property of encapsulation and performs I/O only 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 is introduced for the *expansion* the framework. Instead, the DT Facilitator orchestrates running each DT and the interactions between them, maintaining the modular characteristic of system-of-systems.

Using this approach enables systems on different scales to work together. For example, in the demonstrations shown in this paper, 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, which 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.

B. REASONING LAYER

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, and predict cascading effects. The Reasoning layer is adapted from a generalised reasoning system derived as part of the background work for setting up this framework [2]. The Reasoning layer consists of a domain module acting as the brain of FAVER. The domain module comprises diagnostic functions at the systems level and an expert system at the vehicle level. It also has the vehicle level reasoning algorithm that consists of a forward chaining mechanism of rule based reasoning, wherein the goals of identifying whether the 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.

1) FAVER'S EXPERT SYSTEM

The significance of FAVER's expert system at the vehicle level is its knowledge base, which is made up of a novel combination of a relationship matrix and a fault attributes database. The relationship matrix is created at the vehicle level by experts to establish the interacting relationships between the aircraft systems. To complement this, the fault attributes database is a tuple consisting of five elements as given by:

$$\text{Fault Attributes Database} = \{\text{L, C, O, MC, ME}\} \tag{1}$$

where, L = System fault label
C = Fault codes

TABLE 1. Requirements of FAVER.

Level	Type of Requirements	Requirements	Tests/ Actions to address requirements	Works addressing the requirements
Level 1: FAVER Platform	Operational Requirements	Enable identification of root cause of fault and prediction of its cascading effect in aircraft systems	i) **Known Fault isolation - Single & Interaction**: A known fault injected in any of the systems should be isolated, whether it affects the single system, or the system it interacts with, along with the root cause, and its cascading effect should be predicted. ii) **New (previously unknown) system Fault isolation – Single & Interaction**: A single fault that is previously unknown from an unfamiliar system that could affect the system of origin or the system it interacts with, should be accommodated by expanding the framework. iii) **Ambiguous Faults**: 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.	Demonstrations on the abilities of FAVER's reasoning in section IV
	Functional Requirements	1. Identify root cause and cascading effects of faults 2. Isolate single and interacting faults 3. Flag ambiguities		
	Non-functional Requirements	1. Scalability 2. Flexibility	**Scalability:** Demonstrate expansion and accommodation of new systems into the framework and new faults into the existing systems, without affecting the performance of the existing framework. **Flexibility:** Demonstrate accommodation of different types of system representations and diagnostic methods	Demonstrations on the strength of FAVER's architecture in section III
Level 2: Digital Twin Layer	Operational Requirements	1. Emulate aircraft systems operations and their interactions 2. The modules should behave as individual systems and be independent of each other. 3. The DT layer shall include both simulation models as well as Hardware in the loop.	Demonstrate the emulation of aircraft systems operations and their interactions, while maintaining their independent identities through separate simulations of their respective DTs. Showcase the various types of representations used to demonstrate flexibility within the DT layer	Section IV showcases the results for use case scenarios simulated through FAVER that involve aircraft systems operations and their interactions. These use cases involve simulations of independent DTs and Hardware in the loop.
	Functional Requirements	1. Output for simulation models of DT: Symptom vectors for the given healthy/ faulty input conditions. 2. Modules should be the representations (virtual twin) of the aircraft systems.		
Level 3: DT modules	Functional Requirements	Develop required modules to address the requirements of the DT layer	1. Choose and develop DT modules of interacting aircraft system with sufficient representation of the real-life counterparts. 2. Ensure the DT modules are capable of running independent of each other, as individual entities. 3. Choose the systems to include simulation DTs as well as Hardware-in-the-loops. 4. Finalise the parameters that form the symptom vectors, representing the health of the systems.	The papers on developing aircraft systems DTs and system diagnostics address the functional requirements of the DT modules.[20,23]
Level 2: Reasoning Layer	Operational Requirements	1. Layer with knowledge and reasoning mechanism to isolate faults at both system level and vehicle level. 2. Reasoning layer should act as the brain of FAVER 3. Each system diagnostics should be independent of the other diagnostics	1. Demonstrate the ability to expand the knowledge base to accommodate new systems and new faults through simple steps 2. Demonstrate the ability of system diagnostics to function independently and isolate faults from their respective systems 3. Demonstrate the ability of FAVER's reasoning to isolate faults with vehicle level interactions	Section III of this paper demonstrates the 1st requirement and section IV demonstrates the 2nd and 3rd requirements.
	Functional Requirements	1. Possess the systems diagnostics from all the concerned systems 2. Possess the knowledge to find the relationship between the systems. 3. Has algorithms to identify the root cause of faults that affect local systems as well as have macro effect on the other systems 4. Able to add new faults and systems to its knowledge base, supporting the expansion of the framework		
Level 3: Domain modules	Functional Requirements	Develop required modules to address the requirements of the reasoning layer	1. Develop diagnostics for each system and ensure they are independent of each other. 2. Develop Vehicle level knowledge that understands the relationship between the chosen systems. 3. Develop vehicle level reasoning algorithm to isolate faults and their root causes	The papers on developing aircraft systems DTs and system diagnostics address the 1st functional requirement of the Domain modules. [20,23] Sections III and IV of this paper demonstrate the 2nd and 3rd requirements.
Level 2: Communications Layer (Central Facilitator)	Functional Requirements	1. Serve as the bridge between the layers in the framework. 2. Facilitate the connection between the Reasoning layer and the Digital Twin layer. 3. Fetch input from the user regarding the symptom vector that needs to be checked for its health. 4. Provide results regarding the symptom vector, its fault status, and the label as the output.	1. Test the connections between the Communications layer and the Digital Twin layer 2. Test the connections between the Communications layer and the Reasoning layer 3. Demonstrate the input/output functionality of FAVER with user interaction	*The connection between the Communications layer and the Digital twin layer is manual; it will be automated in future work.* The connections between the Central Facilitator and the Reasoning layer and the interaction between the user and the CF are tested and verified when conducting the demonstrations of use cases.

O = Fault's system of origin
MC = Macro_Cause
ME = Macro_Effect

While data for the first three attributes can be collected from the respective system diagnostics, the attributes Macro_Cause and Macro_Effect are defined below.

a: MACRO_CAUSE

This attribute is defined as the possibility of a fault being caused by another system and, with pc being the possibility of a fault caused by another system, is defined by:

$$\text{Macro\_Cause} = \begin{cases} 1, & \text{if } pc > 0 \\ 0, & \text{if } pc = 0 \end{cases} \tag{2}$$

b: MACRO_EFFECT

This attribute is defined as the possibility of a fault affecting another system. With pe being the possibility of a fault affecting another system, it is given by:

$$\text{Macro\_Effect} = \begin{cases} 1, & \text{if } pe > 0 \\ 0, & \text{if } pe = 0 \end{cases} \tag{3}$$

The combination of a vehicle level relationship matrix with a fault attributes database enables a scalable expansion of knowledge, containing cause and effect relationships between the aircraft systems in a simpler and more efficient way than with a traditional relationship matrix.

The following illustration shows the difference between the knowledge base of FAVER's expert system and a traditional relationship matrix. Since the first three attributes of the fault attributes database are updated from the system diagnostics without any changes, they are not shown in its structure in the illustrations.

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, is:

Total number of elements in Matrix,

$$\mathrm{M} = \left(\sum_{i=1}^{n} (No.\ of\ faults\ in\ System\ i)\right)^2 \tag{4}$$

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 (4), a traditional relationship matrix would be of size 4 × 4, with 16 cells, as seen in fig.5. Consider the scenario where two new systems C and D are added in FAVER with four fault modes each: C1, C2, C3, C4 and D1, D2, D3, and D4. Now, the relationship matrix has to be increased in size to develop the relationship between the 12 faults. Hence the resulting matrix will have 144 elements for four systems with a total of 12 faults. From fig.5, the number of cells to be updated becomes 128. The experts

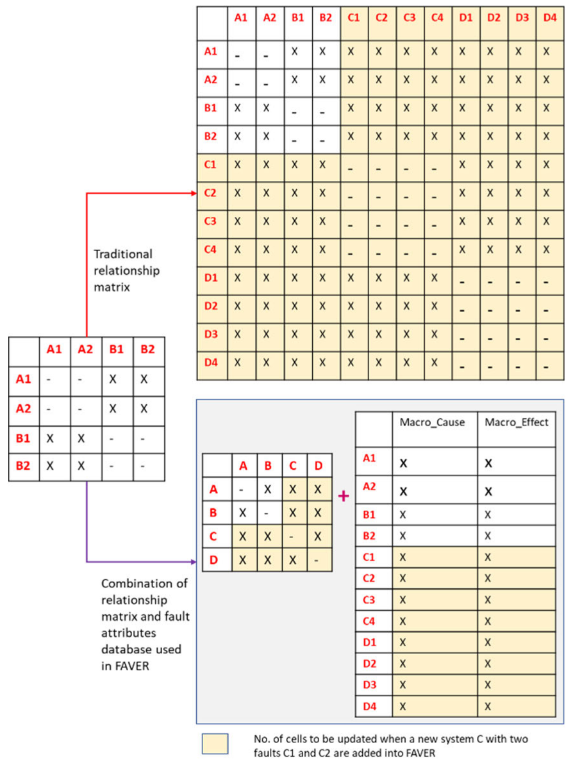


FIGURE 5. Traditional relationship matrix versus the combination of vehicle level relationship matrix with fault attributes database.

