Application of Common Cause Failure Methodology to Aviation Safety Assessment Model

Seungwon Noh
Systems Engineering and Operations Research
George Mason University
Fairfax, VA, USA
snoh2@gmu.edu

John Shortle
Systems Engineering and Operations Research
George Mason University
Fairfax, VA, USA
jshortle@gmu.edu

***Abstract*—The Federal Aviation Administration (FAA) has been developing the Integrated Safety Assessment Model (ISAM) to provide a baseline risk assessment for the National Airspace System and to evaluate the safety impact of proposed changes to the system. ISAM consists of a set of event sequence diagrams and underlying fault trees for various accident scenarios. In the current model, all basic events in the fault trees are assumed to be independent. However many basic events throughout the model appear with the same descriptive label. Such events might have some dependence, rather than being completely independent as is currently assumed. This paper evaluates the dependency between basic events having the same label in order to see the overall impact on accident risk. A common cause failure (CCF) methodology is applied to the event sequence diagrams (ESDs) in ISAM. A modified beta-factor model is applied, and a binary decision diagram method is implemented to evaluate end-state frequencies of an ESD. Accounting for CCFs, this paper observes a wide range of changes in accident frequency relative to the current assumption of independent events. Results for different ESDs range from a decrease in accident frequency by 50% to an increase by more than a factor of 1,000.**

Keywords - risk assessment; aircraft accident; common cause failure; beta-factor model; binary decision diagram

I. INTRODUCTION

The Federal Aviation Administration (FAA) has been developing the Integrated Safety Assessment Model (ISAM) [1] to provide a baseline risk assessment for the National Airspace System (NAS) and to evaluate the safety significance of proposed changes such as new regulations, new vehicles or planned changes in NextGen. ISAM models accident and incident scenarios of the NAS through a set of event-sequence diagrams (ESDs) and supporting fault trees. ISAM has a total of 35 ESDs and 240 associated fault trees to capture all possible accident scenarios. Each ESD contains a unique initiating event, several end states and multiple intermediate pivoting events. There are more than 3,400 fault-tree basic events throughout ISAM [2]. Fig. 1 shows one example of an ESD and the underlying fault trees. An end state occurs when the initiating event occurs and the combination of pivoting events along the path to the end state also occurs. The occurrence or non-occurrence of the initiating event and each pivoting event in turn depends upon the outcome of a fault tree underneath each event. Based on this structure, the outcome of a particular end event can be modeled by an equivalent single fault tree that combines the underlying fault trees for the initiating event and pivoting events (or their negation) at the top level via 'AND' gates.

All events in ISAM are currently assumed to be independent. Reference [2], however, shows that many of the pivoting events of the ESDs and basic events of the fault trees in ISAM appear multiple times, at least in the sense that the node labels are the same. For example, there are more than 3,400 basic events in the fault trees, yet there are only 226 unique labels for these events. Several labels appear more than 150 times across all ESDs, which means that these labels appear multiple times in a single ESD. In Fig. 1, for example, the basic event of 'No warning system in place-ATC' appears at two different points in the fault trees underlying a single ESD.

Reference [2] assumes that events having the same label in

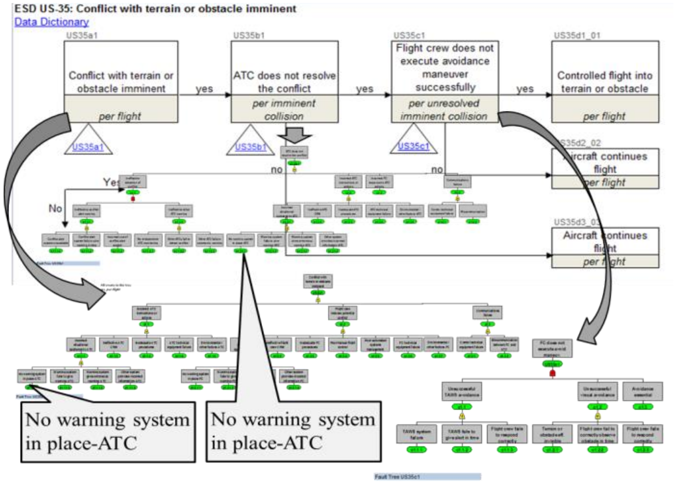
ESD US-35: Conflict with terrain or obstacle imminent
Data Dictionary
US35a1
Conflict with terrain or obstacle imminent
per flight
yes
US35b1
ATC does not resolve the conflict
per imminent collision
yes
US35c1
Flight crew does not execute avoidance maneuver successfully
per unresolved imminent collision
yes
US35d1_01
Controlled flight into terrain or obstacle
per flight
US35a1
US35b1
US35c1
no
no
US35d2_02
Aircraft continues flight
per flight
Yes
No
US35d3_03
Aircraft continues flight
per flight
No warning system in place-ATC
No warning system in place-ATC

Figure 1. Example of ESD and fault trees in ISAM

ISAM are the *same event* – in other words, if an event occurs in one part of the model, then any event elsewhere in the model with the same label must also occur. This assumption was made to conduct a first-pass sensitivity analysis on parameters in the model. However, such an assumption is not correct in the sense that each event is really a conditional probability predicated on the upstream events in the tree. For example, two pivoting events with the same label of "Flight crew does not maintain control", where one event occurs after initiating a rejected approach whereas the other event occurs without initiating a rejected approach, are different events, since they correspond to different phases of flight. On the other hand, assuming same-label events are completely independent may not be accurate either since if a basic event occurs in one situation, the other basic event having the same label might occur as well due to a similar cause.

