INL/CON-06-11743
PREPRINT

Space Propulsion System Phased-Mission Probability Analysis Using Conventional PRA Methods

PSAM 8 Conference Proceedings

James K. Knudsen
Curtis L. Smith

May 2006

The INL is a
U.S. Department of Energy
National Laboratory
operated by
Battelle Energy Alliance

SPACE PROPULSION SYSTEM PHASED-MISSION PROBABILITY ANALYSIS USING CONVENTIONAL PRA METHODS

James K. Knudsen

Idaho National Laboratory

PO Box 1625, Idaho Falls, ID 83415-3850

Curtis L. Smith

Idaho National Laboratory

PO Box 1625, Idaho Falls, ID 83415-3850

ABSTRACT

As part of a series of papers on the topic of advance probabilistic methods, a benchmark phased-mission problem has been suggested. This problem consists of modeling a space mission using an ion propulsion system, where the mission consists of seven mission phases. The mission requires that the propulsion operate for several phases, where the configuration changes as a function of phase. The ion propulsion system itself consists of five thruster assemblies and a single propellant supply, where each thruster assembly has one propulsion power unit and two ion engines. In this paper, we evaluate the probability of mission failure using the conventional methodology of event tree/fault tree analysis. The event tree and fault trees are developed and analyzed using Systems Analysis Programs for Hands-on Integrated Reliability Evaluations (SAPHIRE). While the benchmark problem is nominally a "dynamic" problem, in our analysis the mission phases are modeled in a single event tree to show the progression from one phase to the next. The propulsion system is modeled in fault trees to account for the operation; or in this case, the failure of the system. Specifically, the propulsion system is decomposed into each of the five thruster assemblies and fed into the appropriate N-out-of-M gate to evaluate mission failure. A separate fault tree for the propulsion system is developed to account for the different success criteria of each mission phase. Common-cause failure modeling is treated using traditional (i.e., parametrically) methods. As part of this paper, we discuss the overall results in addition to the positive and negative aspects of modeling dynamic situations with non-dynamic modeling techniques. One insight from the use of this conventional method for analyzing the benchmark problem is that it requires significant manual manipulation to the fault trees and how they are linked into the event tree. The conventional method also requires editing the resultant cut sets to obtain the correct results. While conventional methods may be used to evaluate a dynamic system like that in the benchmark, the level of effort required may preclude its use on real-world problems.

1. BACKGROUND INFORMATION

For a science mission to the outer solar system, an ion propulsion system is designed to reach the destination. The mission consists of seven mission phases, where each phase requires the propulsion system to operate for the entire phase duration or only part of the duration. Two of the seven mission phases require the propulsion system to operate only part of the phase. For these two mission phases, the propulsion system operates at the start of the phase up until the specified operating time expires.

The design of the propulsion system consists of 5 separate thruster assemblies and a single propellant supply. Each thruster assembly consists of one propulsion power unit (PPU) and two ion engines. The thruster assembly is in operation when the power propulsion unit is providing power to one of the ion engines. The other ion engine is in standby unless needed due to failure of the operating ion engine.

Each mission phase has a different success criterion for the thruster assemblies. For mission phase 1, two of the five thrusters are required for success. For mission phases 3, 4, 5, and 7, three of the five thruster assemblies are required for success. For mission phases 2 and 6, the thruster assemblies are shutdown and not required for propulsion. Table 1 provides the mission phase durations.

Table 1. Mission Phase Durations.

Phase	Duration (hours)	Propulsion System Operation
1	5520	5520
2	336	0
3	9043.2	9043.2
4	26280	13140 Propulsion (4A) 13140 No Propulsion (4B)
5	26858.5	25001 Propulsion 1857.5 No Propulsion
6	500 (plus 1857.5)	0
7	9501.5	9501.5

The operation of the thruster assemblies are thruster assembly 1 and 2 will operate in mission phase 1 and thruster assemblies 1, 2 and 3 will operate in the other mission phases requiring propulsion. In the event a thruster assembly fails, the next largest numbered thruster will start and take the place of the failed thruster assembly. In other words, if during mission phase 1, thruster assembly 1 fails, then thruster assembly 3 will take its place and if thruster assembly 3 fails, then thruster assembly 4 will take its place and so forth. If no further failures occurred except thruster assembly 1, then for mission phase 3, thruster assemblies 2, 3, and 4 will provide the propulsion.

