IJSRA

International Journal of Science and Research Archive

eISSN: 2582-8185

Cross Ref DOI: 10.30574/ijsra

Journal homepage: https://ijsra.net/

Int. J. Sci. Res. Arch.
IJSRA
International Journal of Science and Research Archive
Research Journal Archive, INDIA

(Review Article)

Check for updates

Multi-state safety analysis of civil Aero-engine: From functional block diagram to bayesian network

Albertiny Zenda Tavares Monteiro *, Zhong Lu, and Wellington Mandibaya

College of Civil Aviation, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China.

International Journal of Science and Research Archive, 2025, 16(03), 539–550

Publication history: Received on 27 July 2025; revised on 01 September 2025; accepted on 04 September 2025

Article DOI: https://doi.org/10.30574/ijsra.2025.16.3.2542

Abstract

This paper presents a multi-state safety-analysis framework for civil aero-engines to support compliance with CCAR §33.75 (Safety Analysis). Functional block diagrams are mapped to Bayesian networks, enabling component and function behaviors to be modeled beyond binary states and capturing causal dependencies among failure modes and hazardous engine effects (like loss of thrust control, inability to shut down, and overspeed, etc.). Occurrence probabilities are computed via junction-tree propagation, allowing forward prediction and evidence-based diagnosis under varied operating conditions. Two case studies evaluate the approach against traditional methods (fault-tree analysis and failure modes and effects analysis), demonstrating improved expressiveness for multi-state dependencies, transparent probability updates, and consistent quantitative estimates for §33.75 hazard categories. The results indicate that the proposed framework provides a rigorous and traceable basis for quantitative safety substantiation and can complement existing certification workflows in civil aviation airworthiness.

Keywords: Airworthiness Certification; Bayesian Network; Junction Tree Algorithm; Multi-State Safety Analysis

1. Introduction

Safety analysis is an important means of enhancing the safety of aircraft systems and is also the primary method for conducting compliance verification of civil aircraft systems during the certification process [1,2]. Safety refers to a system's ability to avoid accidents and serves as a measure of whether the system's risk is within an acceptable range [3]. The safety of civil aircraft is typically described using the probability of top-level failure states. The airworthiness certification authority mandates quantitative analysis for three categories of unsafe conditions, "catastrophic," "hazardous," and "major," and requires that their occurrence probabilities be controlled within specified limits [4–7]. Since a failure in the aero-engine can have catastrophic consequences for the aircraft, personnel, and property, it is essential to ensure its safety.

Traditional safety methods such as Fault Tree Analysis (FTA) and Failure Modes and Effects Analysis (FMEA) are well-established, but they struggle with multi-state, multi-consequence behavior in complex aero-engine systems. In particular, Fault Trees (FTs) (and Bayesian networks generated from them) typically analyze one top event at a time, becoming unwieldy when modeling component's own mechanical failures, and assuming binary node states limitations that hinder representing components with multiple degradation levels or capturing cross-coupled hazards. Additionally, FT-based importance measures consider only a single failure state per component, missing how different failure modes shift system risk.

To address these gaps, this paper frames a multi-state Bayesian network (BN) approach that maps multifunctional block diagrams (FBD) directly into the BN structure and performs exact inference via the junction-tree algorithm for both

[*] Corresponding author: Albertiny Zenda Tavares Monteiro.

forward inference and backward inference. The framework supports consolidated or one-to-one treatment of redundant components and naturally extends to multi-state nodes. We demonstrate the method on four engine consequences—significant thrust in the opposite direction to the direction commanded by the pilot, complete inability to shut down, loss of thrust control, and overspeed—and show that BN results closely match FT and fault-tree-based BN baselines, validating the approach while enabling simultaneous multi-consequence analysis. Importance analysis further highlights high-leverage components, including the overspeed protection device, bypass/ΔP valves, speed-control pilot valve, and FMV, as key drivers of risk.

The remainder of the paper reviews related work, formalizes the MFBD→BN mapping and inference procedure, and details the engine case setup. We then report posterior hazard probabilities, diagnostic rankings, and importance metrics, followed by a quantitative and qualitative comparison against FT/FBN, and concluding remarks.

2. Related work

The evolution of engineering systems toward greater complexity has underscored the limitations of traditional binary-state safety analysis, which often fails to capture the full spectrum of system performance degradation [8]. Consequently, multi-state system (MSS) reliability and safety analysis has emerged as a critical field of study, acknowledging that systems can exist in various operational states between perfect functioning and complete failure [9]. In this context, Bayesian networks have gained significant traction as a robust framework for modeling and analyzing complex systems under uncertainty [10]. The graphical nature of BNs facilitates the representation of causal relationships and dependencies among system components, while their probabilistic foundation allows for both predictive and diagnostic reasoning, making them particularly well-suited for comprehensive safety assessments [11,12].