Since existing dependency between those events having the same labels may cause significant impacts on risk quantification results, more attention should be given to events that could occur simultaneously due to a common cause. The objective of this paper is to apply a Common Cause Failure (CCF) methodology to ISAM in order to see the impacts of dependency between basic events having a same label. The analysis in this paper also varies the level of the dependency to see the sensitivity of CCFs in ISAM.

The rest of this paper is organized as follows: A literature review on CCF analysis and fault tree quantification is described in the next section. Then the analysis methodology used in this paper is illustrated with a simple example. Results are given for different types of CCF. The last section provides conclusions.

II. Literature Review

A. Common Cause Failure Analysis

A common cause failure (CCF) is defined as a set of dependent events in which two or more component fault states exist at the same time, or in a short time interval, and are the direct result of a shared cause [3]. In the 1980's, the comprehensive Probabilistic Safety Assessment (PSA) of nuclear power plants demonstrated the significance of CCFs [4]. The aviation industry has also paid attention to CCFs [5].

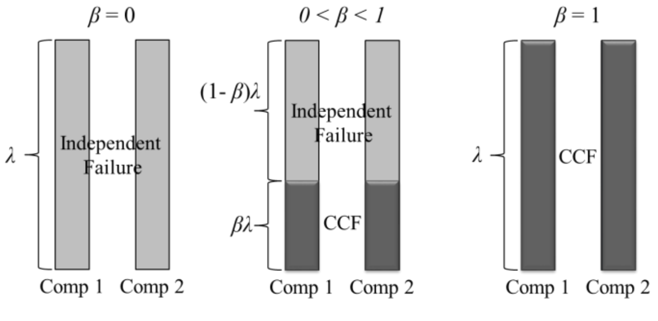

Figure 2. Beta-factor model

Several parametric models for common cause failures have been introduced. One of the most commonly used CCF models is the beta (β)-factor model, which was introduced by Fleming in 1974 [6]. The beta-factor model assumes that all components in a CCF group are identical with a constant failure rate of λ. Each component can fail in one of two ways – either as an independent failure or as part of a group failure in which all components within the group fail. (Failures of subsets of the group are not considered in the model.) The components fail independently with rate $(1 – \beta)\lambda$ and the group fails with rate $\beta\lambda$. Thus the total failure rate of each component is λ, regardless of the value of β. If $\beta = 0$, the component failures are completely independent. If $\beta = 1$, the components are perfectly correlated – they survive and fail collectively as a group. Fig. 2 illustrates the concept of the beta-factor model.

Other CCF models include the basic parameter model (BPM), the alpha (α)-factor model, the multiple-Greek-letters (MGL) model, and the binomial failure rate (BFR) model [4]. Each model has a different way to calculate the probability Q_k of a failure involving k specific components. BPM estimates the quantity of Q_k directly while the others use intermediate parameters, e.g., alpha (α) or beta (β), to calculate the probability Q_k. In this paper, the beta (β)-factor model is chosen for use (with some modification) because not only it is one of the most commonly used CCF models but it is also simple and easy to understand [6]. More detailed models have potentially more modeling power but also require more data to apply.

B. Fault Tree Quantification Method

The conventional gate operations to evaluate a fault tree are not appropriate if there are dependencies between basic events in a fault tree. A cutset-based method is one way to compute the top event probability of a fault tree. A cutset is defined as a set of basic events whose occurrence ensures that the top event occurs. A cutset is said to be minimal if the set cannot be reduced without losing its status as a cutset [6]. A cutset method begins by finding the minimal cutsets. Then, it evaluates the probability of the union of every minimal cutset C_i to obtain the system unreliability U_{sys} [7]:

$$U_{sys} = \Pr(\cup_i^n C_i). \qquad (1)$$

In order to evaluate (1) the inclusion-exclusion principle, which is the rule for computing the probability of the union of two events, is commonly applied [7]. A difficulty in applying this method to ISAM is that the inclusion-exclusion calculations involve a huge number of minimal cutsets. Similar computational challenge exists when other cutset-based methods (e.g. sum of disjoint products [7]) are applied. Nevertheless, summing up the probability of each minimal cutset can provide an upper bound on the system unreliability.

Another way to compute the top event probability of a fault tree is to use Binary decision diagrams (BDDs). The BDD method is a relatively recent method to solve a fault tree model for the system reliability analysis [8, 9]. Introduction of BDD in the reliability analysis field has improved accuracy and efficiency in fault tree analysis [10, 11]. According to [8], the BDD method changes the analyzing fault trees process significantly: 1) minimal cutsets are not necessary to evaluate a fault tree, 2) BDD provides the exact result of top-event probability, but it also has a disadvantage that the size of BDD can increase exponentially as the worst case. More detailed explanation of the BDD method is in the methodology section with a simple example.