should develop the relationship for the $n + j^{th}$ system against all the elements belonging to n systems, as given by:

No. of new elements to be updated in matrix,

$$M = \sum_{i=1}^{i=n+j} (No.\ of\ faults\ in\ System\ i)^2 - \sum_{i=1}^{i=n} (No.\ of\ faults\ in\ System\ i)^2 \tag{5}$$

where, n = no. of existing systems in the framework
j = no. of new systems added to the framework

However, in the case of FAVER's expert system, when a new system, n + j, is brought into the framework, the size of the existing relationship matrix n^2 goes to $(n + j)^2$. The number of new elements to be updated in the vehicle level relationship matrix is given by j(2n + j). Experts will have to update the fault attributes database only for two attributes per fault ('Macro_Cause'; 'Macro_Effect'). Hence the total number of elements to be updated in FAVER's expert system is given by:

$$\text{No. of new elements to be updated in FAVER's expert system} = j(2n + j) + \left(\sum_{i=1}^{j} (\text{No. of faults in } i^{th} \text{ system}) * 2\right) \tag{6}$$

As seen from fig.5, 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 according to (6) are 12 + 16 = 28. This difference will be larger in practical scenarios when more aircraft systems with multiple faults are added to the framework. Thus, this arrangement of knowledge base in FAVER's expert system saves considerable time for the experts to update the knowledge base when new systems are added to the framework.

c: ADVANTAGES

This novel arrangement of knowledge is similar to the traditional relationship matrix, in the sense that it is developed by experts. However, when a new system is 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 a fault attributes database is that the knowledge that needs to be populated for a new fault is only for two attributes: {'Macro_Cause'; 'Macro_Effect'}. 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. Even when the traditional relationship matrix could use advanced formats like sparse matrix and update only the non-zero elements of the matrix, it will still take a lot of memory and time to maintain such a database and validate every relationship. In the case of a fault attributes database, adding new faults for the existing systems or adding new faults from new systems will require only two attributes for each one of them. The use of a fault attributes database rearranges the same available information regarding a fault's cause and effect, resulting in a more scalable and efficient knowledge database for FAVER.

C. COMMUNICATIONS LAYER

This 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. 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 architecture of OSA-CBM, used as a guideline, consists of Information specification (data model) and Interface specification. The data model of OSA-CBM consists of four classes: DataEvent, Configuration, Explanation, and Extension. The Interface specification of OSA-CBM has four types: Asynchronous, Synchronous, Service, and Specification [29]. In line with these specifications, the DT layer and the Reasoning layer of FAVER's architecture are mapped to the OSA-CBM data model as a Configuration data class, comprising algorithms, simulation models, domain knowledge and all the required input sources. The communication layer of FAVER interfaces with the user and constitutes the data belonging to DataEvent class of OSA-CBM data model, comprising of input parameters and output results. As FAVER is in the initial stages of implementation, Explanation and Extension classes are not used in FAVER. A synchronous interface is adapted from the Interface specification of OSA-CBM to return the required data to the calling functions. The functions are developed in MATLAB and data are stored as MS Excel files. As and when FAVER is adapted for real-world industrial applications all data will be converted to XML formats to align fully with OSA-CBM standards.

D. ILLUSTRATION OF FAVER ARCHITECTURE

An illustration of FAVER's architecture showcasing the characteristics of each layer described in this section, along with the various information flowing between them, and the steps to expand the framework are shown in fig.6. In Stage 1 basic functionality is established, while Stage 2 shows how the framework is expanded to accommodate new systems. The faults for all four aircraft systems are selected based on Failure Mode and Effect Analysis (FMEA) carried out for each aircraft system, and categorized in one of two classes:

i. A faults that leads only to local (or micro) effects,
ii. A fault that leads to global (or macro) effects.

Table 2 shows the FMEA carried for the faults chosen for FAVER demonstration. A thorough FMEA for all faults in the four aircraft systems is not feasible, due to the lack of openly available knowledge of aircraft failure modes. Instead, only a subset of faults is chosen for FMEA in such a way that each system has a certain number of faults with micro and macro effects affecting the systems in their interaction network. The faults with only local and system level effects are considered single faults with micro effects and the faults with global effects are considered as interacting faults with cascading effects. This is done to help demonstrate the capabilities of isolation of both single and interacting faults by FAVER's reasoning and using the vehicle level knowledge to isolate their root causes and predict their cascading effects in their interaction network.

1) STAGE 1

The three layers of FAVER are set up and are tested against their requirements from table 1 using two systems: i) the EPS, and the ii) Fuel system, as seen in the middle portion of fig.6.

The chosen representation of the EPS DT is a B737 model developed in Matlab Simulink, with an ANFIS diagnostics capable of isolating three fault modes: i) ***AC Instru***

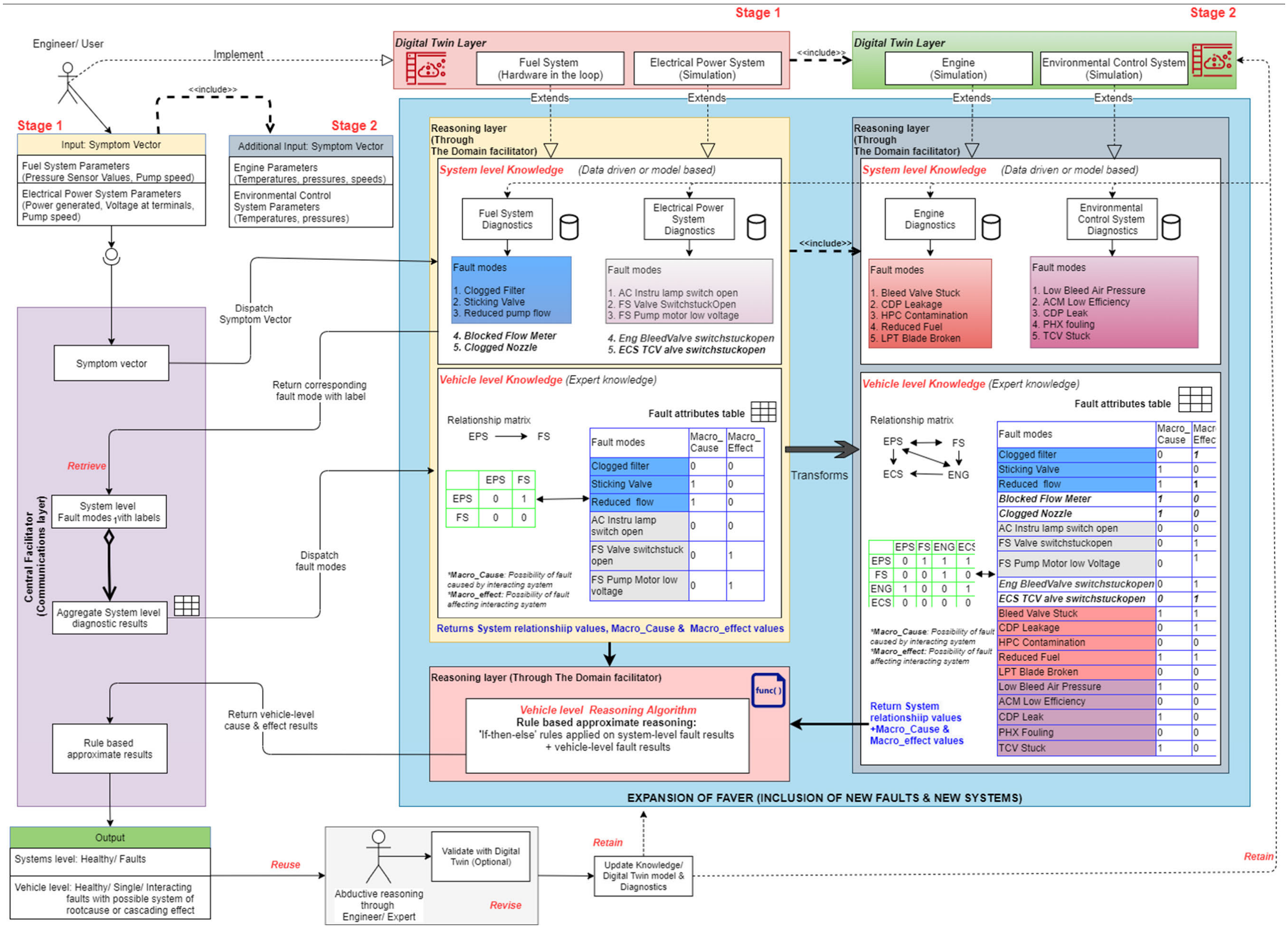


FIGURE 6. An illustration of activities in FAVER architecture.

lamp switch open, ii) ***FS Valve switch open***, and iii) ***FS Pump motor low voltage***. The EPS interacts with all the systems under consideration by powering their pumps and valves. It also receives shaft power from the engine and this reaction will be considered in Stage 2 when the engine is included in the framework. From table 2, it can be seen that the fault mode, ***AC Instru lamp switch open*** does not have any Global effect on other systems and hence, it is considered single {Macro_Cause = 0; Macro_effect = 0}. On the other hand, ***FS Valve Switch Open*** has the global effect of interfering with the valve opening in the fuel system {Macro_Cause = 0; Macro_effect = 1} and ***FS Pump motor low voltage*** reduces the input voltage to fuel pump {Macro_Cause = 0; Macro_effect = 1}. Further details on these fault modes and diagnostics can be found in the background work [23].