The construction of a BN model is a crucial step in the analysis, and various approaches have been proposed to derive BNs from existing system representations. A common practice involves the conversion of traditional safety analysis models, such as Fault Trees and Event Trees (ETs), into BNs to leverage the advanced analytical capabilities of the latter [13,14]. More recent research has focused on the direct generation of BNs from system architectural descriptions, such as functional block diagrams, to create a more intuitive and direct mapping between the system's functional structure and the probabilistic model [15]. This approach allows for a more detailed and nuanced representation of the system's behavior, capturing the intricate interactions between components and their various failure modes. The use of causal Bayesian networks (CBNs) further enhances this by providing a deeper understanding of fault propagation pathways and enabling more effective safety interventions [16].

Given the often large and complex nature of the BNs resulting from real-world systems, the computational cost of probabilistic inference can be a significant challenge. To address this, efficient inference algorithms are essential. The junction tree algorithm is a widely adopted method for exact probabilistic inference in BNs, which works by transforming the BN into a tree structure of cliques, upon which a message-passing scheme is executed to compute marginal probabilities [17]. This algorithm has been successfully applied in various domains for reliability and safety analysis, demonstrating its efficiency and scalability [18]. By integrating the junction tree algorithm, the safety analysis of complex, multi-state systems can be performed in a computationally tractable manner, providing a powerful tool for identifying system vulnerabilities and informing risk management decisions. This paper builds upon these foundations by presenting a novel methodology for multi-state safety analysis that leverages the automated construction of a BN from an FBD and employs the junction tree algorithm for efficient and accurate probabilistic inference.

3. Multi-state BN construction based on FBD

This section introduces a process for creating a multi-state engine safety model. The technique involves building a Bayesian network by sequentially adding nodes derived directly from a functional block diagram, resulting in a model that considers various hazardous consequences.

3.1. System functional block diagram construction

The first step in conducting probabilistic safety analysis is to define the top-level failure states and, based on the functions related to these failure states, develop a functional block diagram. A functional block diagram represents the relationships among different components of the system as well as the functional logic sequence, information flow, and interfaces across the system. It serves as a functional model and forms the foundation for the Bayesian Network modeling in this work.

This section of this work proposes a method to construct a multifunctional block diagram by analyzing the specific components of the system involved in implementing the related functions. The detailed process is as follows:

1. Identification of relevant systems that implement the function based on the functions associated with the failure state.

2. Classification of system components into three categories according to their specific functions: a. Sensing and Detection Components: These mainly include various types of sensors that detect relevant parameters of operating conditions and transmit the information to the information processing and control components. b. Information Processing and Control Components: These primarily consist of electronic computers and their interfaces, which process the received information and perform control functions. c. Actuating Components: These mainly include various types of actuators that carry out the specific actions commanded by the information processing and control components.

3. Identification of the actuating components that are required to accomplish the function and determine whether the completion of a specific function requires one or multiple actuating components. If only a single actuating component is required for the completion of a certain function, analyze the factors leading to the failure of the function, including: the actuating component responsible for function failure; the information processing and control component that may cause the actuating component to fail; and the sensing and detection component that may cause the information and control component to fail. If multiple actuating components are required, proceed to the analysis step 4.

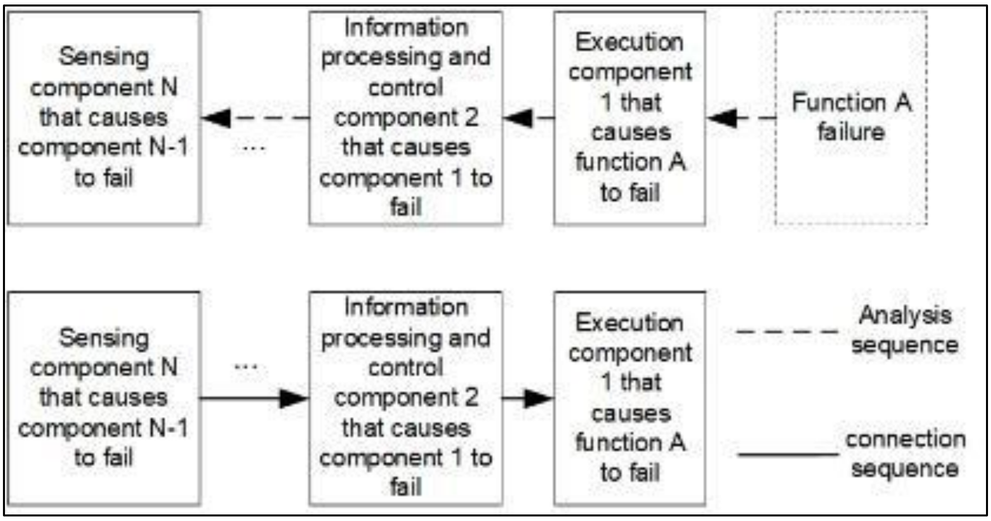
Sensing component N that causes component N-1 to fail
...
Information processing and control component 2 that causes component 1 to fail
Execution component 1 that causes function A to fail
Function A failure
Sensing component N that causes component N-1 to fail
...
Information processing and control component 2 that causes component 1 to fail
Execution component 1 that causes function A to fail
Analysis sequence
connection sequence

Figure 1 Functional block diagram analysis and connection sequence for a single actuating component

4. If two or more actuating components are required, first analyze each actuating component according to step 3. Then, examine whether there are interconnections between components across the two levels. If the components are the same, they should be merged, and if there are interactions between components, establish node connections accordingly. As done before, connect the components in the reverse order of the analysis sequence, as shown in Fig.1.

