AN OVERVIEW OF THE PHASE-MODULAR FAULT TREE APPROACH TO PHASED MISSION SYSTEM ANALYSIS

Leila Meshkat
Jet Propulsion Laboratory
California Institute of Technology

Liudong Xing
Department of Electrical and Computer Engineering
University of Massachusetts – Dartmouth

Susan K. Donohue
Department of Systems and Information Engineering
University of Virginia

Yong Ou
Center for Railroad Safety Excellence
University of Virginia

Abstract

Phased-mission systems (PMS) are systems supporting missions consisting of multiple, consecutive, and non-overlapping operational phases. System composition, structure, and behavior can change from phase to phase; as a result, components may be modeled in different modes over the course of the mission. The need to model components and relationships in different modes poses unique challenges to analysis methods. We look at how fault tree analysis (FTA), a primary means of performing reliability analysis of PMS, can meet this challenge in this paper by presenting an overview of the modular approach to solving fault trees that represent PMS. Traditional solution methods are either combinatorial or Markov-chain based. The modular approach uses both combinatorial and Markov-chain solution methods as appropriate. This approach works by initially identifying modules that are independent throughout the PMS. This is accomplished by finding the union of all the components that are interdependent in any phase. It then solves for the reliability of each module within each phase and uses a combinatorial approach for finding the PMS system reliability based on the reliability of the modules. Further for solving for the reliability of the independent modules, if they correspond to static fault trees throughout all phases, it uses a combinatorial approach suitable for static PMS's (combined BDD's) and if they correspond to a dynamic fault tree in at least one phase, it uses a combined Markov Chain approach. Since the

reliability measure of every module is needed for feeding into the higher level static fault tree, it uses both the combined MC approach and the combined-BDD approach recursively to obtain the reliability measures of all the modules in a fault tree.

1. Introduction / Motivation

A phased-mission system (PMS) is a system that is used in a mission characterized by multiple, consecutive, and non-overlapping operational phases. A classic example of a PMS is an aircraft flight, which involves take-off, ascent, level flight, descent, and landing phases. The system may be subject to different stresses as well as different reliability requirements as it carries out its task(s) over the course of the mission. Thus, system configuration, success/failure criteria, and component failure parameters may change from phase to phase. Also, statistical dependencies exist across the phases for a given component. For example, the state of a component in the beginning of a new phase is identical to its state at the end of the previous phase. The complexity of the reliability analysis of PMS arises due to the dependencies across the phases for a given component and the system's dynamic behavior. Traditional methods of reliability analysis cannot cope with the system's dynamic behavior and dependencies, and therefore must be modified and/or extended. We present one such extension to fault tree analysis in this paper.

Two different classes of approaches have been developed for the reliability analysis of PMS: combinatorial and Markov-chain based. The combinatorial approaches, one example of which is the binary decision diagrams (BDD) based approach, are computationally efficient, but are applicable only when the reliability model for each phase is a combinatorial model. In other words, every phase of the PMS that can be solved by a combinatorial approach is a static phase, in which all components are functionally independent and the order in which failures occur does not matter. Markov based approaches capture the functional dependencies among components and the required order of failures, or dynamic failure criteria, using Markov-chain models. The major limitation with Markov based approaches is that if the failure criteria in only one phase are dynamic, then a Markov approach must be used for every phase. Due to the well-known state explosion problem of Markov approaches, it is often computationally intensive and even infeasible to solve the model.

The combinatorial and Markov-chain based approaches share a limitation, that is, both classes of methods assume statistical independence among the failures of the components within each phase. It's important to note that both classes of methods can capture the statistical dependence across the phases for a given component as well as the dynamic behaviors.

Motivated by the facts that most phased-mission systems are composed of both static and dynamic phases, and that combinatorial approaches and Markov approaches both have their pros and cons in the system modeling and analyzing, we introduce the phase-modular fault tree approach to phased-mission analysis in this paper. This approach

identifies modules of the fault trees that remain independent throughout the phase mission. It then finds the reliability of each independent module in each phase with an appropriate technique and combines the modules in a system level BDD to find the system reliabiilty measures.

We first cover the traditional approaches for solving fault trees in order to provide background for the presentation of the phase-modular approach. We outline the phase-modular approach using a simple example. We then apply this approach to a space mission example, and compare the results of that reliability analysis with those obtained via the traditional approaches. We finish by presenting conclusions and noting areas for future work.