III. Methodology

This paper presents a modified version of the beta-factor model to evaluate fault trees and ESDs in ISAM. The detailed methodology is described in this section using a simple example. Fig. 3 shows an example ESD and associated fault trees. A fault tree is located underneath each event in the ESD so that each event occurs by failure of its supporting fault tree. In the example, basic events ‘D1’ and ‘D2’ underneath pivoting event 1 are assumed to have the same label. These basic events may fail simultaneously due to a common cause or independently by different causes.

A. Beta-factor Model

The beta-factor model (Fig. 2) assumes that all elements in a group fail with the same rate. One issue in applying this model to ISAM is that common-label events do not always have the same baseline failure probability. For example, the two basic events having a label of ‘No warning system in place-ATC’ in Fig. 1 have probabilities of 2.45E-6 and 1.44E-4 respectively. In some cases, some same label events even have zero probability [2]. In order to resolve this issue, the following assumption is made: The minimum failure probability among the probabilities of events that have a common label is the maximum failure probability due to CCF. For example, if we have three components (a, b and c), and the failure probabilities of each component are 0.1, 0.2 and 0.3 respectively, then the maximum failure probability due to a CCF for these three components is 0.1, which is the minimum failure probability among the components (Fig. 4). As a special case, if some of the components in a group have zero failure probability, then the components having zero failure probability in the group fail as purely independent events.

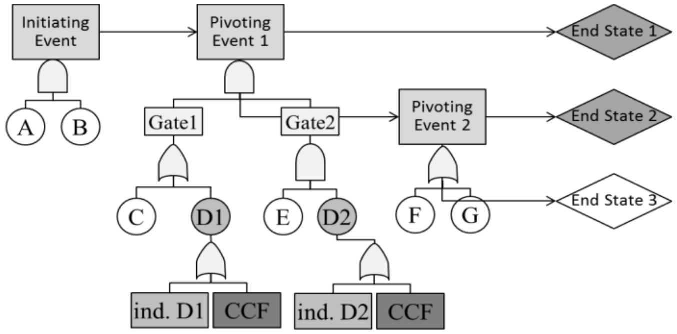

Figure 3. Example ESD and underlying fault trees

In order to apply the beta-factor model, basic events that have a same label are split into two sub-basic events, one for independent failure and the other for common cause failure, which are combined with an ‘OR’ gate. In Fig. 3, for example, basic events ‘D1’ and ‘D2’ are same-label events. Each of these is split into two sub-basic events. The probabilities of newly generated sub-basic events, ‘Ind.D1’, ‘Ind.D2’ and ‘CCF’ in Fig. 3, are calculated by solving the following system of equations

$$\begin{cases}\beta = \Pr(CCF)/[\Pr(ind.D1) + \Pr(CCF)] \\ \Pr(D1) = 1 - [1 - \Pr(ind.D1)][1 - \Pr(CCF)]\end{cases} \tag{2}$$

The first equation defines beta (β) as the ratio of a CCF failure to the total failure rate of the component. The second is the logic gate equation of ‘OR’ gate. Given β, *Pr*(*D1*) and *Pr*(*D2*), where *Pr*(*D1*) < *Pr*(*D2*), the system of equations is solved to obtain *Pr*(*CCF*) and *Pr*(*ind.D1*) are obtained as follows:

$$Pr(CCF) = \begin{cases} 0, & (\beta = 0) \\ \frac{1-\sqrt{1-4\beta(1-\beta)Pr(D1)}}{2(1-\beta)}, & (0 < \beta < 1) \\ Pr(D1), & (\beta = 1) \end{cases} \tag{3}$$

$$Pr(ind.D1) = \frac{Pr(D1)-Pr(CCF)}{1-Pr(CCF)} \tag{4}$$

Then, similar to (4), *Pr*(*ind.D2*) is calculated by solving a logic gate equation for ‘D2’.

$$Pr(ind.D2) = \frac{Pr(D2)-Pr(CCF)}{1-Pr(CCF)} \tag{5}$$

B. Binary Decision Diagram (BDD) Method

A BDD is a directed acyclic graph based on Shannon’s decomposition of a Boolean function. A BDD is composed of terminal nodes which indicate system success (value 0) or system failure (value 1) and non-terminal nodes corresponding to basic events of a fault tree. Each non-terminal node has two out-branches: One is called the 0-branch representing the non-

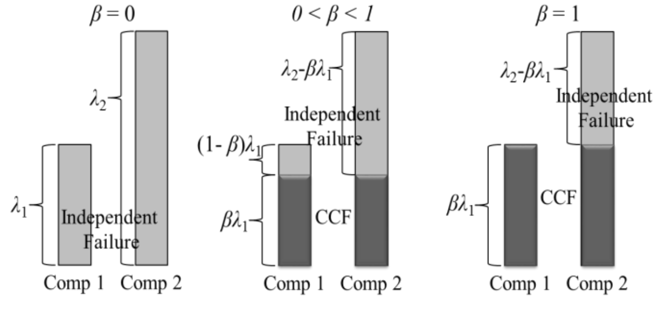

Figure 4. Modified beta-factor model

occurrence of a basic event (working state). The other is called the 1-branch representing the occurrence of the basic event (failed state). The BDD method converts a fault tree to a binary decision diagram encoding an if-then-else (**ite**) structure [10]. '**ite**(*x*, *f1*, *f2*)' means that **if** *x* is true, **then** consider function *f1*, **else** consider function *f2*, where *x* is a Boolean variable. Fig. 5 and Fig. 6 illustrate procedures to convert a fault tree to a BDD summarized in [11] using the example fault tree under 'Pivoting Event 1' in Fig. 3.