5. Repeat steps 3 and 4 until all functions have been fully analyzed.

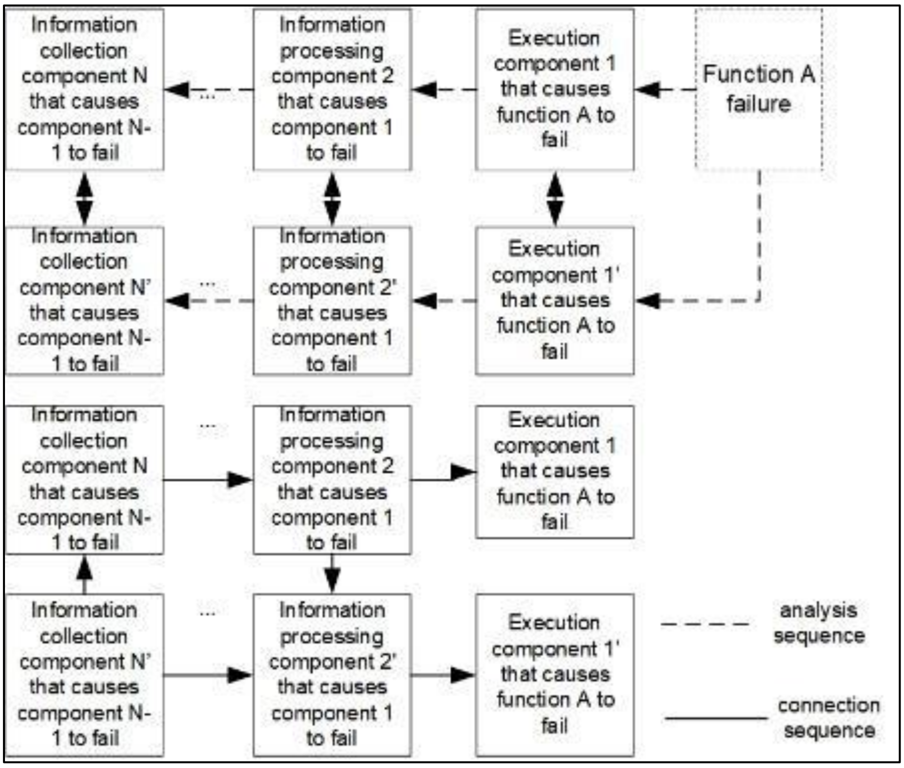
Information collection component N that causes component N-1 to fail
Information processing component 2 that causes component 1 to fail
Execution component 1 that causes function A to fail
Function A failure
Information collection component N' that causes component N-1 to fail
Information processing component 2' that causes component 1 to fail
Execution component 1' that causes function A to fail
...
analysis sequence
connection sequence

Figure 2 Functional block diagram analysis and connection sequence for two actuating components

To simplify the functional block diagram, it is agreed that

- If a function is a sub-function of another function, the sub-function's block diagram does not need to be shown separately.
- Similar redundant components can be represented using a single block diagram.
- Other influencing factors related to functional failure are represented using dashed boxes.

3.2. Mapping principles between the system functional block diagram and the BN

When mapping a system functional block diagram into Bayesian network nodes, the following principles apply

- The established BN has two levels: The first level is the Component Layer Nodes - these nodes represent the components involved in the functional block diagram. Components enclosed in dashed boxes are referred to as external nodes. The second level is the Event Nodes - these nodes represent significant or hazardous engine consequence events, as well as intermediate nodes added for modeling convenience.

- In the process of handling nodes, there are two principles for dealing with multiple similar redundant components: 1. One-to-One Mapping: Each component corresponds to a single BN node, meaning that *n* similar redundant components are represented by *n* BN nodes. 2. Merged Mapping: Multiple similar redundant components are represented by a single BN node, meaning that *n* similar redundant components are mapped to one BN node.

- For Merged Mapping, it is necessary to modify the failure states of the node by increasing the number of failure states according to the number of redundant components. That is, for a redundant component A, the corresponding BN node should include failure states such as: one component failed, two components failed, up to *n* components failed.

3.3. Bayesian network node order based on a multifunctional block diagram

3.3.1. Node classification

The nodes are classified as follows

- For first-layer component nodes, divide them into three types based on their input and output characteristics.
- Nodes with no input and with output: These are typically root nodes, often representing external inputs or independent components.
- Nodes with input and with output: These represent components influenced by others and also affecting subsequent nodes.
- Nodes with input and no output: These are usually terminal components whose state does not further affect other components directly.