2. MODELING OF PROPULSION SYSTEM

The propulsion system was modeled using Systems Analysis Programs for Hands-on Integrated Reliability Evaluations (SAPHIRE).[1] The propulsion system was modeled using standard fault tree convention, while handling the mission phases used an event tree. The event tree that was created to handle the mission phases is shown in Figure 1. The mission-phase event tree is straightforward. Mission phase 1 is questioned first and if it fails then the mission fails. If mission phase 1 succeeds, then mission phase 2 is queried. The event tree progresses through each phase accordingly until the success of the last mission phase. If this mission phase (mission phase 7) succeeds, then the mission is successful. Each sequence is solved individually to determine the probability of mission phase failure.

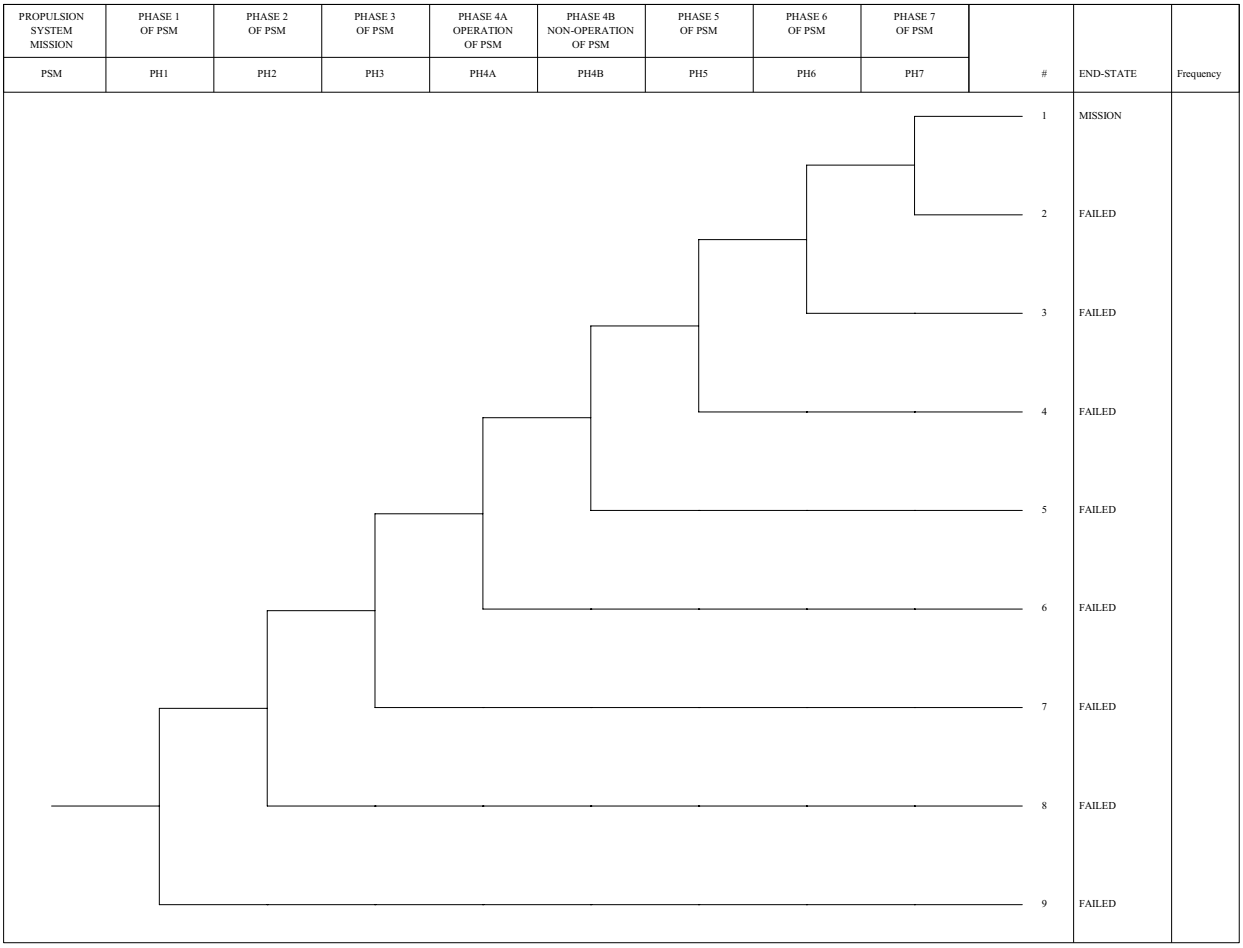

Figure 1. Mission phase event tree.