The fuel system is represented by an experimental rig included in the framework as Hardware-in-the-loop, with a Decision tree diagnostics chosen to isolate three fault modes: i) ***Clogged filter***, ii) ***Sticking Valve***, and iii) ***Reduced flow***, which are detailed in the background work [20]. The fuel system supplies fuel to the engine, and its pump and valves are powered by the EPS. The fault modes ***Sticking Valve*** and ***Reduced flow*** are chosen as they could be of mechanical origin or could have been caused by ***FS Valve Switch Open*** and ***FS Pump motor low voltage*** in the EPS respectively {Macro_Cause = 1; Macro_effect = 0}, thus resulting in ambiguity. While the fault mode ***Reduced flow*** would automatically affect the engine, since the engine is not considered in Stage 1, its Macro_effect attribute is considered '0' at this stage. It will later be updated when the engine is included in the framework. The fault mode ***Clogged filter*** is simulated to obstruct the fuel flow; the effect of this degradation over the EPS is considered to be negligible and can be updated later {Macro_Cause = 0; Macro_effect = 0}.

The above mentioned interactions between the EPS and the fuel system, and their attribute values for each fault modes are populated in the form of relationship matrix and the fault attributes database, forming the vehicle level knowledge in the reasoning layer, as seen in the middle portion of fig.6. The EPS and the fuel system chosen are of different scales, they do not interact with each other directly but only through the DT facilitator (where the interaction is converted to 0-100% scale) and use different types of system level diagnostics to ensuring the modularity, showcasing the flexibility of FAVER's architecture.

The middle portion of fig.6 also shows the vehicle level reasoning algorithm that is used to isolate the cascading faults and their root causes. Once the modules in the reasoning layer and the DT layer are set up, the requirement of FAVER can be tested, to isolate single faults in the EPS and the fuel system, and interacting faults between these systems, resulting in ambiguity. For this, the user has to generate and input the symptom vector under investigation, into the dialog box generated by the Central facilitator (as seen in the left portion of fig.6). The Central facilitator then follows case based reasoning methodology, passes on the symptom vector to the reasoning layer and after a series of steps, shown in the left and middle portions of fig.6, provides the output to the user for the final step of abductive reasoning.

Thus, in Stage 1, the architecture of FAVER is set up and demonstrated for its flexibility and its ability to isolate single and interacting faults in presence of ambiguity.

2) STAGE 2

In stage 2 of development of FAVER's architecture, the framework is tested for its ability to expand with ease, to retain new knowledge for existing systems and to include new aircraft systems for investigation, as shown in the right portion of fig.6. This is to satisfy the requirement of scalability 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. It is not possible to include all aircraft systems in this work, due to time constraints and logistics constraints, hence testing of FAVER for scalability is necessary. While FAVER uses two aircraft systems (the FS and the EPS) in Stage 1, scaling up the framework in Stage 2 to four systems (adding engine and ECS), is necessary to achieve health assessment with a holistic view of the aircraft and demonstrate scalability.

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 (as seen in Stage 1) it has to adapt to process new sets of faults and new knowledge from these aircraft systems, without much changes to its structure, in order to isolate their root causes and predict cascading effects.

FAVER can be expanded in the following ways:

a: ADDING NEW FAULTS TO EXISTING SYSTEMS

Given that the respective system DTs and their diagnostics are updated to provide and isolate the symptom vectors, the corresponding new faults can be added to the framework just by updating the fault attributes database. Two new fault modes are added to both the EPS and the fuel system. These are highlighted in middle portion of fig.6, under each system's fault modes. For the EPS, the new fault modes added are i) ***Eng BleedValve switch stuck open*** and ii) ***ECS TCValve switch stuck open.*** From table 2, it can be seen that the former fault mode can affect the power supplied to the bleed valve in the engine, whereas the latter can affect the power supplied to the temperature control valve in the ECS. Hence, the attribute values for both these faults are {Macro_Cause = 0; Macro_effect = 1}. The new fault modes added to the fuel system are i) ***Blocked Flow Meter*** and ii) ***Clogged Nozzle***. Both these faults could either be of mechanical origin or could have been caused by any fault in the power supplied by the EPS. Hence, their attribute values are {Macro_Cause = 1; Macro_effect = 0}. Once these attribute values are updated for all four faults along with their fault codes in the fault attributes database, FAVER's reasoning will be capable of isolating these faults, isolating their root causes and warning of their cascading effects at the vehicle level.

b: ADDING NEW SYSTEMS TO THE FRAMEWORK

As seen in the right portion of fig.6, for the new systems, the engine and the ECS to be added to the framework, both these systems require their standalone DTs and system diagnostics. Similar to stage 1, the DTs and fault modes with diagnostics are chosen from the previous work [20].

The engine is represented by a Pratt and Whitney JT9D model developed in Matlab Simulink with five fault modes chosen: i) ***Bleed Valve Stuck***, ii) ***CDP Leakage***, iii) ***HPC Contamination***, iv) ***Reduced Fuel***, and v) ***LPT Blade Broken;*** a Linear Discriminant Analysis algorithm is developed to isolate these faults. As the engine interacts with all four systems under consideration in FAVER, it raises several ambiguous scenarios. Fault mode ***Bleed Valve Stuck*** could be of mechanical origin located in the engine itself or it could have been caused to the fault mode ***Eng BleedValve switch stuck open*** from the EPS, giving raise to ambiguity. This fault could affect the ECS by obstructing the bleed supply. Hence, its attribute values are {Macro_Cause = 1; Macro_effect = 1}. Customer Discharge Pressure ***(CDP) leakage*** is simulated as a mechanical leak affecting the bleed supply to the ECS, and its attribute values are {Macro_Cause = 0; Macro_effect = 1}. Fault mode ***Reduced Fuel*** is simulated to reduce the output of engine and thus it affects all the systems it interacts with. This fault mode is caused by ***Reduced flow*** from the fuel system. Hence, its attribute values are {Macro_Cause = 1; Macro_effect = 1}. Fault modes High Pressure Compressor (***HPC) Contamination*** and Low Pressure Turbine ***(LPT) Blade Broken*** are simulated to reduce the efficiency of the engine but not affect its output, and they are of mechanical origin, thus having attribute values {Macro_Cause = 0; Macro_effect = 0}.

The ECS is represented by a B737-800 model developed in a simulation package called SESAC [22]. Five fault

TABLE 2. Failure Mode and Effect Analysis (FMEA) for faults in four aircraft systems.

Fault Codes	Origin_ System	Mode (Cause)	Local Effect	System effect	Global effect
AC_Motor_Low _Voltage	EPS	FS Pump Motor Low Voltage	Affects the power supplied to FS Pump	No effect to the performance of EPS	Reduced speed of FS Pump leading to reduced output from FS. Depending on the level of degradation, reduced performance of all connected systems: ENG
InstruLamp	EPS	AC instrument Lamp switch open	Disconnects the AC instrument Lamp	Lack of indication from the instrument lamp for the corresponding instrument	No effect on other systems
FSN	EPS	FS Nozzle switch open	Disconnects power supplied to FS Nozzle	No effect to the performance of EPS	FS Nozzle stuck open resulting in clogged nozzle in FS
FSV	EPS	FS Valve switch open	Disconnects power supplied to FS Valve	No effect to the performance of EPS	FS Valve stuck open resulting in Sticking valve in FS
EBV	EPS	ENG Bleed valve switch open	Disconnects power supplied to ENG bleed valve	No effect to the performance of EPS	ENG Bleed valve stuck open not responding to Control System demand
EPV	EPS	ECS TCV switch open	Disconnects power supplied to ECS TCV	No effect to the performance of EPS	ECS TCV stuck open not responding to Control System demand.
CF	FS	Clogged Filter	Filter slows down the fuel flow increasing the pressure in the surrounding area of the fuel line	Clogged filter leading to FS Pump overworking to meet the Control system demand of fuel supply	No effect as long as FS Pump can meet the Control System demand for fuel supply. Might increase demand on EPS on extreme cases.
CN	FS	Clogged Nozzle Mechanical	Nozzle slows down the fuel flow increasing the pressure in the surrounding area of the fuel line	Clogged Nozzle leading to FS Pump overworking to meet the Control system demand of fuel supply	No effect as long as FS Pump can meet the Control System demand for fuel supply. Might increase demand on EPS on extreme cases.
BFM	FS	Blocked Flow meter	Flow meter slows down the fuel flow due to blockage, increasing the pressure in the surrounding area of the fuel line	Blocked Flow meter leading to FS Pump overworking to meet the Control system demand of fuel supply	No effect as long as FS Pump can meet the Control System demand for fuel supply. Might increase demand on EPS on extreme cases.
SV	FS	Sticking Valve Mechanical	Valve slows down the fuel flow increasing the pressure in the surrounding area of the fuel line	Sticking Valve leading to FS Pump overworking to meet the Control system demand of fuel supply	No effect as long as FS Pump can meet the Control System demand for fuel supply. Might increase demand on EPS on extreme cases.
RF	FS	Degraded Pump	FS Pump not able to operate to meet demand	FS not able to meet the Control System demand of fuel supply	Reduced fuel input to Engine. Depending on the level of degradation, reduced performance of all connected systems: ECS, EPS