For second-layer event nodes, classify them according to their actual meaning into

- Intermediate Event Nodes: Represent transitional or supporting events used for structural clarity in the BN.
- Engine Consequence Nodes: Represent critical top-level outcomes such as loss of function or system failure.

3.3.2. Node number

- C_{ij}^F: Component node, where i represents the system number and j the component number within that system.
- E_j^F: External node, representing external influencing factors.
- M_j^F: Intermediate event node, introduced for modeling clarity or logical connection.
- R_j^F: Engine consequence node, representing critical failure or hazardous outcomes.
- P_k: Common node, shared among multiple subsystems or functions
- α: Node ordering sequence in the Bayesian network.
- N: Structure of the Bayesian network.

In the above notation:

- *F* denotes the function being analyzed. To conveniently number the analyzed functions, let *F* = 1,2, ..., *n*.
- *i* represents the type of component node based on its input-output properties:
- *i* = 0: components node with no input and with output
- *i* = 1: component node with input and output
- *i* = 2: component node with input and no output.
- *j* is the component serial number, where *j* = 1, 2, ..., *n.*
- *k* is the number of common nodes, where *k* = 1, 2, ..., *n*. generally *k* < *j*.

3.3.3. Node order

This section requires determining the order in which nodes are added and constructing a complete Bayesian network structure based on the multifunctional block diagram. The steps are as follows

- Determine the Bayesian network $N_1$for the first functional failure, starting from an empty graph, add nodes to N_1 one by one in the order α_1. The node addition order α_1 is

$$\alpha_1 = \left\langle C_{0j}^1, C_{1j}^1, C_{2j}^1, E_j^F, M_j^1, R_j^1 \right\rangle \qquad j = 1, 2, \ldots n \quad (1)$$

- Determine the Bayesian network N_2 for the second functional failure

Before proceeding to the second step, it is necessary to first identify the corresponding common nodes P_k between function block diagram 1 and function block diagram 2, which are generally component nodes. Based on N_1, add nodes to N_2 one by one in the order α_2. The node addition order α_2 is

$$\alpha_2 = \left\langle P_k, C_{1j}^2, C_{0j}^2, C_{2j}^1, E_j^F, M_j^2, R_j^2 \right\rangle \qquad j = 1, 2, \ldots, n \qquad (2)$$

- Determine whether there are any other functional failures

If there are still function block diagrams corresponding to other functional failures, repeat Step 2 until all function block diagrams corresponding to functional failures have been considered.

- Directed line connection

Connect each node according to the functional relationship between each component in the working principle, and based on the analysis of functional failure events, direct the component nodes collectively toward intermediate event nodes or functional failure nodes.

3.4. Probability calculation of a Bayesian Network based on the junction tree algorithm

A junction tree is an undirected graph, while a Bayesian network is a directed graph. The fundamental idea of the algorithm is to convert the directed graph into a tree and then perform computations through message passing. The first step of the junction tree algorithm is to convert the Bayesian network into a junction tree structure. This involves the following three stages:

3.4.1. Construct the moral graph

Take a Bayesian network N, for each node X in N, identify its parent nodes P_a (X). Connect all nodes in P_a (X) pairwise with undirected edges to construct the moral graph M of N. As shown in Fig. 3.

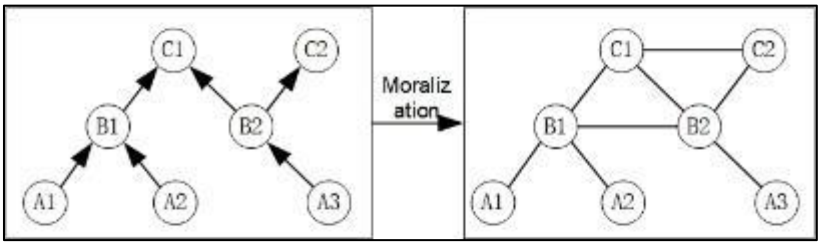

Figure 3 Moral graph construction

3.4.2. Triangulate the moral graph.

For the moral graph M obtained in the previous step, connect any two non-adjacent vertices within a cycle using undirected edges to obtain the triangulated moral graph M′. As shown in Fig. 4.

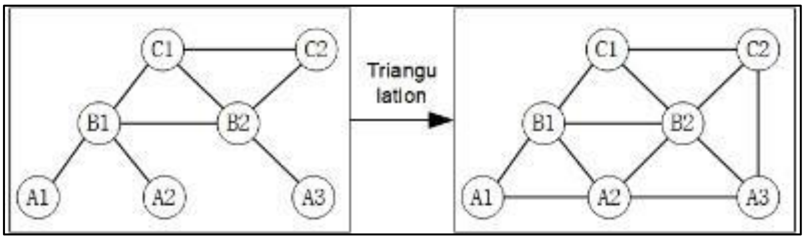

Figure 4 Triangulation of the Moral graph

3.4.3. Generate the Junction tree.

When generating the Junction tree, start with the triangulated moral graph M′ obtained in the process before, and eliminate variables following a certain order. When eliminating a variable X, all nodes adjacent to X form a junction. After elimination, the resulting junctions are connected to form the Junction tree T. For the triangulated moral graph M′, using the maximum cardinality search (MCS) algorithm, an elimination order of {A1, A2, A3, B1, B2, C2, C1} is determined.