The first step is to assign each basic event in the fault tree the **ite** structure, **ite**(basic event name, 1, 0), which means that **if** the basic event occurs, **then** the system fails, **else** the system works. At the second step, gates and the top event are considered in a bottom-up manner, e.g. Gate1 = D1<+>F1. Lastly, every gate and the top event is rewritten in terms of an **ite** structure of basic events by the following operation rules.

- For event A > B let J= **ite**(*A*, *S1*, *S2*) and H= **ite**(*B*, *U1*, *U2*); then J<op>H = **ite**(*A*, *S1*<op>*H*, *S2*<op>*H*)
- If A=B, *i.e.* let *J*= **ite**(*A*, *S1*, *S2*) and *H*= **ite**(*A*, *U1*, *U2*); then *J*<op>*H* = **ite**(*A*, *S1*<op>*U1*, *S2*<op>*U2*)
- 1 <·> H = H, 0 <·> H = 0, 1 <+> H = 1, 0 <+> H = H

Examples of the operation rules applied to the example are as follows:

- D1 = F3<+>F2 = **ite**(*CCF*, *1*, *0*) <+> **ite**(*ind.D1*, *1*, *0*)
  = **ite**(*CCF*, *1*, **ite**(*ind.D1*, *1*, *0*))
- Gate2 = D2<·>F4
  = **ite**(*CCF*, *1*, **ite**(*ind.D2*, *1*, *0*)) <·> **ite**(*E*, *1*, *0*)
  = **ite**(*CCF*, **ite**(*E*, *1*, *0*), **ite**(*ind.D2*, **ite**(*E*, *1*, *0*), *0*))

The ultimate **ite** structure for the top event of the fault tree is obtained by operations for all gates which are conducted from the bottom to the top of the fault tree. The top event **ite** structure represents the BDD of the fault tree. Fig. 6 shows the BDD of the example fault tree, and the **ite** structure for the top event, which is 'Pivoting Event 1'.

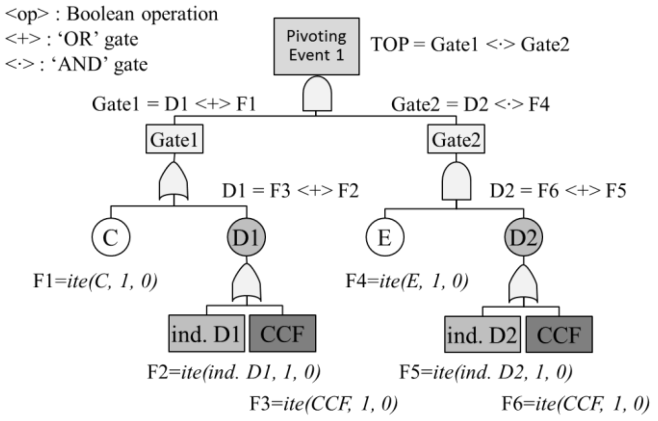

Figure 5. Conversion to BDD (1)

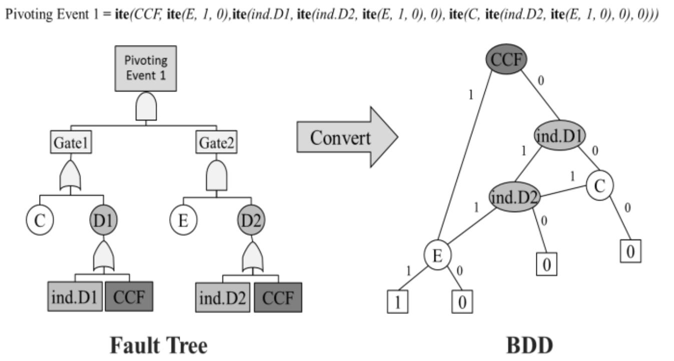
Pivoting Event 1 = **ite***(CCF,* **ite***(E, 1, 0),***ite***(ind.D1,* **ite***(ind.D2,* **ite***(E, 1, 0), 0),* **ite***(C,* **ite***(ind.D2,* **ite***(E, 1, 0), 0), 0)))*

Figure 6. Conversion to BDD (2)

In a BDD, paths from the top event to a terminal node with a "1" represent the conditions for occurrences of the top event. For example, in Fig. 6, the occurrence of the CCF and event E will cause the top event to occur. In order to evaluate the probability of the top event in a fault tree, all disjoint paths leading to a terminal node with a "1" need to be tracked, e.g., {CCF, E}, {non CCF, ind.D1, ind.D2, E}. Secondly, the probability of each disjoint path is computed by multiplication of the probabilities of the basic events failure or success in the path. For example, the probability of the path {CCF, E} is multiplication of the probability of the CCF occurrence and the probability of event E. Lastly, the probability of the top event occurrence is obtained by summing the probabilities of all disjoint paths in the BDD [12].