modes are chosen for its integration to FAVER: i) ***Low Bleed Air Pressure***, ii) ***CDP Leak***, iii) ***ACM Low Efficiency***, iv) ***PHX Fouling***, and v) ***TCValve stuck***. The ECS interacts mainly with the engine for the bleed air supply and its Temperature Control Valves (TCV) are powered by the EPS. Its output affects the cabin pressure directly. Fault modes ***Low Bleed Air Pressure*** and Customer Discharge Pressure ***(CDP) Leak*** are simulated to have been caused by the engine faults ***Bleed Valve Stuck*** and ***CDP Leakage*** respectively and hence their attribute values are {Macro_Cause = 0; Macro_effect = 1}. As seen in table 2, fault modes Air Cycle Machine (***ACM) Low Efficiency*** and Primary Heat Exchanger (***PHX) Fouling*** are of local origin and their attribute values are {Macro_Cause = 0; Macro_effect = 0}. Fault mode Temperature Control Valve ***(TCValve) stuck*** could either be a local mechanical fault or it could have been caused by the fault mode ***TCValve switch stuck open*** from the EPS and hence its attribute values are {Macro_Cause = 1; Macro_effect = 0}.

In order to add these aircraft systems and their faults to the framework, the abovementioned faults and their attributes for both the engine and the ECS must be added to the fault attributes table. The Reasoning layer has to be given access to the system diagnostics and, as seen in the left portion of fig.6, the Central Facilitator should be updated to fetch the symptom vector from these systems to make the final vehicle symptom vector as the input. These are the only changes to FAVER's architecture to expand and include more systems into the framework. The run-time of the vehicle level reasoning algorithm to isolate faults at the vehicle level is not majorly affected by the number of systems included in the framework, due to modularity, thus proving that FAVER is easily scalable without much penalty on its performance.

c: UPDATING NEW KNOWLEDGE TO THE FRAMEWORK

One of the main requirements of FAVER is the ability to update and retain new knowledge on existing database. Consider the fault mode ***Reduced flow*** in the fuel system, added to the framework in Stage 1. At that point, the engine is not included in FAVER and hence its effect on the engine is not considered. After the engine is added in stage 2, the ***Reduced flow*** fault mode should be updated to reflect its cascading effect on the engine, resulting in ***Reduced fuel*** mode in the engine. Hence the only change needs to be done is to change its attribute values from {Macro_Cause = 1; Macro_effect = 0} to {Macro_Cause = 1; Macro_effect = 1}. Another example of updating knowledge is the ***Clogged filter*** mode in the fuel system. The effect of this fault mode on the EPS is considered to be negligible initially. However, if the experts find more evidence that ***Clogged filter*** seems to affect the EPS considerably, the relationship matrix should be updated to reflect that the fuel system affects the EPS and the attributes in fault attributes database should be updated as {Macro_Cause = 0; Macro_effect = 1}. Only these changes by the experts are sufficient for FAVER's reasoning algorithm to isolate the root causes and identify the cascading effects of

TABLE 2. *(Continued.)* **Failure Mode and Effect Analysis (FMEA) for faults in four aircraft systems.**

Fault Codes	Origin_ System	Mode (Cause)	Local Effect	System effect	Global effect
LPT BladeBroken	ENG	LPT Blade Broken	Reduced performance efficiency of LPT	No effect as long as the overall performance of LPT is not affected severely	No effect as long as the overall performance of Engine is not affected severely
LPC Fouling	ENG	LPC fouling	Reduced performance efficiency of LPC	No effect as long as the overall performance of LPC is not affected severely	No effect as long as the overall performance of Engine is not affected severely
HPT BladeBroken	ENG	HPT Blade broken	Reduced performance efficiency of HPT	No effect as long as the overall performance of HPT is not affected severely	No effect as long as the overall performance of Engine is not affected severely
Bleed Valve Stuck	ENG	ECS Bleed Valve Stuck Mechanical	ENG Bleed Valve not responding to the Control System	ENG Bleed air System not able to supply demanded Bleed air	Reduced bleed air supplied to the ECS
HPC Contamination	ENG	HPC Contamination	Reduced performance efficiency of HPC	No effect as long as the overall performance of HPC is not affected severely	No major effect as long as the overall performance of Engine is not affected severely
Fan FOD	ENG	Fan FOD	Reduced performance efficiency of Fan	No effect as long as the overall performance of Fan is not affected severely	No major effect as long as the overall performance of Engine is not affected severely
CDP Leakage	ENG	Leakage in CDP duct	Bleed air supplied at reduced pressure	ENG Bleed air System not able to supply demanded Bleed air	Bleed air supplied to ECS at reduced pressure than demanded CDP. Depending on the level of degradation, might not meet the demand from cabin pressure
PHXFouling	ECS	PHX Fouling	Reduced performance efficiency of PHXFouling	No effect as long as the overall performance of PHX is not affected severely	No major effect as long as the overall performance of ECS is not affected severely
SHXFouling	ECS	SHX Fouling	Reduced performance efficiency of SHXFouling	No effect as long as the overall performance of SHX is not affected severely	No major effect as long as the overall performance of ECS is not affected severely
ACM	ECS	ACM low efficiency	Reduced performance efficiency of ACM	No effect as long as the overall performance of ACM is not affected severely	No major effect as long as the overall performance of ECS is not affected severely
TCVStuck	ECS	TCV Stuck Mechanical	TCV not responding to the demand from the Control System	Temperature of Bleed air not regulated by the ECS	Bleed air temperature not regulated as per the demand from the Cabin

these faults on the systems at the vehicle level. This helps in keeping the knowledge base of the expert system updated in FAVER in times of evolving scenarios, with relative ease.

IV. FAVER's REASONING: USE CASE DEMONSTRATIONS

While section III detailed about FAVER's architecture and its key features of flexibility, expandability and scalability, this section demonstrates the strength of FAVER's reasoning.

The interactions between the EPS, the fuel system, the engine, and the ECS in the aircraft are substantial from the operations, maintenance, and safety perspective [30]. Accident events, like the engine rollback on a B777 as a result of fuel starvation [14], [15], the engine inflight shutdown in an experimental aircraft due to a burnt electric fuel boost pump [31], and an emergency descent due to cabin pressurisation resulting from engine bleed air system failure [32], 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 [23], [30]. 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 fig.6 is used for demonstrations in this section.

A. ACCIDENT SCENARIO 1

In line with the incidents of engine rollback caused by a blockage in the fuel system [14] and [15], and an engine inflight shutdown due to a burnt fuel boost pump resulting in fuel starvation [31], 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. Fig.7 shows the development of the use case scenario as four stages 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.