The second step of the junction tree algorithm is initialization. After converting the Bayesian network into a junction tree in the first step, the conditional probability tables from the Bayesian network must be transferred into the junction tree, and parameters must be assigned to all nodes in the junction tree.

The third step of the junction tree algorithm is message passing, which is divided into two stages: the evidence collection stage and the evidence distribution stage. As shown in Figure 5, φi represents the evidence collection stage, and ψi represents the evidence distribution stage.

The fourth step of the junction tree algorithm is belief computation and evidence updating.

$$
\begin{aligned}
&\phi_5(A2,B1)=\sum_{A1}P(B1\,|\,A1,A2)P(A1)P(A2)\\
&\phi_4(B1,B2)=\sum_{A2,A3}P(B1\,|\,A2,A3)P(B2\,|\,A2,A3)P(A2)P(A3)\\
&\phi_3(C2)=\sum_{A3,B2}P(B2\,|\,A3)P(C2\,|\,B2)P(A3)\\
&\phi_2(C1,B1)=\sum_{B2}P(C1\,|\,B1,B2)P(B1)P(B2)\\
&\phi_1(C1,C2)=P(C1)P(C2)
\end{aligned}
\tag{3}
$$

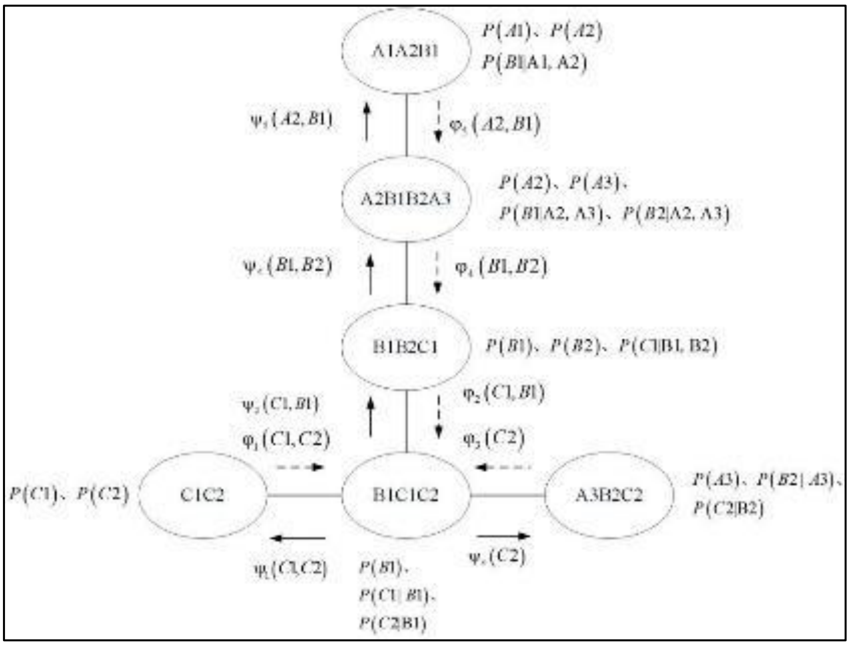

Figure 5 Message passing diagram

During the evidence distribution phase,

$$
\begin{aligned}
&\psi_1(C1,C2)=\sum_{B1}P(B1)P(C1\,|\,B1)P(C2\,|\,B1)\phi_2(C1,B1)\phi_3(C2)\\
&\psi_2(C1,B1)=\sum_{C2}P(B1)P(C1\,|\,B1)P(C2\,|\,B1)\phi_1(C1,C2)\phi_3(C2)\\
&\psi_3(C2)=\sum_{B1,C1}P(B1)P(C1\,|\,B1)P(C2\,|\,B1)\phi_1(C1,C2)\phi_2(C1.B1)\\
&\psi_4(B1,B2)=\sum_{C1}P(C1\,|\,B1,B2)P(B1)P(B2)\phi_4(B1,B2)\\
&\psi_5(A2,B1)=\sum_{C1}P(C1\,|\,B1,B2)P(B1)P(B2)\phi_5(A2,B1)
\end{aligned}
\tag{4}
$$

Given the evidence P(A1) = e, when calculating the probability of C2, select B1C1C2 as the pivot node:

$$
P(C2\,|\,A1=e)=\frac{\sum_{B1,C1}P(C1\,|\,B1)P(C2\,|\,B1)P(B1)\phi_1(C1,C2)\phi_2(C1,B1)\phi_3(C2)}{\sum_{B1,C1,C2}P(C1\,|\,B1)P(C2\,|\,B1)P(B1)\phi_1(C1,C2)\phi_2(C1,B1)\phi_3(C2)}
\tag{5}
$$

3.5. Importance analysis

Importance refers to the degree of contribution a component makes to the top event. Different importance analysis methods are used for different targets.