Fault trees were developed to account for each mission phase success criteria and fed into the mission phase event tree for evaluation. For mission phase 1, the success criterion is 2-of-5 thruster assemblies, which translates to a 4-of-5 failure gate. For mission phases 3, 4, 5, and 7 the success criterion is 3-of-5 thruster assemblies, which translates to a 3-of-5 failure gate. The fault trees for mission phase 2, part of 4, and 6 require the operating thruster assemblies to shutdown; therefore, these fault tree top gates are dependent upon the previous mission phase success criteria. However, for conservatism, the fault tree requires at least one of the operating thruster assemblies to shut down for success. The top gate for all of these mission phases was a 2-of-5 gate. Therefore, if two of the operating thrusters failed to shut down then these mission phases failed.

Individual fault trees were developed to model each of the five thruster assemblies, which transfer to the mission phase success criteria fault trees. These individual thruster assembly fault trees model the potential failures of the ion engines, propulsion power unit, and propellant supply. Each thruster assembly fault tree is identical for the mission phases that require propulsion except for the basic event names of the operating components. The different naming scheme is required in order to account for the different mission phase durations. The fault tree for thruster assembly 1 is shown in Figure 2 and can be viewed as a reference to the other thruster assembly fault trees.

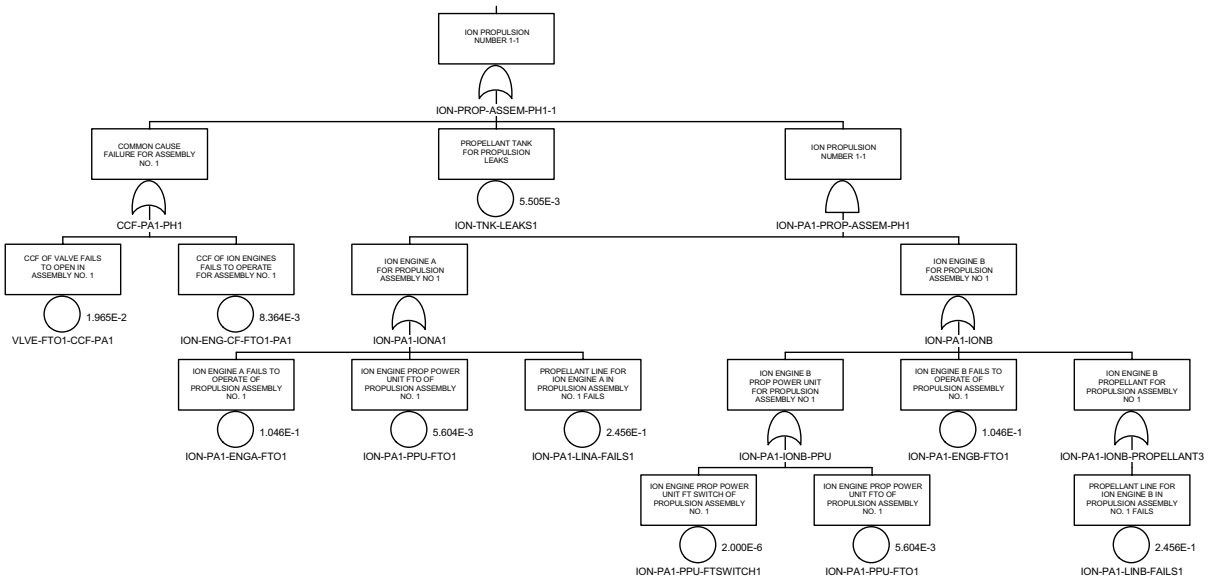

Figure 2. Fault tree for thruster assembly 1.

The fault tree shown in Figure 2 uses modules or compound basic events in order to eliminate each individual failure event. This allows for easier bookkeeping and fewer overall cut sets. The probability for these compound events utilizes the minimal cut set (mincut) probability equation within SAPHIRE. Therefore, the individual probabilities are correct, the only problem is many individual combinations are removed which slightly under estimates the fault tree top event probability.

Common cause failure was also modeled in the individual thruster assembly fault trees. Common cause failure was only considered across the two ion engines per thruster. Common cause failure of the ion engines A and B to start and run along with the propellant valves to open were modeled. Expanding the common cause failure across all ion engines or across the thruster assemblies was not modeled. The common cause conditional probabilities only went to four groups and to model across all of the ion engines would require a larger conditional probability group to account for this correctly. One could develop and add a common cause failure across the five thruster assemblies but many assumptions would have to be made along with correctly modeling what would cause an independent thruster assembly to fail. Once the independent thruster assembly failure was developed and quantified, the four-group conditional common cause probability would be used for mission phase 1 and the three-group conditional common cause probability would be used for mission phases 3, 4, 5, and 7. Because of the complexity and development of correct assumptions, only the independent ion engine failures within a thruster assembly were modeled as potential common cause failures.