2. *Background*

Fault trees were developed to facilitate the study of the Minuteman missile system; they have become, over the decades, a widely used, extremely useful, and a well studied analysis tool [1, 7, 8]. A fault tree is a logic diagram that describes the relationships between a potential critical event (accident) in a system and the reasons for this event [7]. It also provides a visual representation of the failure mechanisms of the system, which in turn facilitates the study of that system. Traditional or static fault trees provide a mathematical and graphical representation of the combinations of events that can lead to system failure. Dynamic fault trees [2] can also represent the failure behavior of the system that is related to the order or sequence in which events occur.

A fault tree consists of a top event, basic events, and the gates that connect the events. The top gate represents the system or subsystem failure we're interested in analyzing. This top event is connected to gates or basic events. Basic events are the basic causes of the failure. Gates are logical compositions of other gates and/or basic events. The top event is resolved into its constituent causes using these gates. Static gates include AND, OR, and M-out-of-N gates. Dynamic gates include FDEP, PAND, SEQ and SPARE gates [2]. A static fault tree has only static gates; a dynamic fault tree has at least one dynamic gate.

Solving a fault tree involves using appropriate techniques for finding the probability of occurrence of the top event based on the probability of occurrence of the basic events. Traditional or static fault trees can be solved with a variety of techniques such as sum of disjoint products, inclusion-exclusion methods, and BDD's [5]. The most efficient solution technique is the BDD. Fault trees that include dynamic gates cannot be solved with those techniques. Instead, dynamic fault trees are translated to Markov chains for solution. The biggest drawback of Markov models is that the number of system states increases exponentially with the size and complexity of the system being analyzed. It is for this reason that we integrate Markov chain-based solutions with combinatorial solutions wherever possible. The approach used for this integration is called modularization. Modularization is the process of finding the independent modules within a fault tree and solving the static modules with combinatorial techniques and the dynamic modules with Markovian methods and integrating the two to obtain the system reliability

result. Modularization algorithms have been developed which help identify the independent modules in a dynamic fault tree [3, 4].

For PMS with at least one dynamic phase, it's possible to convert the entire mission into a big Markov chain for solution. While this approach yields the exact reliability measure, it is inefficient and computationally extensive. Along the lines of the modularization approach for dynamic fault trees, it would be a good idea to solve each static or dynamic module with it's appropriate technique and integrate the results to obtain the system reliability measures. Nevertheless, it is important to take into consideration that the various phases in a PMS are not independent of each other. Meshkat [5] specifies interfaces between static and dynamic phases, which allow for the integration of results obtained by each to obtain the mission reliability measure. Ou and Dugan [6] present an approach to finding the modules that remain independent throughout the entire mission of a PMS. Those modules are then solved independently, and their results are combined to obtain the overall PMS reliability. This approach is the basis of our discussion in the next section.

3. *The Phase-Modular Approach*

The reliability of a PMS is the probability that the mission successfully achieves its objectives in all phases. In order to obtain this reliability measure, it is necessary to assess the reliability of each single phase and somehow link the models associated with the various phases together, or to treat the multiple phases as a single phase and analyze the system accordingly. The phase-modular approach takes the latter course.

We present the basic elements of the phase-modular approach in the following via a simple example PMS, which has three phases and seven components. Our discussion is based on [6].

1. Represent each mission phase by a fault tree; link the phase fault trees with a system top event. The fault tree for each phase is then divided into independent subtrees. Subtrees are identified as static or dynamic in different phases depending on their characteristics. For the example fault tree in figure 1, the fault tree of phase one includes two main modules {A,G,B,F} and {C,D}, which are both static. The fault tree in phase two includes two modules {A,B,F}, which is static and {C,E}which is dynamic. The fault tree in phase three includes three modules {A,G}, {B} and {C,D,E}, all of which are static.

