

A Reverse Modeling Method of MATLAB/Simulink Model for The Functional Resonance Analysis Method

MASAMICHI KAKESHITA^{1,a)} KENJI HISAZUMI² YASUTAKA MICHIIHIRO³ KEITA SAKEMI³
MICHIIHIRO MATSUMOTO³

Abstract: In recent years, Model-Based Development (MBD) has become widespread in control embedded systems, and development specifications are being written in MATLAB/Simulink and other tools. For socio-technical systems, FRAM (Functional Resonance Analysis Method) is a kind of safety analysis method. It is used as the analysis to foster success factors without being trapped in failures. Applying FRAM to MBD models is expected to improve the quality of MBD models from a new perspective, but the method of applying FRAM to MBD has not been established. Therefore, this study proposes a method to convert MATLAB/Simulink models into FRAM models by reverse modeling. As a result of actually applying the proposed method to the Simulink model, we proposed an improvement by creating and analyzing the FRAM model.

Keywords: Embedded System, Safety Analysis, MATLAB/Simulink, Functional Resonance Analysis Method

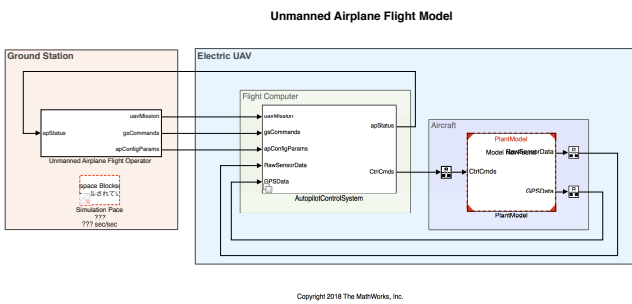


Fig. 1 Example of a Simulink model[1]

1. INTRODUCTION

Previously, we wrote development specifications for embedded systems in control systems using natural language during requirements analysis, followed by the design process. In recent years, however, model-based development (MBD) has gained popularity, and we now write development specifications in MBD models, such as MATLAB/Simulink or Simulink models instead. An example of a Simulink model is depicted in **Figure 1**. Conventional development specifications, which were earlier written in natural language, posed the risk of varied interpretations by various people, owing to their ambiguous descriptions. However, Simulink models can eliminate such risks because they are expressed in an elaborate manner.

The functional resonance analysis method (FRAM)[2] is a safety analysis method for sociotechnical systems. Traditional

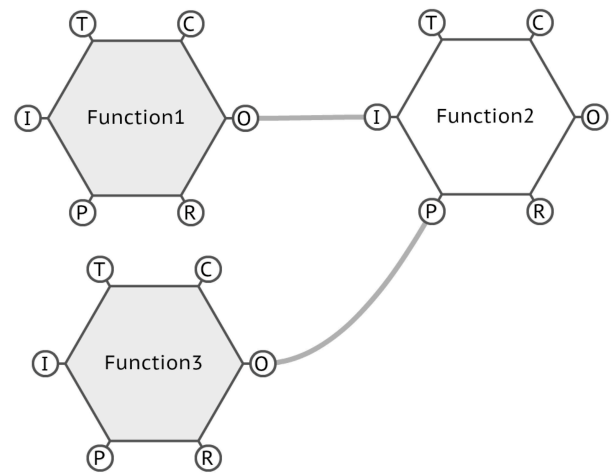


Fig. 2 FRAM Model

safety analysis methods, such as fault tree analysis [3], are based on the idea that poor outcomes result from failures; however, success often accounts for the majority of events. Furthermore, there are various cases in which successful events exist despite the occurrence of unexpected accidents. Therefore, it is not necessarily reasonable to focus only on failures. On the other hand, FRAM, as shown in **Figure 2**, defines functions and characterizes each function based on six different aspects of I, O, P, C, R, and T to create a model and conduct an analysis of the variation of these functions. Thus, it allows us to analyze without getting trapped in failures, and cultivate success factors, such as "how does it work well?".

As mentioned earlier, new safety analysis methods are emerging, and the application of these methods to Simulink models may lead to new discoveries. However, thus far, no method for applying FRAM to Simulink models has been established. In fact,

¹ Kyushu University, 744 Motoooka, Nishi-ku, Fukuoka 819-0395 Japan
² Department of Electronic Information Systems, Faculty of Systems Science and Engineering, Shibaura Institute of Technology, 307 Fukasaku, Minuma-ku, Saitama-shi, Saitama, 337-8570, Japan
³ Japan Manned Space Systems Corporation, 8F, Otemachi Bldg., 1-6-1, Otemachi, Chiyoda-ku, Tokyo 100-0004, Japan
^{a)} kakeshita@f.ait.kyushu-u.ac.jp

various difficulties have been encountered in the application of FRAM to Simulink models. The development specifications of MBD are often at a low level of abstraction. However, creating a FRAM model requires reverse modeling with a detailed understanding of the specifications' intent. Therefore, it is difficult to implement FRAM adequately. We herein proposed a method for converting a Simulink model into a FRAM model.