To analyze an ESD, i.e., to calculate the end-state probabilities in an ESD, occurrences and/or non-occurrences of the pivoting events in the path leading to an end-state need to be considered. In Fig. 3, for example, the end state 2 occurs if the initiating event occurs and pivoting event 1 does not occur and pivoting event 2 occurs. That is, {End-state 2} = {Init' event} ∩ {non-occurrence of PE1} ∩ {occurrence of PE2}. The BDD methodology provides another advantage of converting a fault tree to a success tree, which indicates the non-occurrence case of a pivoting event. This conversion needs many complicated steps in the tree structure, e.g. conversion of failure probabilities to success probabilities, and "AND" gates to "OR" gates. In the BDD form, however, it needs only one step, converting terminal nodes with a "0" to a "1" and vice-versa. The converted BDD, which represents a success tree, is called the Dual BDD (DBDD) [12].

C. *ESD analysis method*

In order to apply the BDD methodology to an event tree with multiple underlying fault trees, we use the algorithm suggested by [12]. The algorithm is summarized as follows:

- Convert each underlying fault tree in an event tree to a BDD.
- Convert BDDs to DBDDs for non-occurrence case.
- Find paths to each end-state in an event tree.

- Construct the combined BDD with BDDs and/or DBDDs in the path for each end-state.
- Evaluate the combined BDDs in ESD.

Fig. 7 shows the combined BDD for the end-state 2 in Fig. 3. Each path, which ends in a terminal-1 node from the top of the BDD, represents the combination of occurrence/non-occurrence of basic events in the ESD for end-state 2 to occur. One example path which will cause end-state 2 to occur is {occurrences of CCF, A, B, and F, and non-occurrence of E}. There are two more similar BDDs corresponding to the other two end-states.

IV. Results

In this section, we apply the beta-factor model to several ESDs in ISAM to see the effects of dependency between basic events having the same label. Three different examples are discussed – one involves a single CCF within one ESD and two involve multiple sets of CCFs with an ESD. CCFs across different ESDs are not considered.

A. *Single CCF*

The simplest application of the beta-factor model is to an ESD with two same-label events located within a single fault tree. The example ESD is US-31 which is initiated by the event that two aircraft are positioned on collision course in flight. The selected common label is ‘Flight crew fails to respond correctly’, which appears twice in the fault tree under the pivoting event of ‘US31c1’. Fig. 8 shows the example ESD and the underlying fault tree of the pivoting event US31c1 (there are also fault trees underneath the other events, which are not shown). Note that this ESD contains additional same-label events, but we are only considering a single pair for this first example. The numbers in the figure are baseline frequencies in ISAM.

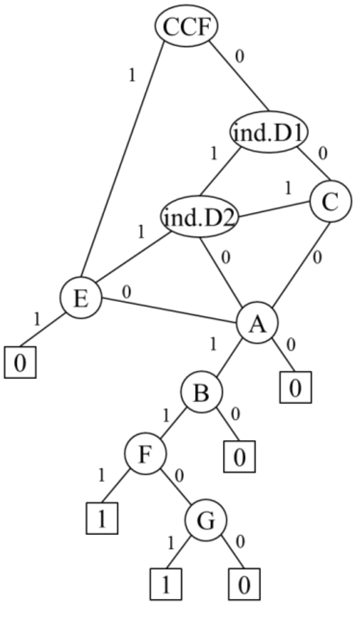

Figure 7. Example: Combined BDD for End-State

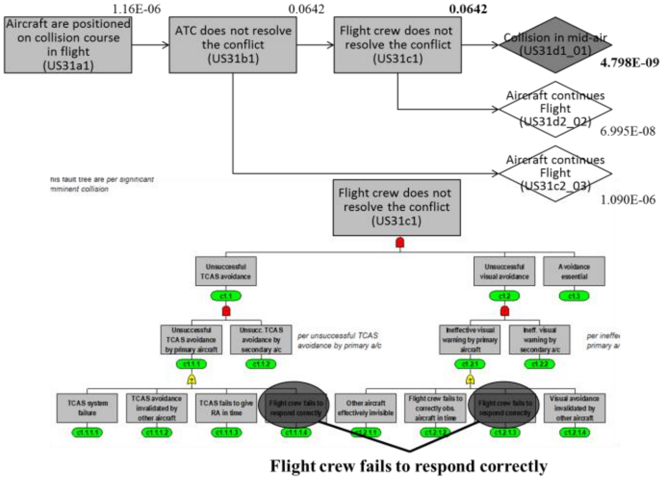
1.16E-06
0.0642
0.0642
Aircraft are positioned on collision course in flight (US31a1)
ATC does not resolve the conflict (US31b1)
Flight crew does not resolve the conflict (US31c1)
Collision in mid-air (US31d1_01)
4.798E-09
Aircraft continues Flight (US31d2_02)
6.995E-08
Aircraft continues Flight (US31c2_03)
1.090E-06
his fault tree are per significant
mminent collision
Flight crew does not resolve the conflict (US31c1)
Unsuccessful TCAS avoidance
c1.1
Unsuccessful visual avoidance
c1.2
Avoidance essential
c1.3
Unsuccessful TCAS avoidance by primary aircraft
c1.1.1
Unsucc. TCAS avoidance by secondary a/c
c1.1.2
per unsuccessful TCAS avoidance by primary a/c
Ineffective visual warning by primary aircraft
c1.2.1
Ineff. visual warning by secondary a/c
c1.2.2
per ineffe
primary a/
TCAS system failure
c1.1.1.1
TCAS avoidance invalidated by other aircraft
c1.1.1.2
TCAS fails to give RA in time
c1.1.1.3
Flight crew fails to respond correctly
c1.1.1.4
Other aircraft effectively invisible
c1.2.1.1
Flight crew fails to correctly obs. aircraft in time
c1.2.1.2
Flight crew fails to respond correctly
c1.2.1.3
Visual avoidance invalidated by other aircraft
c1.2.1.4
Flight crew fails to respond correctly