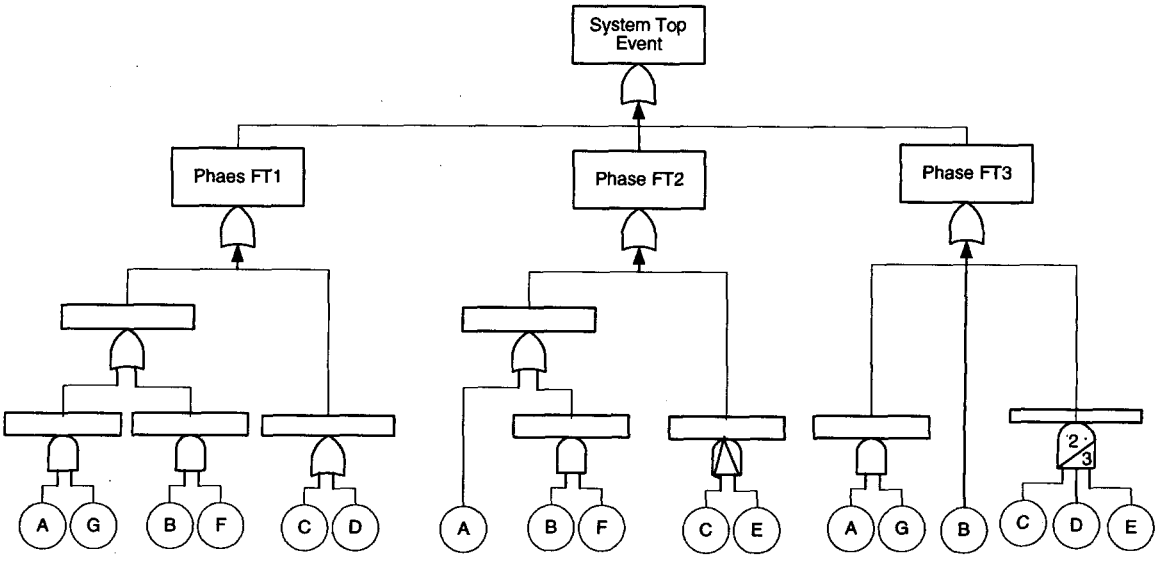

Figure 1 - Example PMS Fault Tree

2. Find the system-level independent modules. This is accomplished by finding the unions of the components in all the phase modules that overlap in at least one component.

   For instance, in our example PMS fault tree there are two modules, {A,G,B,F} and {C,D,E}. Note that in phase 3, the modules {A,G} and {B} are independent, but since they are not independent in other phases, they are lumped into one module.

3. Identify each phase module as static (all AND, OR, and/or K-OF-M gates) or dynamic (has at least one PAND, CSP, WSP, or HSP gate). Identify each phase module as bottom-level (has no child modules) or upper-level (has child modules). For the example fault tree, the module {C,D} in phase one is a bottom level module, but the module {A,G,B,F} is an upper level module since it has the child modules {A,G} and {B,F} which are each linked to a gate. The identification of the child and parent modules is important from the perspective

   The module {A,G,B,F} is static and {C,D,E} is dynamic. This is because components C and E are connected to a dynamic gate, the PAND gate in phase 2.

4. Find the joint phase module probabilities for all bottom-level modules. We use the BDD method on modules that are static across all the phases; the solution techniques are presented in [9, 10]. We use the combined Markov Chain method as presented in [5,6] on modules with at least one dynamic property. For example, we can use the BDD method on phase module {A,G,B,F}, since it has static behavior in all the three phases. And we must use the Markov-chain method

on phase module {C,D,E}, since it has dynamic behavior – a priority AND gate – in phase 2.

5. Consider each module a basic event of a static fault tree and solve the corresponding BDD to find the system reliability equation based on the reliability measures of the modules. Since we've already solved for the reliability measures of the modules in step 4, this step concludes our solution.

Figure 2 shows the modularized fault tree for our example PMS. Basically, each module is solved independent of the other module, but with consideration of its own behavior in previous phases. For instance, in order to find the reliability of $M12$, we use a combined BDD approach for $M11$ and $M12$, and in order to find the reliability of $M23$, we use the combined Markov Chain approach on $M21$,$M22$ and $M23$. We then consider solving the static PMS fault tree with the basic events $M11$, $M21$, $M12$, $M22$, $M13$, and $M23$ using the combined BDD approach. This is possible as we have already computed the reliability measures of the modules. It is important to note that solving this simple PMS fault tree without modularization would involve solving a Markov chain with approximately 128 states. The Markovs chain we currently have to solve with this approach have a maximum of 10 states.

