Model-based safety analysis of a control system using Simulink and Simscape extended models

Nian Shao^{1,2}, Shuguang Zhang^{1,2}, and Hui Liang^{1,2}

¹School of Transportation Science and Engineering, Beihang University, Beijing, China

²Airworthiness Technology Research Centre, NLAA, Beijing, China

Abstract. The aircraft or system safety assessment process is an integral part of the overall aircraft development cycle. It is usually characterized by a very high timely and financial effort and can become a critical design driver in certain cases. Therefore, an increasing demand of effective methods to assist the safety assessment process arises within the aerospace community. One approach is the utilization of model-based technology, which is already well-established in the system development, for safety assessment purposes. This paper mainly describes a new tool for Model-Based Safety Analysis. A formal model for an example system is generated and enriched with extended models. Then, system safety analyses are performed on the model with the assistance of automation tools and compared to the results of a manual analysis. The objective of this paper is to improve the increasingly complex aircraft systems development process. This paper develops a new model-based analysis tool in Simulink/Simscape environment.

1 Introduction

Safety and reliability are essential for commercial aircraft design. The conventional safety analysis methods became very mature after long time research and development. However, with the increase of the complexity of aircraft systems, some limitations of these methods arise.

Fenelon [1] pointed out that the design process and the safety assessment process are usually not satisfactorily integrated. He used the expression “over the wall” to describe the gap between design and safety process. In addition, conventional safety analysis methods are faced with the problem of low efficiency.

With the problems of conventional safety analysis methods, an increasing demand of effective methods arises. One approach is the utilization of model-based technology. The ESACS [2] project and its following up project ISAAC developed some safety analysis methodology and tools in Simulink, SCADE, Statemate, NuSMV and AltaRica [3]. In the year 2005, Joshi et al. proposed the term “Model-Based Safety Analysis(MBSA)” for this kind of safety analysis method [4]. The European project MISSA [5] was dedicated to integrating the model-based security assessment into the system development process described by ARP4754A [6] and testing these processes in industrial examples. In 2016, Schallert [7] established safety analysis methods using multi-domain object-oriented models in Modelica.

MBSA becomes more and more popular and gradually gets the recognition from the authority. AC 20-

115C [8] released by FAA in 2013 formally recognizes the model-based methodology.

2 System safety assessment process

2.1. Typical system safety assessment process

Fig. 1. System safety assessment process described in SAE ARP-4761.

Fig.1 shows the overall safety assessment process according to ARP-4761 [9]. The whole process includes the requirements validation and the products verification. An aircraft-level Functional Hazard Analysis (FHA) is

* Corresponding author: gnahz@buaa.edu.cn

initially conducted, followed by a System FHA, then Preliminary System Safety Assessment (PSSA). During PSSAs, safety requirements are derived using Fault Tree Analysis (FTA) and probability budgeting. This is usually an iterative process including the PSSA and the system design definition and refinement.

After the determination of the final design, the compliance of the implemented products with the safety requirements are verified during the System Safety Assessment (SSA). During this process, the system Failure Modes and Effects Analysis (FMEA) provides for the failure probabilities of the single items. Those are the inputs for the system-level FTA and the aircraft-level FTA, which are conducted to verify that the systems and the overall aircraft satisfy the top-level safety requirements.

2.2 Utilization of MBSA in typical safety process

Simulation models mentioned in this article can be the formal specification in a model-based development. It is also available for assistance of conventional safety processes. The integration work is enlighten by the qSafe project [10].

In the PSSA process, the objective of MBSA is to assist the validation of system requirements. The model describes the system architecture and functional properties. The models are based on Simscape language, which provide the possibility for computer assistance. MBSA is introduced to conventional safety assessment process to assist the FTA process. MBSA can traverse all possible failure mode connections of a system without omission in a short period of time. More importantly, MBSA can generate minimal cut sets for FTA.

In the SSA process, FMEA and bottom-up FTA are applied for the verification of system implementation. MBSA focus on the physical implementation and specific performance of components and systems. Information of failure modes and effects can be identified through simulation and analysis. MBSA is utilised into the ARP-4761 [9] safety process to assist FTA and FMEA of complex multi-domain systems in the PSSA and SSA processes.

2.3 Simscape modelling

The complete work described in the article is developed in the MATLAB environment.