3. QUANTIFICATION OF MISSION PHASE EVENT TREE AND FAULT TREES

The cut set generation and quantification of the cut sets for the individual fault tree mission phases and the event tree were performed. The results from this quantification showed that it is very unlikely that the mission would succeed. The results from the individual mission phase fault trees are listed in Table 2. Table 2 lists the mission phases, the SAPHIRE mincut probability of that mission phase, and then an evaluation of the cut sets using a binary decision diagram algorithm. A binary decision diagram (BDD) evaluation was performed (not in SAPHIRE) in order to calculate the exact probability of the mission phases. The SAPHIRE mincut calculation overestimates the numerous cut sets that have very high failure probabilities – hence its designation as an upper bound. A discussion of different attempts and assumptions used to analyze this model will be discussed below.

Table 2. Individual Mission Phase Fault Trees.

Mission Phase	SAPHIRE mincut Probability	Binary Decision Diagram Probability
1	8.5E-03	6.1E-03
2	1.2E-08	1.2E-08
3	3.3E-01	1.6E-01
4A	8.1E-01	3.8E-01
4B	1.2E-08	1.2E-08
5	1.0	8.8E-01
6	1.2E-08	1.2E-08
7	3.9E-01	1.5E-01

3.1. Fault Tree Quantification

The fault tree quantification process was straightforward. Once the fault tree models were developed for each of the mission phases, the minimal cut sets were generated and then quantified using SAPHIRE. The different mission phase probabilities are attributed to the different mission phase durations. Each fault tree was developed under the assumption that all five thruster assemblies are available for operation using the provided constant failure rate.

Mission phase 1 has the lowest failure probability, which equates to the largest potential for success (i.e., 1 - 8.5E-03 = 0.9915). This is due to only requiring two thruster assemblies for propulsion and the shorter mission duration. The dominant cut set for this mission is failure of the single propellant tank. The tank contributes 65 percent to the overall failure probability based on the mincut calculation in SAPHIRE.

Mission phases 2, 4B, and 6 all have the same failure probability. This is because of using the same requirement for success and that is shutting down at least one operating thruster assembly. Therefore, the top gate for these mission phase fault trees was 2-of-5. In other words, if two of the operating thruster assemblies failed to shutdown, then the mission would fail and continuation would not be possible. This success criterion is not correct for all of the mission phases but for handling this process in a static model it was assumed to be reasonable. The shutting down of the thruster assemblies is important for the next mission phase; however, in probability space these mission phase probabilities have little impact.

Mission phase 3 has a failure probability of 0.33 or a 67 percent chance of success. For this phase, three of the five thruster assemblies are required for propulsion. The dominant factor in the failure of this phase is the external leaking of the inlet valves for the ion engines.

The remaining mission phases (i.e., 4, 5, and 7) are also very unlikely to succeed. Mission phase 4A has a 19 percent chance of success and mission phase 7 has a 61 percent chance of success. Again these mission phases are

dominated by the external leaking of the ion engine propellant valves. For mission phase 5, the SAPHIRE mincut calculation assumes that there is a zero chance of success since the calculation equated to 1.0 (i.e., guaranteed failure). However, there are numerous non-independent cut sets, which causes the mincut equation to over-estimate the exact probability. Therefore, this mission phase was analyzed further using a developmental BDD solution package developed at Utah Sate University.[2] The BDD evaluation calculated the mission phase probability to be 0.88, given this phase a 12 percent chance of success.

3.2. Event Tree Quantification

The event tree cut set generation and quantification can be viewed as straightforward or requiring extra manual manipulation depending upon the method of choice. The straightforward process is to let SAPHIRE generate the cut sets for each sequence. Each failed sequence represents the failure of that mission phase. The fault trees that were linked into the event tree are the same ones discussed in Section 3.1. This process generates the same mission failure probability as discussed in Section 3.1 for the individual mission phase fault trees. Again, it was assumed using this straightforward approach that at the beginning of each mission phase all of the thruster assemblies were available for operation.

The other manual manipulation ways to quantify this model was approached using two different processes. Each process will be discussed. The reason these processes require more manual manipulation is based on the computational limit within SAPHIRE and the computer that was used to quantify the event tree.

The first process was to use the same fault tree for each mission phase (i.e., using the same event name) while only changing the success criteria from requiring 2-of-5 thruster assemblies for mission phase 1 to 3-of-5 thruster assemblies for the remaining mission phases. These fault trees were very straightforward in design and linking to the event tree. To evaluate the sequences this way, a special process flag was assigned to the different mission phase fault trees. This process flag in SAPHIRE allows for the success of the events to be carried throughout the evaluation. So going from mission phase 1 to mission phase 3 at least four of the five thruster assemblies must succeed. The failure of mission phase 1 is the same but for the mission to be successful different combinations of four thrusters must succeed.