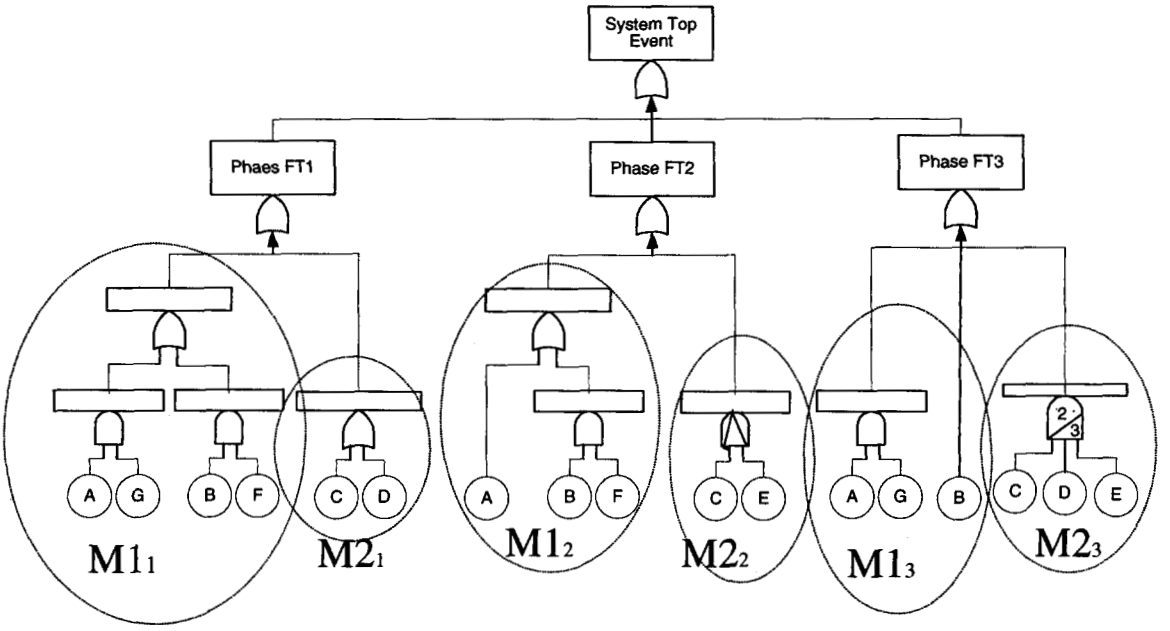

Figure 2: Modularized example PMS fault tree

4. Analyzing an Example Space Mission System

The following example fault tree (figure 3) has been drawn from data extracted from expert opinions about the risk elements of the Mars Smart Lander project (MSL-09). Note that the failure events considered here are a subset of the existing events that can

contribute to a failure. The system characteristics are not fully shown here; rather, we consider a very simplified version for demonstration purposes only.

We consider a three-phased space mission, which consists of the following phases: Launch; Cruise; and Entry, Descent, Landing (EDL). In the first two phases, the system can fail because of RPS induced failures. The RPS induced failures include thermal issues and radiation effects. During the launch phase, the system can additionally fail due to the launch vehicle failure. During the cruise stage, it can fail as a result of the OpNav issues or the Cruise stage related failures. During EDL, propulsion and avionics, thermal and radiation issues as well as the failure of hazard detection & avoidance issues can lead to a system failure. The hazard detection and avoidance issues occur as a result of the failure of both the LIDAR and the RADAR. The fault tree representing this space mission is shown in Figure 3. Each of the basic events connected directly to the top event constitute a static module. The RPS induced failure is a dynamic module.