The models are built and modified in Simulink environment. The formal system model is built with Simscape. Simscape enables rapidly creation of physical system models within the Simulink environment. With Simscape, physical component models based on physical connections that directly integrate with block diagrams and other modeling paradigms can be easily built. Modelling of systems like electric motors and hydraulic actuators can be achieved by assembling fundamental components into a schematic. Furthermore, as the principle of Simscape is physical modelling, the resulting models are intuitively comprehensible without detailed knowledge of numerical simulation methods and

modelling paradigms, which benefits the interaction between design and safety analysis.

Simscape is one of the undirected numerical simulation models. Undirected numerical models describe systems as networks as functional elements that exchange energy through their ports. The component behaviour is encoded by differential algebraic equations. Every component is associated with some variables. Undirected numerical models distinguish protentional (voltage, velocity, pressure, etc.) and flow variables (current, force, flow rate, etc.).

3 A practicable MBSA modelling process using Simscape

3.1 Modelling of the exemplar control system

The example is a simplified control system model for a control surface (see also Fig. 2.). The function of the system is to control the position of the control surface (represented by the mass block m in Fig. 3.). There is a control circuit providing power and transmitting control signals. It includes a resistor, connected with a control voltage source and a fuse, as well as a motor driving the control surface to move. A sensor is connected to the motor to detect the angular displacement.

Fig. 2. A schematic diagram of the exemplar control system.

The system input are the pilot commands on the control column, which are simulated by a position command signal in the “Pos_cmd” block. Then the control signal is transmitted to the PI controller to control the voltage value of the voltage source. The voltage source drives the electric motor, and the motor generates a torque which is applied to the control surface.

Fig. 3. Simscape model of the exemplar control system.

The resulting angular displacement is measured by the sensor. The purpose of the damper block is to roughly approximate the dynamic behaviour of a real motor. The sensor will transmit the angular displacement

value to the controller as a negative feedback value used by the controlled to track the desired position signal: Once the command position is reached, the controller will give a constant value to the voltage source due to its integrating behaviour, and the position of the control surface will stay at the intended position. If the current in the circuit is too large, the fuse will cut off the circuit.

3.2 Fault modelling

3.2.1 Classification of component failures

For the effectiveness and usability of a simulation model, the failures should be modelled in a simple, intuitive way in case of very large system and numerous combinations of possible component states.

Schallert [7] establishes a catalogue of component failures for system safety analysis. This catalogue limits the possible failure modes of every component to three basic types of behaviour. This limitation on the failure numbers intends to avoid unnecessary modelling details and high computational effort during simulation, when combinations of failure modes are considered. A component will have 4 states in total.

Table 1. General component failure modes.

Mode	0	1	2	3
State	Normal function	Loss of function1	Loss of function2	Inadvertent function

Table 2. Components failure modes.

Mode	0	1	2	3
Sensor	Nominal	Loss of signal		Signal bias
Motor_mechanical	Nominal	Disconnection	Jam	
Motor_electrical	Nominal	Open circuit	Short circuit	
Resistor	Nominal	Open circuit	Short circuit	
Fuse	Nominal	Stuck open		Stuck close
Power supply	Nominal	Loss of output		

Mode 0 means the normal function or intact state of a component or function.

Mode 1 means a type of function loss, where the transducing function of a component is lost. This can be an electrical O/C ($i \rightarrow 0$) or mechanical disconnection ($f \rightarrow 0$ or $\tau \rightarrow 0$). For a sensor, mode 1 represents "loss of signal".

Mode 2 is also a type of function loss, where a transducing component is stuck in a certain state. It can be an electrical short to ground ($v \rightarrow 0$) or mechanical jamming ($v_e \rightarrow 0$ or $\omega \rightarrow 0$).

Mode 3 means some inadvertent activation of a component. This can be for example the "stuck close"

failure mode of a switch or an erroneous signal given by a sensor (bias).

In this simplified system. There are five components with failure modes: the energy source, the resistor, the fuse, the motor and the sensor. For convenience and efficiency of simulation, the motor is split into a mechanical part and an electrical part.

3.2.2 Failure mode control

To exhaustively simulate all the combinations of failure modes, a way to control the failure mode must be provided. Simscape enables customized components, which means customer can modify library components or create components. Extended models are mostly modified from Simscape library models. Simscape models are based on equations written in Simscape language (source code). A variable named "failure mode" is added to the source code of each component. The value of the variable will be assigned according to the mode it's intended to fail. The value of "failure mode" changes the equations of each component accordingly.