Given at least four thruster assemblies are available for mission phase 3, mission phase 3 uses this information. The failure of mission phase 3 takes into consideration the different combinations that came from the success of mission phase 1. This information is used in the cut set generation for the failure of mission phase 3. For the success of mission phase 3, three of the five thruster assemblies must succeed. This information is carried through the remaining sequences. However, because of the failure criteria, these sequences are not generated. This is because using SAPHIRE's cut set generation algorithm, these sequences are not possible. Therefore, this attempt was not explored further.

The last attempt at solving the event tree model was to utilize the fault trees used in Section 3.1. These fault trees are similar to that above, except they use a different basic event name in order to account for the different mission durations. Again, the special process flag was assigned to the fault tree top events in order to account for the success of the thruster assemblies. When the sequence cut sets were generated, the number of cut sets grew exponentially. This is due to taking account for the success terms in the final cut sets. Once the cut sets were generated, SAPHIRE quantified them using the mincut calculation. SAPHIRE could only handle the first three mission phases. These mission phases are failure of mission phase 1, mission phase 2 and mission phase 3. The reason only these three mission phases were generated is due to the large number of cut set combinations by including all of the success terms.

If SAPHIRE were able to generate all possible combinations to account for the transition from one phase to the next phase, a major task would be left to evaluate the cut sets in order to prune them down to just those that are non-minimal. To handle this process, multiple post-processing rules would have to be developed. These rules would remove all of the illogical cut set combinations. This would be a large task considering the number of potentially generated cut sets. This process was not expanded primarily due to the large number of potential combinations and the overall failure probability from the fault tree evaluations. The overall failure probability from solving the

individual fault trees assuming all thruster assemblies are available at the beginning of each mission is already very close to 1.0 and expanding the cut sets out would not lower the failure probability.

4. CONCLUSIONS

The exercise to evaluate the space propulsion phased mission using a static model to this dynamic process proved to be very difficult. The fault tree or straightforward event tree evaluation could be performed but several limiting assumptions had to be made. The major assumption was every thruster assembly and both ion engines were available for propulsion at the beginning of each mission phase. The other assumption was the unavailability for each of the operating components. For each mission phase the component's failure probability used the specific mission phase duration. By splitting the component probabilities into each specific mission phase, this lowered the overall failure probability. This assumption would mostly affect the single propellant tank.

The use of SAPHIRE to generate cut sets using the same naming scheme for each of the thruster assemblies and allowing the success terms carry through in order to capture what thruster assemblies are available for each mission phase, showed failure in mission phases 4 and 5 could not occur. This is due to the success requirement for each phase. If three thrusters are required for success, then three thruster assembly failures would cause mission phase failure. Therefore, one cannot have three thruster assemblies fail and succeed at the same time in SAPHIRE (a static model). Therefore, these mission phases contained no cut sets.

The last event tree option has merit on evaluating the model but the number of cut sets and the process to manually manipulate these cut sets became quite large. Another problem with this option is SAPHIRE had a difficult time generating the number of cut sets required to solve the later mission phase sequences. This is due to computer memory and other factors relating to software/hardware issues. Therefore, only the first three mission phase sequences were able to be fully generated. These sequences were calculated in SAPHIRE to have a failure probability of 1.0 because of the number of cut sets and the mincut upper bound equation.

In conclusion, trying to use a static model on a dynamic process does not work very well. Many assumptions are required in order to generate the failed cut sets in order to have some reasonable estimate as to the failure (or success) probability of the individual mission phases and overall mission. The exercise showed the difficulties of trying to evaluate this problem without (a) using major simplifying assumptions in order to correctly solve the model or (b) using a tool designed for dynamic modeling situations.

ACKNOWLEDGMENTS

We would like to acknowledge the work done by Steve Prescott while at Utah State University to develop a software package to quantify fault tree logic structures using BDDs. Also, we would like to Catherine Kreiger from the University of Washington for her assistance on this analysis.

REFERENCES

[1] K.D. Russell, et al. *Systems Analysis Programs for Hands-on Reliability Evaluations (SAPHIRE) Version 6.0 - System Overview Manual*, NUREG/CR-6532, May 1999.

[2] S. Prescott, *Unordered Binary Decision Diagrams for Risk Analysis*, Master's Thesis, Utah State University, 2005.