Figure 8. ESD US-31 and underlying fault tree of US31c1

Table I shows how changes in the CCF parameter β impact the probability of the pivoting event (US31c1) and the frequencies of the two end-states, ‘mid-air collision’ and ‘continue flight’. When β = 0, the component events are independent, which is the current assumption used in ISAM. When β = 1, the CCF component events are completely dependent. The total failure probability of each commonly labeled event remains the same, regardless of β. The table shows that the frequency of mid-air collision increases by almost 10% from the current frequency if complete dependency between the selected basic events (β = 1) is assumed. As expected, even a single CCF between basic events having a same label increases the pivoting probability, which causes the accident frequencies to increase.

TABLE I. Changes in Pivoting Probability and End-state Frequency with Various Betas

Beta	US31c1	US31d1_01 (mid-air collision)	US31d2_02 (continue flight)
0.00	0.0642	4.80E-09	7.00E-08
0.25	0.0655	4.90E-09	6.99E-08
0.50	0.0671	5.01E-09	6.97E-08
0.75	0.0687	5.14E-09	6.96E-08
1.00	0.0703	5.26E-09	6.95E-08

B. *Multiple CCFs (1)*

The number of basic events in an ESD varies from dozens to hundreds. Each ESD typically has multiple labels that are observed multiple times (see Appendix for a complete list). For example, in ESD US-31 (the ESD in the previous example, Fig. 8), there are 18 different labels that appear multiple times, including the same-label pair considered in the previous example. Half of the labels appear in multiple underlying fault trees, e.g. ‘ATC technical equipment failure’ appears twice in US31a1 and once in US31b1. Half of the labels appear multiple times in only a single fault tree. We now consider the impact of multiple CCFs on this ESD.

With multiple CCFs an additional assumption is made. The issue is that, theoretically, different basic event labels can have different CCF ratios, which is beta. This means that dozens more betas, e.g. β_1, β_2, and so on, are needed. It is computationally very expensive to consider all combinations of betas, and it may be challenging to interpret the results. Thus, it is assumed that all common basic event labels in an ESD have the same CCF ratio (β) at a time.

Fig. 9 shows changes in the end-state frequencies as well as the initiating event frequency when multiple CCFs are assumed with different betas. As dependency between same-label basic events increases, the frequency of mid-air collision increases significantly from $4.80\cdot10^{-9}$ to $1.02\cdot10^{-6}$. Not only the accident end-state frequency increases but also the initiating event frequency increases. This is because many of the common labels appear multiple times in the fault tree of the initiating event.

Accounting for CCFs can also cause the initiating event frequency to decrease. For example, Fig. 10 shows the results of a CCF analysis for ESD US-10, which is initiated by ‘Pitch control problem during take-off’. Unlike Fig. 9, as beta increases, the initiating event frequency decreases. The decreasing trend occurs because the fault tree underneath the initiating event consists entirely of ‘OR’ gates. In such a structure, increasing the dependence between sub-events can cause the top event probability to decrease rather than increase. As an illustrative example, consider a fault tree with two basic events, A1 and A2, combined by an ‘OR’ gate with probabilities of 0.1 each. If the basic events are independent, then the top event probability is $0.1 + 0.1 - (0.1)^2 = 0.19$. However, if the basic events are completely dependent, the top event probability is 0.1 which is smaller than 0.19 in the independent case.