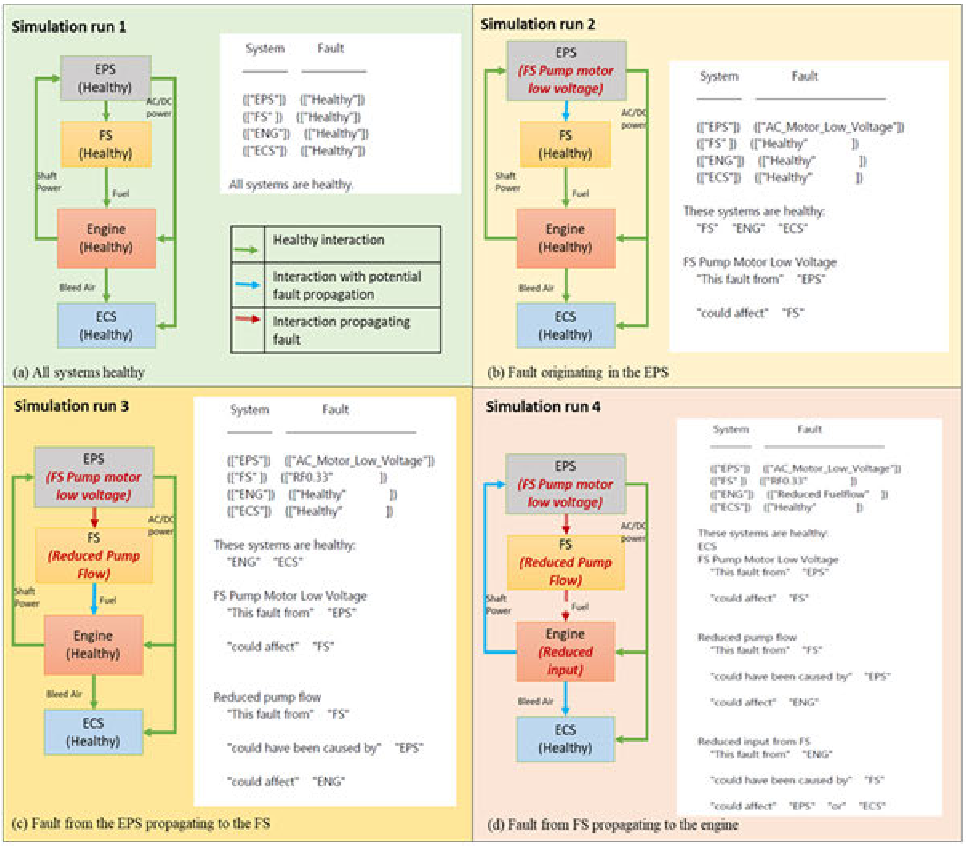
Stage 1: All systems are healthy.

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

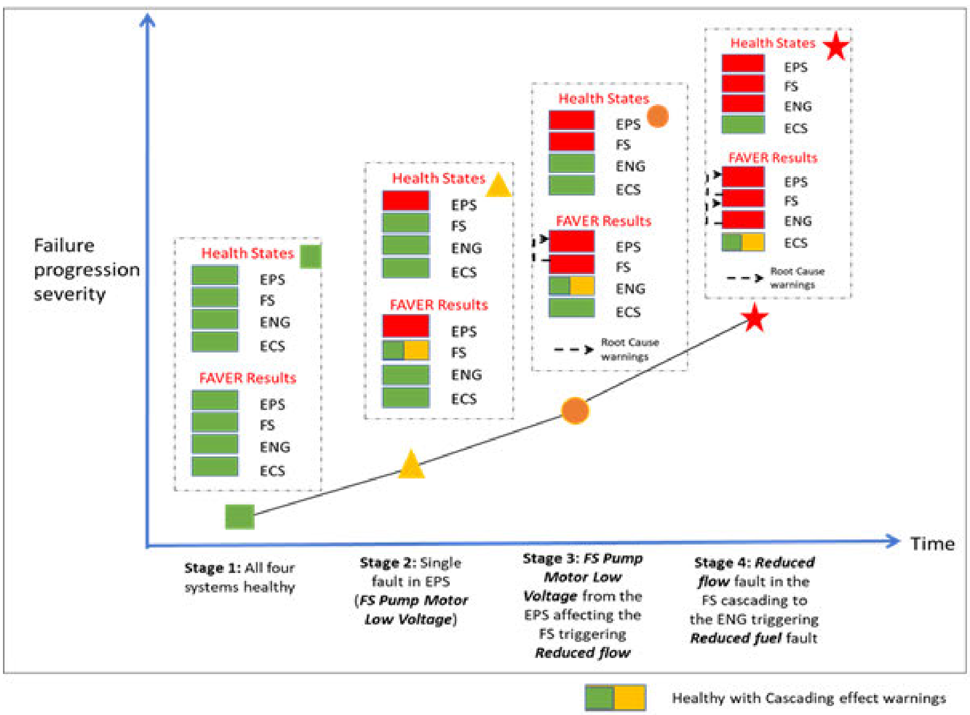
Stage 3: ***FS Pump Motor Low Voltage*** from the EPS affecting the fuel system (FS) triggering ***Reduced flow*** fault, and its effect has not transpired to the interacting systems yet.

Stage 4: ***Reduced flow*** fault in the fuel system cascading to the engine (ENG) triggering ***Reduced fuel*** from the fuel system fault, and the ECS remains healthy.

The reasoning results of abovementioned stages are aggregated as one figure (fig.7(a)) to show how FAVER's reasoning will isolate faults, their root causes and predict their cascading effects during each stage of the fault propagation. The left side of each simulation run in fig.7(a) shows the current state of systems through diagrams. In these diagrams, the healthy interactions in simulation runs are indicated by



(a) Results from FAVER's Central facilitator for every simulation run



(b) Progression of fault versus the results from FAVER's Central facilitator at every stage of fault propagation

FIGURE 7. Use case scenario of fault originating in the EPS propagating to the FS and the engine.

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 stage 1, where four systems are healthy. In the next stages, the simulations are run with fault injected in the systems. The right side of each simulation run in fig.7(a) shows the corresponding results from FAVER's reasoning displayed by the Central Facilitator. Fig.7(b) shows the overall development of fault propagation scenario versus time, indicating the current health states of all four aircraft systems and FAVER's prediction of cascading faults and identification of root causes at each stage of fault propagation.

Simulation run 1 of fig.7 (a), (b) 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 (fig.3). The fault is detected as being in the EPS, and all other systems are judged healthy by the Central facilitator. This is shown by the first six lines of the Central facilitator result for Simulation run 2 in fig.7(a). Physically, this low voltage input to the fuel system AC motor pump could result in reduced rotor speed and torque of the pump (simulation details can be found in previous work [23]). This fault would essentially affect the fuel pump as indicated by the blue colored line in fig.7(a). 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 predicts the possible cascading effect to the fuel system through the Central facilitator, last three lines in fig.7(a). The same is highlighted in fig.7(b), where for stage 2, the EPS is highlighted in red colour for being affected by a fault and the fuel system (FS) is highlighted in amber colour as FAVER predicts against this system for potential cascading effect, Having stepped through this case in detail, the other cases are described more briefly.

Simulation run 3 shows the next potential stage of fault propagation, where the fault ***FS Pump Motor Low Voltage*** affects the fuel system. Here, the vehicle symptom vector input consists of ***FS Pump Motor Low Voltage*** fault in the EPS, and ***Reduced flow* in** the fuel system, and the other two systems are healthy. 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 in fig.7(a)). Further, the possibility of ***Reduced flow*** from the fuel system 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. This is highlighted in fig.7(b), where, for stage 3, the EPS and the FS are highlighted in red colour for being affected by faults, the ENG is highlighted in amber colour for the potential of being affected as a cascading effect, and the possible root cause for the fault is FS is mapped to the EPS.

In the final stage of fault propagation of this use case scenario, the ***Reduced flow*** fault from the fuel system provides a lower fuel flow to the engine; this behaviour is translated to the engine DT by the ***Reduced fuel*** fault mode.

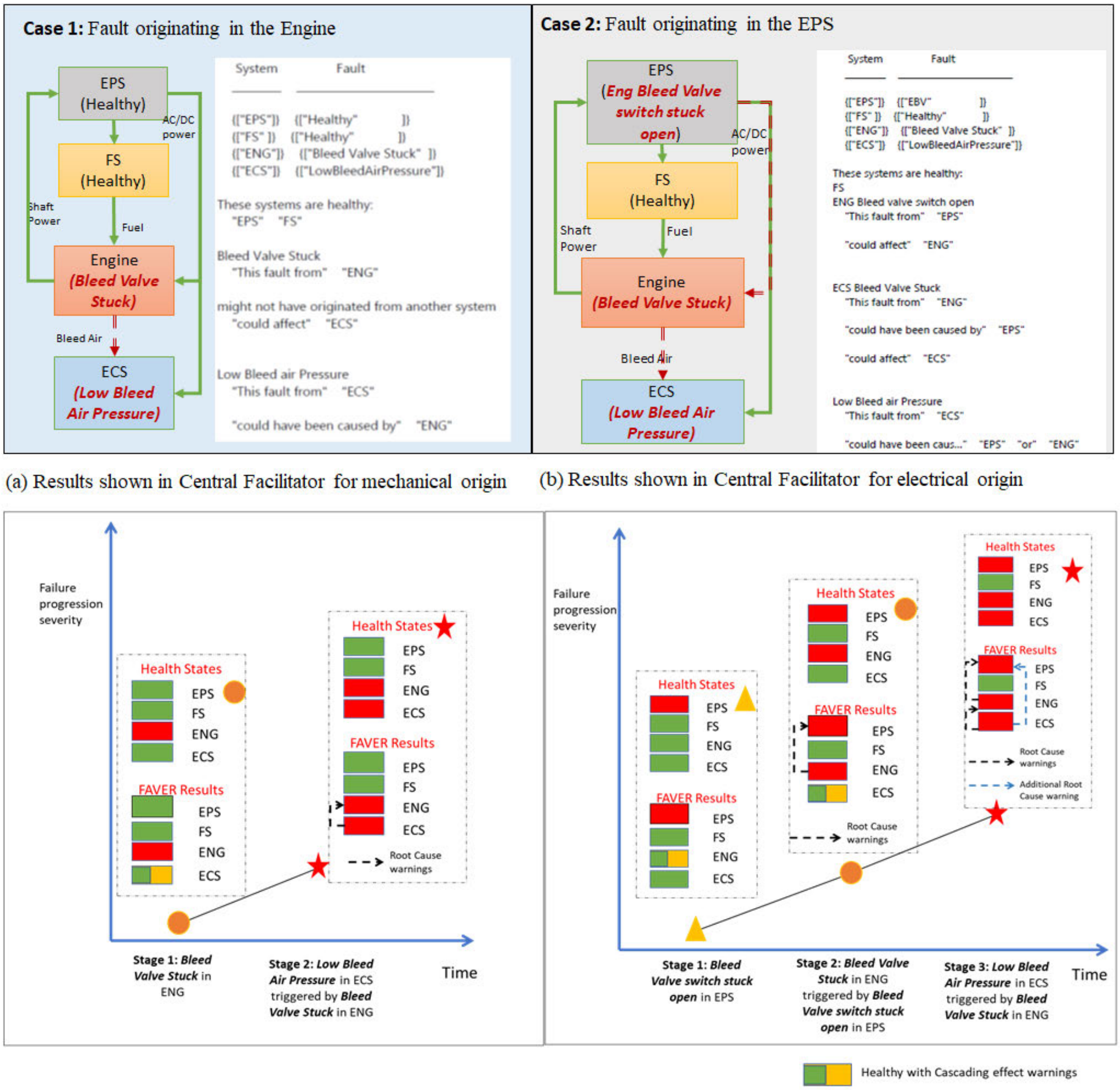
The resulting symptom vector is input to the Central facilitator, and simulation run 4 shows the result. The Diagnostic algorithms return a fault in 3 systems to the Central facilitator, with only the ECS being healthy.