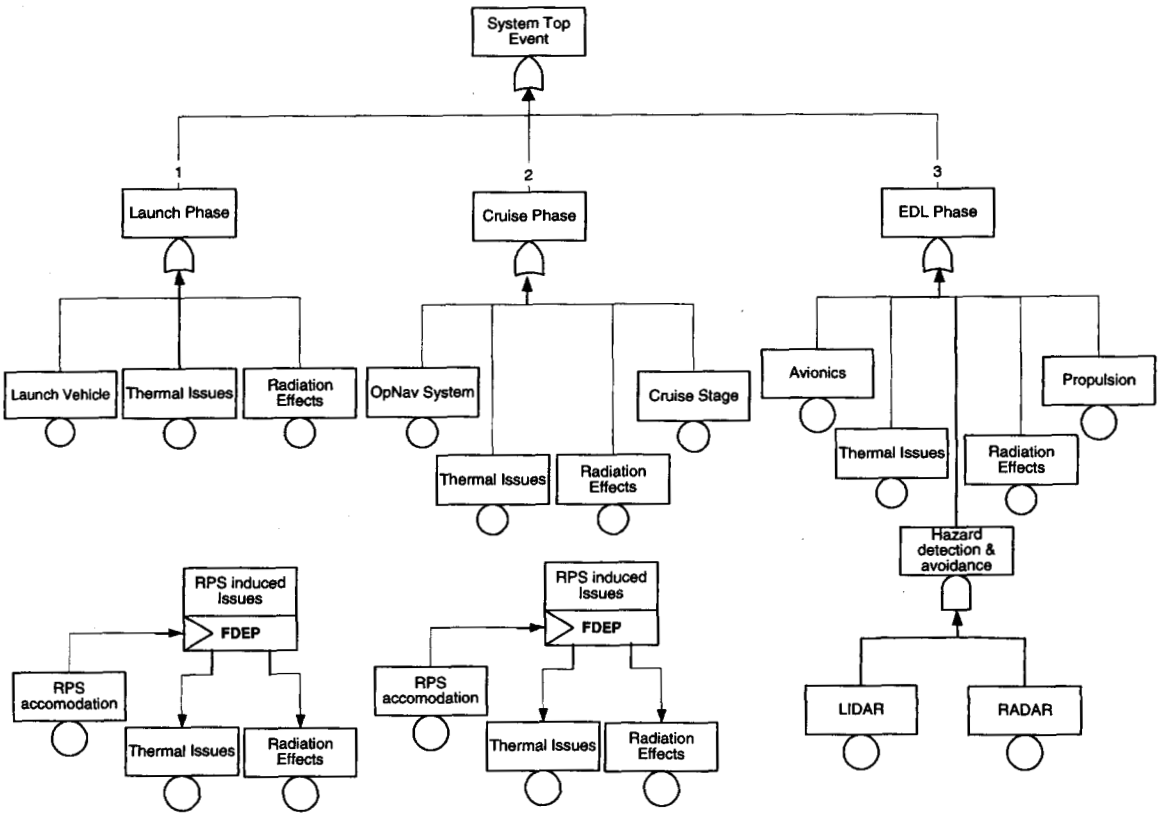

Figure 3 - Fault Tree for an Example Space Mission

(depending on space) We apply the methodology discussed in Section 3 in solving this fault tree. (just present Rsys or step through – these are the phase modules, here are the component probabilities, here are the joint probabilities, and here's rsys.)

Some analysis of the fault tree goes here.

5. *Conclusions and Future Work*

We present an overview of the phase-modular approach for solving fault trees for PMS in this paper. The phase-modular approach provides exact reliability measures in an elegant and efficient manner. It takes advantage of modular approaches for solving fault trees, and it takes into account the s-dependencies across phases. We present two examples: a generic example used in the explanation of the approach, and a basic example of a phased space mission. We hope to expand the phase-modular approach to incorporate other concerns addressed in traditional fault tree analysis, such as common cause failures and imperfect fault coverage.

Acknowledgments

We would first like to acknowledge our collective debt to our mutual advisor, Joanne Bechta Dugan. It's through her wise mentorship that we've been able to achieve so much.

This paper is a survey approaches developed by the authors at the University of Virginia and partially supported by NASA Langley Research Center under contract number NAS1-99098. Further, the work described in this paper was partially carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration

References

[1] R. E. Barlow and H. E. Lambert. *Introduction to Fault Tree Analysis.* Society for Industrial and Applied Mathematics, Philadelphia, PA 1975.

[2] Joanne Bechta Dugan, Salvatore Bavuso, and Mark Boyd. "Dynamic fault tree models for fault tolerant computer systems." In *IEEE Transactions on Reliability*, 41(3), September 1992, pp. 363 - 377.

[3] Yves Dutuit and Antoine Rauzy. "A linear-time algorithm to find modules in fault trees." In *IEEE Transactions on Reliability* 45(3),September 1996, pp. 422 - 425.