2. Methodology for creating a FRAM model using the Subsystem Table and Signal line Table

2.1 Step 1 : Identification of functions

In Step 1, we identify the functions in the FRAM model, assuming that the Simulink model is hierarchically organized by subsystem blocks.

2.1.1 Creating a Subsystem Table

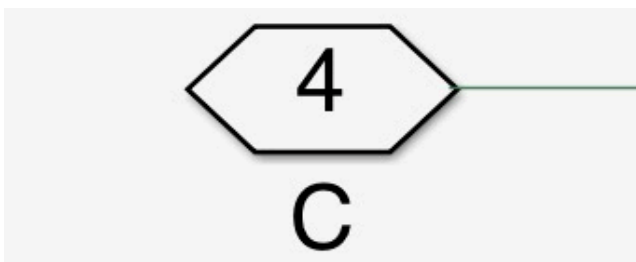


Fig. 3 Signal lines that are not treated as input/output

First, we create a subsystem table from the Simulink model to clarify the role of each subsystem and the state flow chart. Each entry in the subsystem table should contain the name of the subsystem, an overview of the roles played by the subsystem, the inputs and outputs of the subsystem, and the subsystems that comprise the subsystem. At this point, the signals handled by the terminals shown in **Figure 3** are not treated as inputs or outputs of the subsystem. This is because this port acts as a terminal to transfer across the boundary of the subsystem, known as the connection port, and does not affect the subsystem. This clarifies the purpose of each subsystem.

2.1.2 Creating a Signal line Table

Following the construction of the subsystem table, the signal line table is created. Each entry should contain information about the signals that appear in the subsystem table. Specifically, it describes which subsystem the signal belongs to, its signal name, role, unit, and whether there is an equivalent signal line, and if so, what it is. An equivalent signal line refers to a signal that branches from the same signal line. Thus, the aspect of each signal line to the subsystem can be clarified.

2.1.3 Function Abstraction

Once the subsystem and signal line tables are created, we are required to determine the level of abstraction of the functions that are to be converted into the FRAM model.

Currently, there are no definitive ways to determine this. As a general rule of thumb, we first determine the level of abstraction for each actor in the Simulink model. This is because we intend to observe how each actor is affected by other actors in the FRAM model and how it affects other actors. Within an actor,

it is better to have a higher hierarchy than a hierarchy consisting only of subsystems without return values. This is because many such subsystems only perform procedurally necessary operations and are unlikely to provide valid information when converted to FRAM models. In addition, the hierarchy should be such that internal interactions are not significantly lost. This is done to avoid the loss of information about how each subsystem affects the actors within each other. Convert the subsystem blocks and state flow charts in the determined hierarchy into FRAM based model functions on a one-to-one basis.

2.2 Step 2 : Classification of inputs and outputs

Once the abstraction level of a function is determined, we define the aspects of that function. Aspects should be mapped one-to-one to the inputs and outputs of the subsystem in the Simulink model. First, we classify the output into O. Next, we classify the signals that are direct numerical values or data that are the source of the output as I. We convert time-related signals to T. Furthermore, we define signals whose input would cause the entire subsystem to stop functioning, as P if the signal is discrete, and R, otherwise. If none of the aforementioned criteria are satisfied, the signal is assigned to C.

2.3 Implementation

By actually applying the proposed transformation method to the Simulink model, we created a FRAM model diagram and used it to perform safety analysis. For the safety analysis, we examined scenarios that can be assumed when each function fails. As a result, we were able to identify success and risk factors from the created FRAM model and propose improvement measures.

3. Conclusion

We converted the MATLAB/Simulink model into a FRAM model by creating a subsystem and a signal line table from the MATLAB/Simulink model, and identifying the functions and their aspects from the information in the tables. By applying this method to an actual Simulink model, we further created a FRAM model. In addition, by analyzing the FRAM model, we discovered problems in the system and suggested improvements.

References

- [1] The MathWorks Inc.: *simulinkDroneReferenceApp*, <https://github.com/mathworks/simulinkDroneReferenceApp>
- [2] Hollnagel, Erik: *FRAM: the functional resonance analysis method: modelling complex socio-technical systems*, CRC Press (2017)
- [3] Vesely, William E and Goldberg, Francine F and Roberts, Norman H and Haasl, David F: *Fault tree handbook*, Nuclear Regulatory Commission Washington DC (1981)