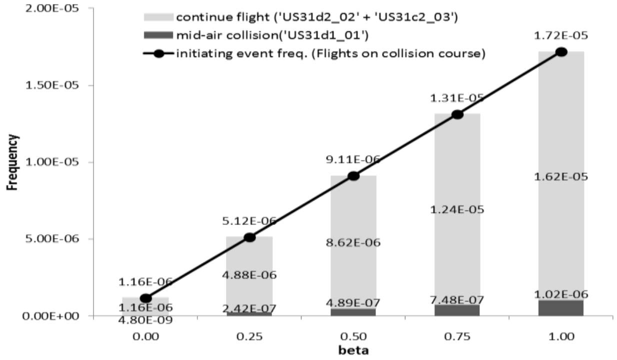

Figure 9. Changes in frequency of end-states and initiating event

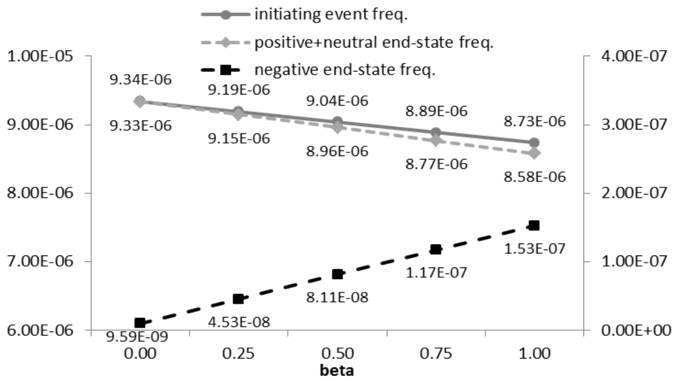

Figure 10. Results of CCF analysis for US-10

A given ESD may have multiple ‘AND’ and ‘OR’ gates, so there are competing effects. In ESD US-10, even though the initiating event frequency decreases with increasing dependence among same label effects, the final negative end-state frequency still increases, as might be expected.

One observation is that the end-state frequencies are approximately linear functions of beta in Fig. 9 and Fig. 10. Mathematically, the relationship is not exactly linear because, the CCF probability in (3) is not linear in beta. However, changes in the frequencies are very close to linear in beta since probabilities in ISAM are very small.

C. *Multiple CCFs (2)*

The second case for multiple CCFs analysis is a kind of counter example of the first case that is explained in the previous section. The selected example is ESD US-12 which is initiated by the event of ‘Flight crew member spatially disoriented’. This ESD has 9 different basic event labels appearing multiple times across all fault trees in the ESD. One interesting feature in this ESD is that any common label does not appear more than once under the initiating event, and this feature makes the CCF behavior different from the previous ESDs (US-31 and US-10). This is important because common labels appearing multiple times under the initiating event make the frequency of the initiating event change when CCF analysis is conducted.

Unlike Fig. 9 and Fig. 10, Fig. 11 clearly shows that assuming CCFs on basic events having same labels results large impacts on the accident frequency. The frequency of the accident end-state, which is ‘collision with ground’, rises rapidly from $4.79\cdot10^{-9}$ to $6.89\cdot10^{-6}$ when beta varies from 0.0 to 1.0, while the frequency of the initiating event stays the same. The fraction of initiating events that result in an accident increases from 0.024% for complete independence to 33.8% for complete dependence, a factor increase of about 1,400.

D. *Overall Results*

Among 35 ESDs in ISAM, CCF analysis has been conducted for 22 ESDs including ESDs that are presented in the previous sections. The remaining 13 ESDs do not need to

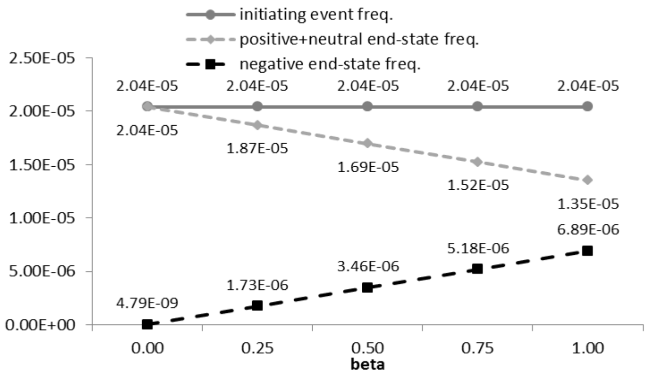

Figure 11. Results of CCF analysis for US-12

be analyzed since they have zero accident frequencies based on historical accident data, so the CCF analysis currently has no impact on the accident frequency. The computation time of the CCF analysis for each ESD varies from seconds to even several hours. It highly depends on the size of the constructed BDD for each end-state, which depends on the number of basic events, common labels and structure of fault trees.

Fig. 12 shows overall results of CCF analysis for 22 ESDs in ISAM. In order to show how relatively accident frequencies change by beta (*β*), accident frequencies are normalized by the baseline accident frequency under the assumption of independence of all basic events (β = 0). Wide ranges of changes in accident frequency are observed. The accident frequency increases by 1,400 times compared to the baseline value in ESD US-12 when β = 1, while it decreases by about 50% in ESD US-01. A decreasing accident frequency is not typically expected in a CCF analysis. However, in these cases, the initiating event and/or pivoting events have underlying fault trees that are constructed only by 'OR' gates, which causes this effect to occur.

ESDs in ISAM can be grouped into two categories. The first group is a set of ESDs in which there is at least one common label appearing multiple times in the underlying fault tree of the initiating event, and the other is a set of ESDs whose common labels appear at most once in the fault tree underneath the initiating event. As explained previously, CCF analysis will make the initiating event frequency change for the first group, whereas the frequency will remain the same in the other group. In Fig. 12, the solid lines are examples of the first group, and the rest (dashed lines) are ones for the other group. There is no certain trend in accident frequency changes by groups, but bias due to changes in initiating event frequency in ESDs of the first group (solid lines in Fig. 12) may be included.