[4] Rohit Gulate and Joanne Bechta Dugan. "A modular approach for analyzing static and dynamic fault trees." In the 1997 *Proceedings of the Reliability and Maintainability Symposium*, pp 57 - 63.

[5] Leila Meshkat. *Dependency Modeling and Phase Analysis for Embedded Computer Based Systems.* Ph.D. thesis, Department of Systems Engineering, University of Virginia, Charlottesville, VA, August 2000.

[6] Yong Ou and Joanne Bechta Dugan. "Modular Solution of Dynamic Multi-Phase Systems." To appear in *IEEE Transactions on Reliability.*

[7] US Nuclear Regulatory Commission. *Fault tree handbook* (NUREG 0492). 1981.

[8] H. A. Watson and Bell Telephone Laboratories. *Launch Control Safety Study.* Bell Telephone Laboratories, 1961.

[9] Liudong Xing and Joanne Bechta Dugan. "Analysis of generalized phased-mission system reliability, performance, and sensitivity." In *IEEE Transactions on Reliability*, R-51(2), June 2002, pp. 99 – 211.

[10] X. Zang, H. Sun, and K. S. Trivedi. "A BDD-based algorithm for reliability analysis of phased-mission systems." In *IEEE Transactions on Reliability*, R-48(1), March 1999, pp. 50 – 59.

Biographies

Leila Meshkat
4800 Oak Grove Drive
Jet Propulsion Laboraboty
California Institute of Technology
Pasadena, CA 91109
Email: Leila.Meshkat@jpl.nasa.gov

Leila Meshkat is a Mission Systems Engineer at the Mission & Systems Architecture Section of the Jet Propulsion Laboratory. Prior to joining JPL, she was a Research Associate at the University of Southern California's Information Sciences Institute and a Lecturer at the USC School of Engineering. She holds a Ph.D. in Systems Engineering from the University of Virginia, an MS in Operations Research from the George Washington University and a B.S. in Applied Mathematics from the Sharif University of Technology. Her current research interests include Reliability and Risk Analysis. She is a member of IEEE and Omega Rho.

Susan K. Donohue
Department of Systems and Information Engineering
151 Engineers Way
Charlottesville, VA 22904 USA
E-mail: SusanD@virginia.edu

Susan Donohue is a Ph.D. candidate in the Department of Systems and Information Engineering at the University of Virginia. Her degrees include a B.A. in Political Science from Marquette University, Milwaukee, WI (1980) and a Master's of Engineering in Systems Engineering from the University of Virginia, Charlottesville, VA (2000). Her academic honors include memberships in Phi Beta Kappa, Alpha Sigma Nu, Pi Sigma Alpha, Phi Gamma Mu, and Omega Rho. Her professional affiliations include a student membership in IEEE. Her main research interests include software reliability engineering, QA/QC, and risk assessment and management.

Yong Ou
Department of Electrical and Computer Engineering
University of Virginia
Charlottesville, VA 22904 USA
E-mail: yo3u@virginia.edu

Yong Ou received his B.S. degree in Electrical Engineering from Huazhong University of Science & Technology, Wuhan, P. R. China, in 1990. He was awarded the M.S. and Ph.D. degree in Electrical Engineering from the University of Virginia, Charlottesville, in 2000 and 2002, respectively. Dr. Ou is a research scientist at the Electrical and Computer Engineering Department at the University of Virginia. He is a member of IEEE.

Liudong Xing
Department of Electrical and Computer Engineering
University of Massachusetts, Dartmouth
285 Old Westport Road, North Dartmouth, MA 02747-2300 USA
E-mail: lxing@umassd.edu

Liudong Xing received her B.E. degree in Computer Science from Zhengzhou University, Zhengzhou, China, in 1996, and became a graduate research assistant in Shenyang Institute of Computing Technology, Chinese Academy of Science, China from 1996 to 1998. She was awarded the M.S. and Ph.D. degree in Electrical Engineering from the University of Virginia, Charlottesville, in 2000 and 2002, respectively. Dr. Xing is currently an Assistant Professor of Electrical and Computer Engineering Department at the University of Massachusetts, Dartmouth. Her major research interests lie in the design and analysis of dependable computer-based systems and networks. She is a member of IEEE.