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, as seen in the RHS of Simulation run 4 in fig.7(a). The same is highlighted in fig.7(b). It should also be noted that, as seen in fig.7(a), the engine's cascading effect prediction 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 LHS of simulation run 4 in fig.7(a)).

B. ACCIDENT 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 depressurisation, and oxygen masks were deployed for the passengers [32].

Two use cases are developed similar to this accident scenario. A ***Bleed Valve Stuck*** fault is triggered in the engine DT, through both mechanical (local) origin and electrical origin by ***Eng BleedValve switch stuck open*** fault mode from the EPS; the 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. Fig.8 shows the results of the bleed valve fault through two different origins. In case 1, the ***Bleed Valve Stuck*** fault is found in the engine and 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 in fig.8(a) identifies the root cause of the *Low Bleed Air Pressure* fault in the ECS to be originated from the engine. This is expanded in fig.8(c), where the affected engine is highlighted in red for stage 1 and the ECS is highlighted to be affected in stage 2. (Only the result of simulation run 2 is shown in fig.8(a) for brevity). In case 2 (fig.8(b)), the ***Eng BleedValve switch stuck open*** fault mode



(a) Results shown in Central Facilitator for mechanical origin

(b) Results shown in Central Facilitator for electrical origin

(c) Fault progression versus FAVER's results for mechanical origin

(d) Fault progression versus FAVER's results for electrical origin

FIGURE 8. Use case scenario of bleed valve stuck fault cascading to the ECS with mechanical and electrical origin.

is found in the EPS and is warned by FAVER to affect the engine. The Central facilitator in fig.8(b) shows the potential root cause of the ***Bleed Valve Stuck*** fault in the engine is from the EPS, and it also shows the possible cascading effect to be found in the ECS. As the ECS has interactions with two systems which has interaction faults, the approximate reasoning of FAVER identifies the potential root cause of ***Low Bleed Air Pressure*** fault to be found in either of the two systems (engine or the EPS). The different stages involved in the fault progression for electrical origin of this fault scenario is seen in fig.8(d) and the results of FAVER at each stage are highlighted along with the potential and additional root cause warnings from FAVER.

Thus, this section demonstrates the ability of FAVER's reasoning to isolate single and interacting faults, as well as resolve ambiguities by highlighting the original root cause of such interacting faults. It can be seen from both the accident scenarios that FAVER is able to predict the potential cascading effects at every stage of fault propagation along with identifying their root causes at the vehicle level. These results could be used by the engineers to troubleshoot and resolve such scenarios during maintenance, well in advance and in a shorter time period.

V. PROS AND CONS OF FAVER'S REASONING

This section presents some of the pros and cons of FAVER's reasoning.

A. 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 which fault is the root cause in case of interaction faults. Fig.9 shows four different scenarios where FAVER can differentiate interacting faults from simultaneous single faults, clearing ambiguity.

Case 1 in fig.9 shows the scenario of an interaction fault with three of the four systems involved while the fuel system has a single fault simultaneously. The interaction fault in Case 1 is the electrical origin case of accident scenario 2, where the ***Bleed Valve Stuck*** fault in the engine, caused by ***Eng Bleed Valve switch stuck open*** fault mode from the EPS resulting in ***Low Bleed Air Pressure*** in the ECS. 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 fig.9. 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.

Another scenario is tested, as shown in case 2 in fig.9, 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 fuel system. Simultaneously, a leak in the 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 fig.9 shows that FAVER's reasoning can differentiate both of interacting faults and suggest their possible root causes and cascading effects.

In case 3 of fig.9, 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 fig.9 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 fig.9, one interaction fault starting from the EPS affects all the other systems. This use case is a projection of the accident scenario 1, demonstrated in the previous subsection and seen in fig.9. While in the previous subsection, the ECS is shown as healthy, with the potential of ***Reduced fuel*** 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 case 1 in fig.9, 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.

B. COST OF USING THE NOVEL COMBINATION OF KNOWLEDGE

In cases 1 and 4 of fig.9, 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.

C. 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.

D. METRICS FOR EVALUATING FAVER

Since FAVER is a layered reasoner, it is not possible to evaluate the performance of FAVER by commonly used metrics like fault detection rate and fault isolation rate. Hence, evaluation metrics developed for reasoners, such as diagnostic coverage, prognostic coverage, accuracy of inferences, latency in making inferences, and sensitivity to different faults and degradation conditions should be considered [33].

1) DIAGNOSTIC COVERAGE

A reasoner's effectiveness is measured by either testing its ability to isolate test cases that cannot be otherwise isolated by existing approaches or it can be measured by improvement in latency, accuracy and sensitivity of the faults can be isolated by existing approaches [33]. FAVER's effectiveness is measured by its ability to identify the root causes of the interacting faults. For this purpose, a comparative study with data-driven reasoning is performed using supervised neural networks.

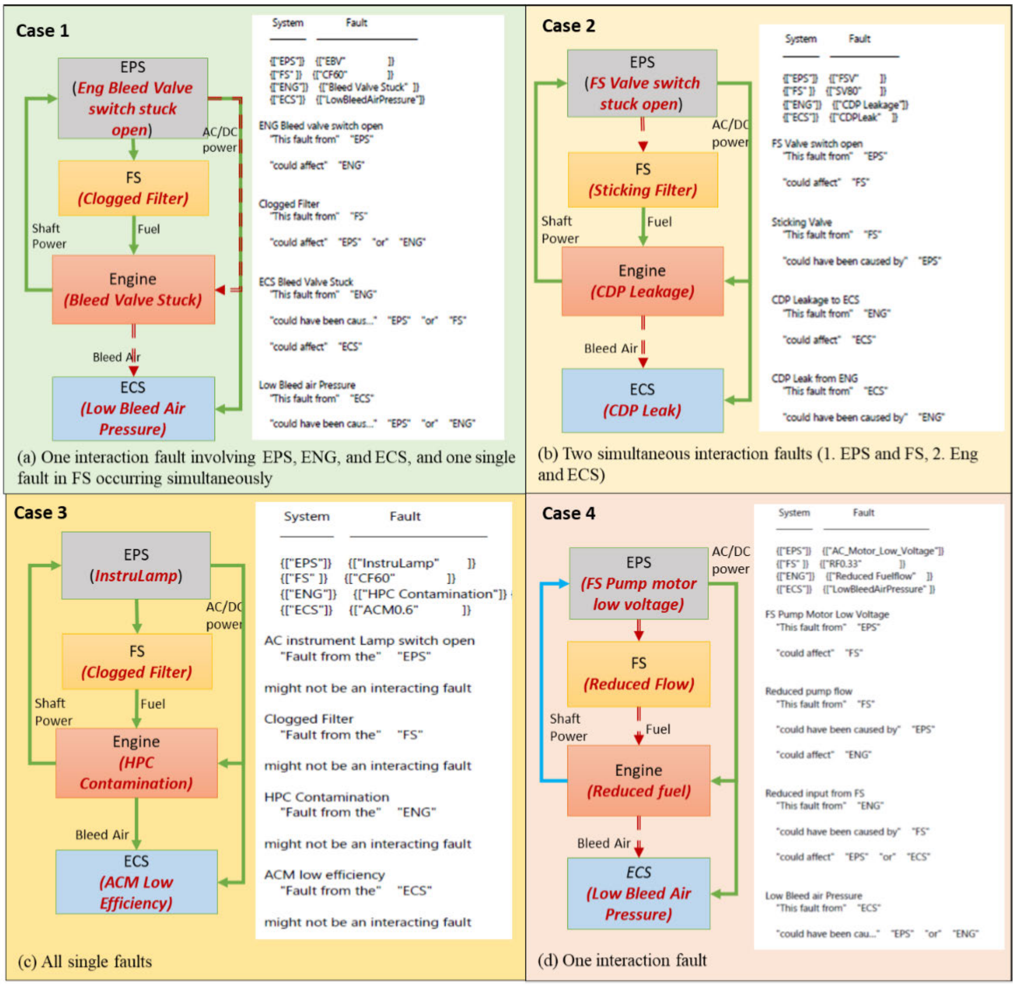


FIGURE 9. Different multiple fault scenarios (some resulting in ambiguity) reasoned by FAVER.

a: USE CASES DEMONSTRATION WITH NEURAL NETWORKS

To evaluate FAVER's reasoning, a few use case scenarios were 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 3.