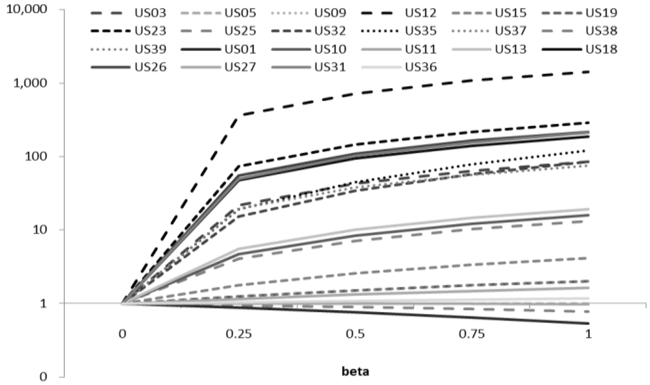

Figure 12. Results of CCF analysis (normalized value)

V. Conclusions

This paper presented a way to apply a common cause failure methodology to event sequence diagrams for aircraft accident scenarios, which are developed in ISAM. CCF analyses are conducted for basic events appearing multiple times in underlying fault trees of individual ESD to see impacts of dependency between basic events having the same labels. The modified beta-factor model is applied, and a binary-decision-diagram method is implemented to evaluate end-state frequencies of multiple ESDs.

Assuming dependency between basic events having a same label certainly impacts the frequencies of the end-states of an ESD, e.g., a single CCF on 'Flight crew fails to respond correctly' can cause the frequency of mid-air collision to increase by up to 10% (in ESD US-31). When there are multiple CCFs in an ESD, the impact can be even more significant. The accident frequency in ESD US-12 increases by about 1,400 times compared to the current accident frequency when all basic events having same labels are assumed dependent. The CCF analysis in this paper causes changes in the end-state frequencies as well as the initiating event frequency when any basic event appears multiple times in the underlying fault tree of an initiating event.

Acknowledgment

This work was sponsored by Dr. Sherry Borener, Program Manager, System Safety Management Transformation. The authors also thank Mr. Alan Durston and Mr. Brian Hjelle for providing data, code, and support to run ISAM in the context of this research.

References

[1] S. Borener, S. Trajkov, and P. Balakrishna. "Design and development of an Integrated Safety Assessment Model for NextGen," International Annual Conference of the American Society for Engineering Management, 2012.

[2] S. Noh and J. Shortle, "Sensitivity analysis of event sequence diagrams for aircraft accident scenarios," Proceedings of Digital Avionics Systems Conference, Prague, 2015, 3E2-1 - 3E2-12.

[3] A. Mosleh et al., "Procedures for treating common cause failures in safety and reliability studies, U.S. Nuclear Regulatory Commission and Electric Power Research Institute, NUREG/CR-4780, and EPRI NP-5613. Vol. 1 and 2, 1988.

[4] A. Mosleh, "Common cause failures: An analysis methodology and examples," Reliability Engineering and System Safety, vol. 34, 1991, pp. 249-292

[5] M. Stamatelatos et al., "Probabilistic risk assessment procedures guide for NASA managers and practitioners," NASA Headquarters, Washington DC, 2011

[6] M. Rausand and A. Hoyland, System reliability theory; models and statistical methods, Wiley, New York, 2004

[7] L. Xing and S. Amari, Handbook of performability engineering; Chap. 38: Fault tree analysis, Springer, 2008

[8] A. Rauzy, Handbook of performability engineering; Chap. 25: Binary decision diagrams for reliability studies, Springer, 2008

[9] A. Rauzy, "New alorithms for fault trees analysis," Reliability Engineering and System Safety, vol. 40, 1993, pp. 203-211

[10] R. Sinnamon, and J. Andrews, "Improved accuracy in quantitative fault tree analysis," Quality and Reliability Engineering International, vol. 13, 1997, pp. 285-292

[11] R. Sinnamon, and J. Andrews, "Improved efficiency in qualitative fault tree analysis," Quality and Reliability Engineering International, vol. 13, 1997, pp. 293-298

[12] J. Andrews and S. Dunnett, "Event-tree analysis using binary decision diagrams," IEEE Transactions on Reliability, vol. 49, 2000, pp. 230-238

APPENDIX

This appendix provides a sense of how many potential CCFs exist within each ESD in ISAM. The second column is the number of basic events in the fault trees of each ESD, and the third column shows the number of labels that are unique among the basic events within each ESD. The last column is the number of labels that appear multiple times within each ESD.

ESD	**# of basic events**	**# of labels**	**# of common labels**
US01	323	43	28
US02	119	49	17
US03	96	24	17
US04	120	35	27
US05	94	26	17
US06	41	25	9
US08	80	26	13
US09	89	28	17
US10	128	36	27
US11	167	42	18
US12	38	28	9
US13	119	23	19
US14	14	14	0
US15	31	26	5
US16	98	23	17
US17	70	31	21
US18	60	33	11
US19	150	51	30
US21	172	51	25
US23	147	40	30
US25	131	44	18
US26	42	17	9
US27	119	25	18
US31	79	37	18
US32	65	45	20
US33	123	34	30
US35	44	33	11
US36	116	61	37
US37	45	36	9
US38	21	14	7
US39	134	51	32
US40	151	53	28
US41	134	51	32
US42	47	38	9
US-43	47	38	9