Fault-tree–based importance analysis evaluates only the effect of a single component in a single failure state on the system and cannot capture how multiple failure states of the same component influence overall reliability. To overcome this limitation, we compute importance measures via junction-tree inference in Bayesian networks (BN), quantifying each component's contribution to the top event. The importance measures are defined as follows:

3.5.1. BN Probability Importance

Probability importance quantifies how a change in a component's unreliability translates into a change in the system's unreliability. In a Bayesian network, it is evaluated as the change in the top-event (system failure) probability when the component is clamped to the failed state versus the normal state. For component i, the BN probability importance is

$$I_i^p = \left| p\left(s=1|i=1\right) - p\left(s=1|i=0\right) \right| \qquad (6)$$

Where: I_i^p represents the BN probability importance, $P(s = 1 \mid i = 1)$ is the system failure probability when component i is in a failed state, $P(s = 1 \mid i = 0)$ is the system failure probability when component *i* is in a normal state.

3.5.2. BN Criticality Importance

Criticality importance captures the normalized sensitivity of the top-event probability to a component's failure probability. In a Bayesian network, it is the proportional change in system-failure probability produced by a proportional change in the failure probability of component *i*.

$$I_i^c = I_i^p \frac{p(i=1)}{p(s=1)} = \frac{\left| p(s=1\,|\,i=1) - p(s=1\,|\,i=0) \right| p(i=1)}{p(s=1)} \qquad (7)$$

Where: $\mathrm{I}_\mathrm{i}^\mathrm{c}$ represents the BN criticality importance, *P* (*i* = 1) is the failure probability of component *P* (*s* = 1) is the system failure probability.

The foregoing derivation assumes binary nodes. For a multi-state node with states {0, 1,...,*n*} (where 0 denotes the normal state), we treat it as a set of binary contrasts: normal ($i = 0$) versus each specific state *j* (*i* = *j*, *j*=1..., *n*). The importance measure is then computed separately for each j.

4. Multi-state engine safety analysis

4.1. Case study

This section applies a Bayesian network–based multi-state safety analysis to civil aero-engines. Four illustrative cases are examined: two hazardous engine effects specified in the Airworthiness Standards for Aircraft Engines (CCAR-33R2 §33.75)—(i) Significant thrust in the opposite direction to that commanded by the pilot

and (ii) complete inability to shut the engine down—along with one major engine effect, loss of thrust control, and the overspeed event.

According to the description in AC33.75-1A, when both the HPSOV and LPSOV fail to close, the hazardous engine consequence "complete inability to shut down the engine" occurs. The main components of the engine fuel system and its functional block diagram are shown in Figure 6.

LOTC is a major engine effect event that is required according to the CCAR 33.28. It is treated as a failure mode of the thrust reverser function. The functional diagram is identical to Figure 6. In practice, LOTC may be triggered by failure of the fuel-metering valve (FMV), malfunction of the FMV torque motor, or erroneous thrust-control signaling from the engine control unit (ECU).

Overspeed increases mechanical loads and can lead to uncontained high-energy debris (UHED). It may also weaken the mounting lug's structural integrity, risking engine-mounting system failure and unintended engine separation. The overspeed condition can arise if excessive fuel passes through the FMV while overspeed protection is lost. Therefore, the functional block diagram for "overspeed" is also the same as in Figure 6.