The neural network is trained using the Levenberg-Marquardt algorithm [34] 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 fig.10. 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 customisation of the neural network is deemed unnecessary at this point.

EVALUATION CRITERIA 1: ACCURACY OF DIAGNOSTICS AT SYSTEM LEVEL

The data-driven reasoning to identify if a symptom vector belongs to an output class using neural networks has an advantage over FAVER's reasoning, which is, its independence of system diagnostic functions. The trained neural network was able to detect the faults at system level with 100% accuracy without system level diagnostic functions. The output classes in this demonstration are only trained to identify if a system is healthy or faulty, 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.

On the other hand, FAVER uses system level diagnostic functions to make vehicle level assessments. The accuracy of system level diagnostics used in the case studies is above 99% for all four systems [20], [23]. Inheriting these accuracy levels, FAVER's reasoning can isolate faults at the systems level with an accuracy of above 99%. The vehicle symptom vectors from the system diagnostics required only 20 parameters to carry out system level reasoning in FAVER, whereas, the trained neural network in fig.10 required data

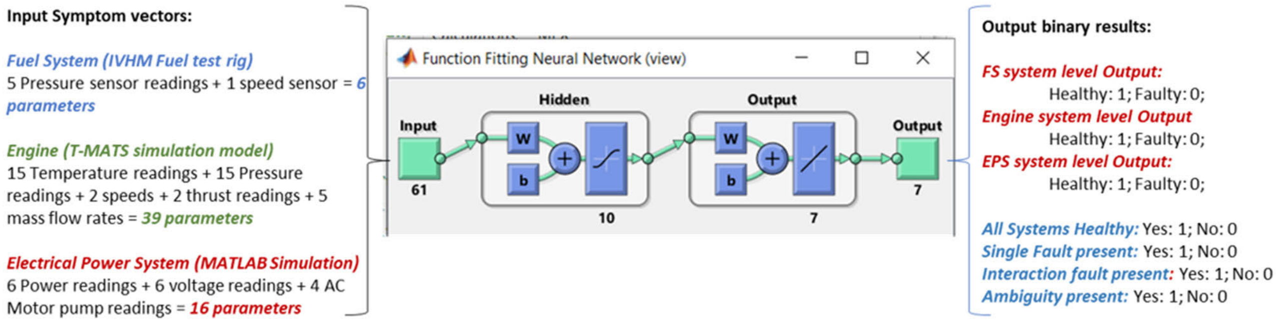


FIGURE 10. Neural network fitting to isolate faults at the vehicle level.

TABLE 3. Use case scenarios for neural network training.

Scenarios	Electrical Power System	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

from 61 parameters, which is three times when compared to FAVER's requirements.

EVALUATION CRITERIA 2: ACCURACY OF DIAGNOSTICS AT VEHICLE LEVEL

At the vehicle level, the results produced by the data-driven reasoning cannot be compared directly with FAVER's reasoning. This is because the neural networks can highlight the presence of single or interacting faults and ambiguities. In contrast, with the help of expert system knowledge, FAVER's reasoning can solve ambiguity and isolate the root cause of an interacting fault.

EVALUATION CRITERIA 3: ACCURACY OF INFERENCES

FAVER can isolate all single faults, interacting faults and clear ambiguities effectively. However, in certain scenarios where there are two or more systems interacting directly with the system containing faults, the results include all the systems in the interaction network, rather than narrowing down to a single system. This, however, is compensated by abductive reasoning.

One main advantage 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, paving way for explainable AI in vehicle health reasoning.

EVALUATION CRITERIA 4: EASE OF IMPLEMENTATION

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 data-driven reasoning, where every change requires careful consideration for readjusting its datasets. With FAVER's architecture, any type of systems DT modeled at any scale and diagnostics with any level of granularities developed by different engineers can be brought in the framework for the investigation of vehicle level health. This is simply not possible with any type of data driven reasoning.

To summarise, while FAVER's reasoning is not 100% accurate for isolating the root causes and predicting cascading effects, but 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 in making inferences. 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. Besides, it can be seen from the case study that the neural network required 3X more number of parameters, which would make it difficult for implementation by the aircraft manufacturers, whereas, FAVER's requirement of less number of sensors to produce explainable results for maintainers provides it with an added advantage towards the ease of implementation in the future.

2) PROGNOSTIC COVERAGE

Similar to diagnostic coverage, a reasoner can either be evaluated for its effectiveness in predicting scenarios that cannot be predicted by other existing approaches or it can be evaluated for the improvement in accuracy, precision, and sensitivity when compared to the existing approaches [33]. Currently, FAVER does not carry out traditional remaining useful life prediction for the systems, to be compared with existing approaches. Instead, in the presence of an interacting fault in an aircraft system, FAVER's reasoning predicts the cascading effect on the system's interaction network, if the fault propagation continues. From the use cases demonstrated in the previous sections, it can be seen that FAVER's reasoning is capable of predicting the potential cascading effects effectively.

Latency in making inferences is not used to evaluate FAVER currently, as FAVER's reasoning is not tested on datasets with timestamps. Similarly, sensitivity to different faults and degradations are also not evaluated, as degradations of faults are not included in the analysis in the current stage.

VI. SUMMARY AND FUTURE WORKS

To summarise, the following contributions are made in this paper.

- i) The working concept behind FAVER's framework, its requirements, architecture, and the previous work done to set up the framework for the demonstrations have been described.
- ii) The strength of FAVER's architecture is demonstrated on three fronts:
  - a. The ability to accommodate different types of representations of system DTs and diagnostics.
  - b. The ability to expand to include new faults to the systems with minimal change to the framework by adding additional faults to two systems.
  - c. 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.
- iii) Real aircraft incidents are 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 is carried out in two stages:
  - a. The advantages and disadvantages of using FAVER's reasoning are demonstrated with examples.
  - b. FAVER's reasoning and data-driven reasoning (neural networks) are compared, and the pros and cons are 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. Also, FAVER can be upgraded to include the degradation levels of faults while reasoning at both system and vehicle levels. The results of FAVER will be adapted to OSA-CBM standards for its future applications in industry. 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 relying 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.

ACKNOWLEDGMENT

The authors would like to thank Dr. Zakwan Skaf for his valuable input during the initial conceptualization phase of this work. They would like to thank Boeing for their support of this project.

REFERENCES

[1] C. M. Ezhilarasu, Z. Skaf, and I. K. Jennions, "Understanding the role of a digital twin in integrated vehicle health management (IVHM)," in *Proc. IEEE Int. Conf. Syst., Man Cybern. (SMC)*, Oct. 2019, pp. 1484–1491, doi: 10.1109/SMC.2019.8914244.

[2] 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, pp. 60–73, Feb. 2019, doi: 10.1016/j.paerosci.2019.01.001.