For example, the source code of the Simscape component Resistor is:

```
component resistor
    R = { 1, 'Ohm' }; % Resistance
end
equations
    assert(R>=0);
    v == R*i;
end
end
```

A resistor can fail short or fail open. According to the classification stated in subsection 4.2.1, the open circuit state of a resistor is represented as "FM_R = 1". Similarly, "FM_R = 2" means short circuit of a resistor. The resistor is extended by modifying the source code. The following source code represents a new component extended from the original resistor. The mode of this new resistor can be controlled by importing different FM_R. In this way, the failure mode of the component is controlled through the modality of its parameter. The source code of the extended resistor equation is as follows:

```
if FM_R == 1
    R = Roc;
else if FM_R == 2
    R = Rsc;
else if FM_R == 0
    R = R;
else
    error('id:id','invalid FM_R');
end
end
end
```

In similar way, failure models of a set of several other commonly used components are built and stored in a failure model library.

4

to plot the results and save the pictures for the analysis reports.

Yet it is not enough to have only the results plotted as time traces. The next goal is to automatically analyse the simulation results, i.e. the determination of the effects resulting from different combinations. The analysis results can also help to find out the corresponding combinations leading to a specific top event (i.e. the cut-sets).

Table 3. Failure effect codes.

FE No.	0	1	2	3	4	5
Failure Effect	Nominal	Follow wind	Stuck	Position bias	Unstable	No effect

A set of expected failure effects can be received from the FMEA and the simulation results. A unique failure effect code is assigned to each failure effect (see Table 3.).

Fig. 6 shows the “Stuck” failure effect, which is the consequence of motor “Jam” failure.

Fig. 6. Failure effect “Stuck”.

With this form, the minimum cut sets of a specific top event can be identified, which is the essence of a fault tree.

Table 4. Combinations of failure modes and failure effects.

S	M _m	M _e	R	F	E	FE
0	0	0	0	0	0	0
1	0	0	0	0	0	4
2	0	0	0	0	0	3
0	1	0	0	0	0	1
1	1	0	0	0	0	1
2	1	0	0	0	0	1
2	2	0	0	0	0	2
...	...	...	...	...	...	...

4 Conclusions

This paper develops a new model-based analysis tool in Simulink/Simscape environment. The objective is to improve the increasingly complex aircraft systems development process. Extended Simscape model is

utilized for the partial automation of typical safety assessment tasks.

During that process, some obvious advantages of model-based safety analysis became clear:

- The ability to exhaustively explore all possible combinations of the components failure modes using automated analysis tools. The quality of the safety analysis will be therefore improved.
- The ability to automatically generate analysis outcomes. These outcomes can be used to enhance or cross-check the manually created ones. Therefore, the method lightens the work load of safety analysts and reduce the costs, while simultaneously improving coverage and quality of the results.
- The possibility to integrate the safety process and the design process

References

- Fenelon, Peter, et al. "Towards integrated safety analysis and design." ACM SIGAPP Applied Computing Review 2.1 (1994): 21-32.
- Bozzano, Marco, et al. "ESACS: an integrated methodology for design and safety analysis of complex systems." Proc. ESREL. (2003).
- Akerlund, O., et al. "ISAAC, a framework for integrated safety analysis of functional, geometrical and human aspects." Proc. ERTS 2006 (2006): 1-11.
- Joshi, Anjali, et al. "A proposal for model-based safety analysis." Digital Avionics Systems Conference, 2005. DASC 2005. (The 24th. Vol. 2. IEEE, 2005).
- European Commission. MISSA More Integrated System Safety Assessment [EB/OL]. www.optics-project.eu/wp-content.
- Society of Automotive Engineers, "ARP5754A: Guidelines for Development of Civil Aircraft and Systems," (SAE Aerospace, 2010).
- Schallert, Christian. "Integrated safety and reliability analysis methods for aircraft system development using multi-domain object-oriented models." (2016).
- Federal Aviation Administration, "Advisory Circular 20-115C: Airborne Software Assurance," (FAA, 2013).
- Society of Automotive Engineers, "ARP4761: Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment," SAE International, 1996.
- Grigoleit, Florian, et al. "The qSafe Project–Developing a Model-based Tool for Current Practice in Functional Safety Analysis."(2016).