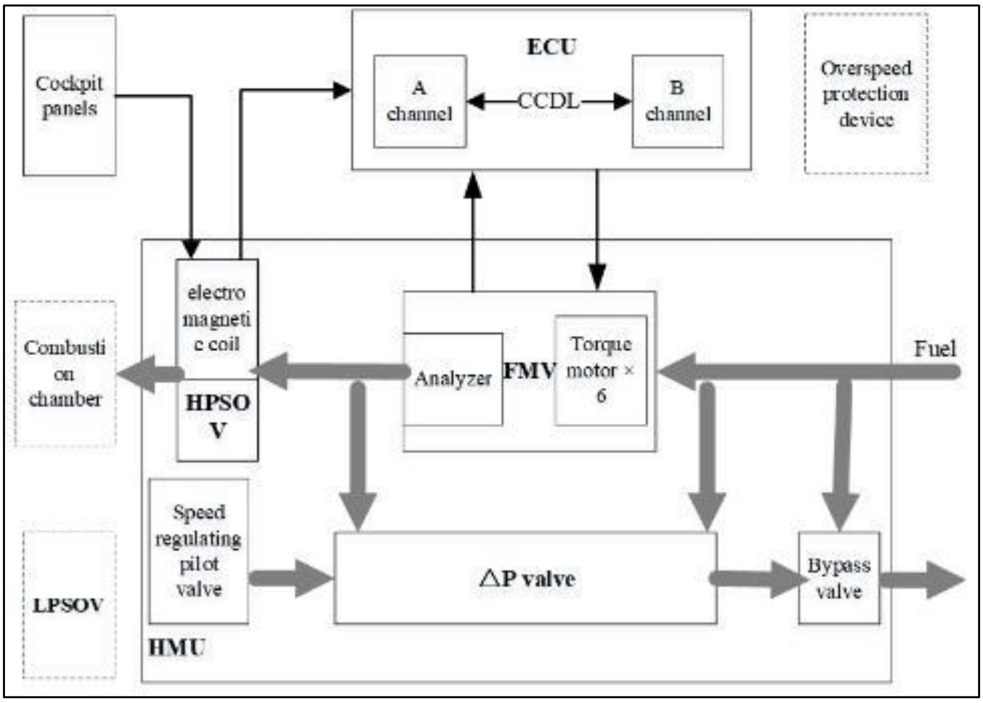

Figure 6 Schematic diagram of engine fuel shutoff valve

The mapping process and node enumeration process were conducted as specified in Section III, where the component nodes were represented by C_{ij}^{F}, external nodes by E_j^F, intermediate event nodes by M_j^F, common nodes by P_k, and engine consequence nodes by R_j^F. During the mapping process, F=1 was defined for the thrust reverser function and F = 2 for the engine fuel shutoff function.

While constructing the BN for the first function, the nodes were added sequentially according to α_1, and connected based on the relationship among the components in the functional block diagram. The second function was added according to α_2, and similarly all the nodes were connected according to the interaction relationships in the functional block diagram.

Table 1 Failure probabilities of root nodes

Part	**State**	**Prior probability (10^{-6}/h)**
Cockpit panels	Output fault	20.6
Speed regulating	Output fault	5.9
TLA Sensor	Wrong output/ insufficient output	2.4/21.6
LGCIU	Wrong output/ insufficient output	2.4/21.6
ADIRU	Output error	2.4
LPSOV	Cannot be closed	4.2
Overspeed device protection	Fault	21.6

In many cases, the upper-level component is functioning normally, but the lower-level component may still fail. The corresponding component failure rates for this scenario are summarized in Table II.

Table 2 Failure probabilities of non-root nodes

Part	**Failure rate (10^{-6}/h)**
Valve type (hydraulic shut-off valve/pressure valve/directional valve/FMV/HPSOV)	4.2
Valves (△ P valve/bypass valve)	5.9
Computer (SEC/EIU/ECU/FMV analyzer)	0.3
Relays (static relays/inhibit relays)	1.3
Torque type (actuator/FMV torque motor)	24.1
Sensors (deployed position sensor/stowed position sensor)	2.4
Solenoid switch	1.2
Master lock	13.4
HPSOV Solenoid Coil	36.1

Assuming a flight duration of 1 hour and substituting the above failure rates, simulations were run using a Bayesian network, and forward and backward inference analyses were performed. The occurrence probability of the hazardous engine consequences is presented in Table III.

Table 3 Simulation results

Hazardous engine consequences	**Occurrence probability**
A greater thrust in the opposite direction to the thrust commanded by the pilot	6.9211×10^{-11}
Complete loss of engine-stopping capability	1.7654×10^{-11}
Loss of thrust control	5.8298037×10^{-5}
Overspeed	$4.7303628 \times 10^{-10}$

The BN posterior rates obtained for the two hazardous engine effects 6.9211x10^{-11} per engine flight hour for "A greater thrust in the opposite direction to the thrust commanded by the pilot" and 1.7654x10^{-11} for "Complete loss of engine stopping capability" are consistent with airworthiness objectives for hazardous effects.

4.2. Comparison with traditional approaches

Assuming a 1-hour mission, three analysis routes were applied: (i) a conventional fault tree (FTA), (ii) a Bayesian network translated from the FTA, and (iii) a BN constructed directly from the functional block diagram. The FTA and its derived BN appear in the appendix, and the posterior probabilities for the hazardous engine effects are summarized in Table III. The FBD-based BN produces estimates that closely track those from the FTA and the FTA-derived BN, supporting the effectiveness of the proposed modeling approach.

By comparing these 3 models, the following observations were made

- The FTA and the BN translated from it evaluate a single top event at a time and do not naturally capture cases where one component failure can trigger multiple top events (system polymorphism).
- When a component's intrinsic mechanical failure must be represented, the FTA requires introducing multiple basic events, which inflates model complexity.
- The Fault tree is unable to express the multi-state component behavior.

Table 4 Probability Comparison with Traditional Approaches

Hazardous engine consequences	Based on a multifunctional block diagram	FBN	FT
A greater thrust in the opposite direction to the thrust commanded by the pilot	$6.9211237 \times 10^{-11}$	$6.9135827 \times 10^{-11}$	$6.9146601 \times 10^{-11}$
Complete loss of engine-stopping capability	$1.7653656 \times 10^{-11}$	$1.7643393 \times 10^{-11}$	$1.7651668 \times 10^{-11}$
Loss of thrust control	5.8298037×10^{-5}	4.6599272×10^{-5}	4.8999154×10^{-5}
Overspeed	$4.7303628 \times 10^{-10}$	$4.7303614 \times 10^{-10}$	$5.3783469 \times 10^{-10}$

5. Conclusion

This study proposed a multi-state safety analysis framework for civil aero-engines by directly constructing a Bayesian Network from a functional block diagram. Unlike conventional fault-tree or failure modes analyses that assume binary states, the framework captures multi-state component degradation and cross-coupled hazardous engine consequences. The case study, a thrust in the opposite direction to the direction commanded by the pilot, inability to shut down, loss of thrust control, and overspeed, demonstrated that the FBD-based BN provides quantitative estimates consistent with established fault-tree baselines while offering greater expressiveness.