[3] Honeywell. (2018). *Health and Usage Monitoring System (HUMS)*. Honeywell. Accessed: Apr. 21, 2021. [Online]. Available: https://aerospace.honeywell.com/en/learn/products/health-and-usage-monitoring

[4] Safran. *Health Monitoring for Military Aircrafts*. Accessed: Apr. 20, 2021. [Online]. Available: https://www.safran-electronics-defense.com/aerospace/military-aircraft/propulsion-system-solutions/health-monitoring-military-aircraft

[5] G. Bastard, J. Lacaille, J. Coupard, and Y. Stouky, ''Engine health management in Safran aircraft engines,'' in *Proc. Annu. Conf. Prognostics Health Manage. Soc., (PHM)*, 2006, pp. 101–108. [Online]. Available: https://pdfs.semanticscholar.org/a1c9/204a386439f2b22dc4f780305d5dda91121e.pdf
[6] Meggitt. (2018). *Engine Health and Condition Monitoring*. Accessed: Apr. 20, 2021. [Online]. Available: https://meggittsensing.com/aerospace/product/engine-health-and-condition-monitoring/
[7] Meggitt. *Health and Usage Monitoring (HUMS)*. Accessed: Apr. 20, 2021. [Online]. Available: https://meggittsensing.com/aerospace/product/helicopter-hums-to-prevent-failures-increase-safety-and-reduce-costs/
[8] V. N. Divakaran, R. M. Subrahmanya, G. V. V. Ravikumar. (2018). *White Paper Integrated Vehicle Health Management of A Transport. Infosys*. Accessed: Apr. 19, 2021. [Online]. Available: https://www.infosys.com/engineering-services/white-papers/Documents/aircraft-landing-gear-system.pdf
[9] Qualtech Systems Inc. (2020). *TEAMS-RT*. Accessed: Apr. 22, 2021. [Online]. Available: https://www.teamqsi.com/products/teams-rt/
[10] Qualtech Systems Inc. *TEAMS-RDS*. Accessed: Apr. 22, 2021. [Online]. Available: https://www.teamqsi.com/products/teams-rds/
[11] 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, doi: 10.1016/j.ifacol.2017.08.816.
[12] M. Daigle, A. Bregon, and I. Roychoudhury, ''Qualitative event-based diagnosis with possible conflicts applied to spacecraft power distribution systems,'' *IFAC Proc. Volumes*, vol. 45, no. 20, pp. 265–270, Jan. 2012, doi: 10.3182/20120829-3-MX-2028.00084.
[13] S. C. Ofsthun and S. Abdelwahed, ''Practical applications of timed failure propagation graphs for vehicle diagnosis,'' in *Proc. IEEE Autotestcon*, Sep. 2007, pp. 250–259, doi: 10.1109/AUTEST.2007.4374226.
[14] P. A. Sleight. (2014). *Carter RDG. Aircraft Accident Report 1/2010*. 2014. [Online]. Available: https://doi.org/EW/C2008/01/01
[15] Aero News Network. (2008). *NTSB Investigates B777 Uncommanded Engine Rollback*. Accessed: Sep. 29, 2020. [Online]. Available: http://www.aero-news.net/index.cfm?do=main.textpost&id=7cba7571-d0e2-4736-a08b-2c24fcd7f225
[16] K. Conradi. (2015). *AAIB Annual Safety Report 2015. Air Accidents Investigations Branch*. Accessed: Apr. 22, 2021. [Online]. Available: https://assets.publishing.service.gov.uk/media/5656b9e2ed915d0367000008/Annual_Safety_Report_2015.pdf
[17] E. O. Nwadiogbu, D. Mylaraswamy, S. Menon, H. C. Voges, and G. Hadden, ''Vehicle health monitoring system architecture for diagnostics and prognostics disclosure,'' U.S. Patent 008 346 429 B2, Jan. 1, 2013. [Online]. Available: https://patents.google.com/patent/US8346429B2/en
[18] 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 *Proc. Eur. Conf. Prognostics Health Manage. Soc.*, 2016, pp. 1–14.
[19] C. M. Ezhilarasu, Z. Skaf, and I. Jennions, ''Progress towards a framework for aerospace vehicle reasoning (FAVER),'' in *Proc. Annu. Conf. PHM Soc.*, vol. 11, no. 1, pp. 1–9, 2019, doi: 10.36001/phmconf.2019.v11i1.887.
[20] C. M. Ezhilarasu, Z. Skaf, and I. K. Jennions, ''A generalised methodology for the diagnosis of aircraft systems,'' *IEEE Access*, vol. 9, pp. 11437–11454, 2021, doi: 10.1109/ACCESS.2021.3050877.
[21] *Condition Monitoring and Diagnostics of Machines, International Organization for Standardization*, document ISO 13374, International Organisation for Standardisation, 2015. [Online]. Available: https://www.iso.org/standard/54933.html
[22] F. Discenzo and W. Nickerson, ''Open systems architecture enables health management for next generation system monitoring and maintenance,'' OSA-CBM Develop. Group, 2001, pp. 1–12. Accessed: Jun. 1, 2021. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.123.1206&rep=rep1&type=pdf
[23] 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, Apr. 2020, doi: 10.3390/app10082854.
[24] J. W. Chapman, T. M. Lavelle, J. S. Litt, and T. Guo. (2014). *A Process for the Creation of T-MATS Propulsion System Models From NPSS Data*. [Online]. Available: https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20150000147.pdf
[25] I. Jennions, F. Ali, M. E. Miguez, and I. C. Escobar, ''Simulation of an aircraft environmental control system,'' *Appl. Thermal Eng.*, vol. 172, May 2020, Art. no. 114925, doi: 10.1016/j.applthermaleng.2020.114925.
[26] C. B. Nielsen, P. G. Larsen, J. Fitzgerald, J. Woodcock, and J. Peleska, ''Systems of systems engineering: Basic concepts, model-based techniques, and research directions,'' *ACM Comput. Surv.*, vol. 48, no. 2, pp. 1–41, 2015, doi: 10.1145/2794381.
[27] A. Saxena, I. Roychoudhury, K. Goebel, and W. Lin, ''Towards requirements in systems engineering for aerospace IVHM design,'' in *Proc. AIAA Infotech@Aerospace (I@A) Conf.*, Aug. 2013, pp. 1–15, doi: 10.2514/6.2013-4659.
[28] S. Burge. (2011). *The Systems Engineering Tool Box*. Accessed: Apr. 18, 2021. [Online]. Available: https://www.burgehugheswalsh.co.uk/Uploaded/1/Documents/Holistic-Requirements-Model-Tool-v2.pdf
[29] T. Sreenuch, A. Tsourdos, and I. K. Jennions, ''Distributed embedded condition monitoring systems based on OSA-CBM standard,'' *Comput. Standards Interface*, vol. 35, no. 2, pp. 238–246, Feb. 2013, doi: 10.1016/j.csi.2012.10.002.
[30] T. Tokarski and A. Gębura, ''Diagnostics and technical status monitoring of electric fuel pumps for aircrafts,'' *Res. Works Air Force Inst. Technol.*, vol. 33, no. 1, pp. 237–252, Jan. 2013. Accessed: Jun. 1, 2021. [Online]. Available: https://www.researchgate.net/publication/272555057_Diagnostics_and_Technical_Status_Monitoring_of_Electric_Fuel_Pumps_for_Aircrafts
[31] National Transportation Safety Board. (2016). *NTSB Identification: CEN16LA219*. Accessed: Apr. 22, 2021. [Online]. Available: https://www.ntsb.gov/Pages/Search.aspx?k=CEN16LA219
[32] (2008). *South African Civil Aviation Authority. Aircraft Incident Report*. Accessed: Apr. 18, 2021. [Online]. Available: https://www.skybrary.aero/bookshelf/books/827.pdf
[33] X. Koutsoukos, G. Biswas, D. A. Mylaraswamy, G. D. Hadden, D. Mack, and D. Hamilton, ''Benchmarking the vehicle integrated prognostic reasoner,'' in *Proc. Annu. Conf. Prognostics Health Manage. Soc., (PHM)*, 2010, pp. 1–9.
[34] Mathworks. (2020). *Trainlm*. Accessed: Apr. 18, 2021. [Online]. Available: https://www.mathworks.com/help/deeplearning/ref/trainlm.html;jsessionid=3ad5f684b2f37dcc87231f5aa45a

CORDELIA MATTUVARKUZHALI EZHILARASU received the bachelor's degree in aeronautical engineering and the master's degree in industrial engineering from Anna University, Chennai, India, in 2011 and 2013, respectively, and the Ph.D. degree from the Integrated Vehicle Health Management (IVHM) Centre in 2020. She has worked as a Data Analysis Engineer with Bloom Energy and as a Senior Industrial Engineer with Lam Research for a period of four years. In her role as an Industrial Engineer, she has worked on various process improvement projects and developed several semi-automated tools that contributed to time and cost savings for the organization. She was a recognized ASQ Certified Six Sigma Black Belt, from 2017 to 2020. She is currently a Research Fellow with Cranfield University, working with the IVHM Centre. Her current research interests include IVHM technologies, digital twin, and machine learning related to aerospace applications.

IAN K. JENNIONS received the B.E. degree in mechanical engineering and the Ph.D. degree in computational fluid dynamics from Imperial College London. His career spans over 40 years, working mostly for a variety of gas turbine companies. He has worked for Rolls-Royce (twice), General Electric, and Alstom in a number of technical roles, gaining experience in aerodynamics, heat transfer, fluid systems, mechanical design, combustion, services, and IVHM. He moved to Cranfield in July 2008 as a Professor and the Director of the newly formed IVHM Centre. He is currently on the Editorial Board of the *International Journal of Condition Monitoring*, the Director of the PHM Society, the Chairman of SAE's IVHM Steering Group, a Contributing Member of the SAE HM-1 IVHM Committee, a Chartered Engineer, and a fellow of IMechE, RAeS, and ASME. He is an editor of five SAE books and coauthor of one book.