A key strength of the approach lies in its ability to analyze multiple hazardous outcomes simultaneously, revealing how a single component failure can propagate through different causal chains to produce diverse consequences. This polymorphic perspective enhances diagnostic accuracy, supports component importance analysis across multiple failure states, and provides a more rigorous foundation for compliance with CCAR §33.75 safety requirements. The results indicate that the framework not only complements traditional certification workflows but also provides a more transparent and systematic means for probabilistic safety verification.

Future work should focus on extending the framework toward larger, more integrated propulsion system models that incorporate environmental factors, maintenance-induced failures, and real-world operational data. Incorporating dynamic Bayesian networks or hybrid models may further enable time-dependent reliability assessments and prognostics. Additionally, integrating machine learning methods for automated parameter estimation from flight data and validating the approach against historical incident databases could strengthen its applicability. These directions would advance the proposed methodology into a robust decision-support tool for both certification authorities and engine manufacturers.

Compliance with ethical standards

Disclosure of conflict of interest

No conflict of interest to be disclosed.

References

[1] Filippo De Florio. Airworthiness: An Introduction to Aircraft Certification, A Guide to Understanding JAA, EASA and FAA Standards [M]. Oxford: Butterworth-Heinemann Ltd, 2006.

[2] Ding Shui-ting, Zhang Gong, Yu Duo-kui et al. Review of probabilistic risk assessment on aero-engine airworthiness [J]. Journal of Aerospace Power, 2011, 26(7): 1441-1451.

[3] SAE International Group. ARP 4754a, Development of Civil Aircraft and Systems [S]. Warrendale, Pennsylvania: Society of Automotive Engineers, 2011.

[4] SAE International Group. ARP4761, Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne System and Equipment [S]. Warrendale, Pennsylvania: Society of Automotive Engineers, 1996.

[5] European Aviation Safety Agency. CS-25/Amendment 24: Certification Specifications for Large Aeroplanes [S]. Koln: European Aviation Safety Agency, 2020.

[6] Federal Aviation Administration. Advisory Circular 25.1309-1B: System Design and Analysis [M]. Washington, DC: FAA, 2024.

[7] Duane K. Aircraft System Safety Assessments for Initial Airworthiness Certification [M]. Duxford UK: Elsevier Ltd, 2016.

[8] Mokhtarabadi H, Faghih-Imani A. Reliability analysis of multi-state system with common cause failure based on Bayesian networks. Eksploatacja i Niezawodnosc - Maintenance and Reliability. 2013;15(2):169-175.

[9] Li W, Fang B. Safety analysis method of mixed failure model using temporal Bayesian network. Journal of Internet Technology. 2022;23(4):727-734.

[10] Fenton N, Neil M. Decision support software for probabilistic risk assessment using Bayesian networks. IEEE Proceedings - Software. 2014;161(3):124-131.

[11] Kabir S, Walker M, Papadopoulos Y. A review of applications of Bayesian networks in safety and reliability. Journal of Risk and Reliability. 2018;232(4):437-456.

[12] Weber P, Medina-Oliva G, Simon C, Iung B. A multi-state Bayesian network for C-BPI based on system structure. Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability. 2012;226(4):430-441.

[13] Bobbio A, Portinale L, Minichino M, Ciancamerla E. A comparative analysis of fault tree and Bayesian network approaches to dependability analysis. In: Proceedings of the 17th International Conference on Computer Safety, Reliability, and Security. Springer-Verlag; 2001. p. 245-256.

[14] Khakzad N, Khan F, Amyotte P. Safety analysis in process facilities: comparison of fault tree and Bayesian network approaches. Reliability Engineering and System Safety. 2011;96(8):925-932.

[15] Gansch R, Putze L, Koopmann T, Reich J, Neurohr C. Causal Bayesian networks for data-driven safety analysis of complex systems. arXiv preprint arXiv:2505.19860. 2025.

[16] Mahadevan S, Zhang R, Smith T. Bayesian networks for system reliability analysis. In: Proceedings of the 9th International Conference on Structural Safety and Reliability. Millpress; 2001. p. 1-8.

[17] Lauritzen SL, Spiegelhalter DJ. Local computations with probabilities on graphical structures and their application to expert systems. Journal of the Royal Statistical Society: Series B (Methodological). 1988;50(2):157-224.

[18] Byun J, Song J. A general framework of Bayesian network for system reliability analysis using junction tree. Reliability Engineering and System Safety. 2021;215:107844.