How Long Should I Simulate, and for How Many Trials? A Practical Guide to Reliability Simulations

Kenneth E. Murphy, ARINC, Albuquerque
Charles M. Carter, ARINC, Albuquerque
Larry H. Wolfe, ARINC, Albuquerque

Key Words: Simulation, Steady State Availability, Endurability, Mission Reliability, Rules of Thumb, Horn of Variance, Horn of Confidence

SUMMARY AND CONCLUSIONS

We have provided RAM analysts with practical rules of thumb that facilitate the resolution of how long and how many trials are appropriate for simulations that focus on particular RAM parameters. The rules of thumb provide a structured methodology that determines a solution space as a function of simulation length and number of trials such that the value of the RAM parameter in question can be considered "good enough". What is defined to be good enough, depends on the analyst's tolerance for the magnitude of the error in the output. This paper provides the applied RAM analyst with a philosophy as to how to approach the resolution of the two most ubiquitous yet burdensome of RAM simulation questions.

1.0 INTRODUCTION

When analysts evaluate a system's RAM characteristics using simulation, they need to resolve two difficult questions, namely: the appropriate length of the simulation and the number of trials. These questions are not easy to answer explicitly since each system's RAM characteristics are unique. How do you know if you are simulating long enough? Is it better to do one long simulation or several trials of shorter duration? How do you determine that simulating more trials is wasting computer resources since your answer is "good enough"? The purpose of this paper is to provide some rules of thumb that should ease the answering of these two questions. These rules of thumb are the product of nearly twenty years of simulating numerous spacecraft, aircraft, and electronic systems. Whenever we are asked the questions of how long and how many trials, we always answer first by replying that "it depends". We need to know what RAM attribute is most relevant for a system before we can begin the process of answering these questions. For example, is the steady state availability of the system desired or the mean time between failure? These two RAM attributes require significantly different methodologies for answering the questions of how long and how many trials. In an effort to scope this paper, we will answer these questions as they pertain to four cases that we believe encompasses most practical reliability simulations. Even if a particular RAM parameter is not discussed within this paper, understanding the methodologies for the four cases will provide analysts with the capability to extend our processes to their simulations.

Acronyms

MDT	Mean Down Time
MTBDE	Mean Time Between Downing Event
RAM	Reliability, Availability and Maintainability
RBD	Reliability Block Diagram
SEM	Standard Error of the Mean
Stdev	Standard Deviation

2.0 THE FOUR CASE STUDIES

Case I focuses on non-steady state availability simulations or "endurability simulations" as it is known in some simulation arenas. This case is concerned with the system's availability over a relatively short span of time such that the system still contains the effects of its start-up transients. Case II concentrates on steady state availability simulations. This case is essentially the antithesis of Case I since we now desire the system's availability after the effects of the start-up transients have been effectively removed. Case III emphasizes mission reliability simulations, that is, the reliability of the system at some specified time. Finally, Case IV focuses on simulations where the system's mean time between downing events (MTBDE) or mean down time (MDT) is desired. In each of the four cases, we are interested in a particular RAM parameter that by its nature will dictate the methodology used. For the purposes of this paper, we chose an error of 1% to be the criteria for which simulation output is considered good enough with respect to our personal tolerances. The actual criteria value used is not relevant since the processes we will describe are valid independent of the chosen tolerance parameter. Similarly, we will use ± three standard deviations of the mean to bound a parameter, since in our case studies 99.7% accuracy is sufficient.

0-7803-6615-8/01/$10.00 © 2001 IEEE

2.1 *Example System and the RAPTOR Tool*

In an effort to make this paper more illustrative in nature, we will use a simple system (as shown in Figure 1) to resolve both questions for all four cases. We will also use a freeware reliability tool called RAPTOR to conduct the simulations.

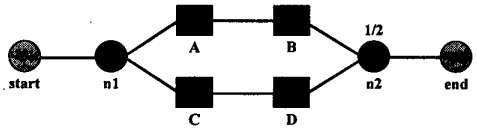

Figure 1. Example System in RAPTOR format

Figure 1 conforms to the Reliability Block Diagram (RBD) structure used by RAPTOR. The blocks in Figure 1 represent actual components of the system. They could be considered diodes, circuit cards, line replaceable units, or major systems in and of themselves, depending on the level of fidelity desired for a simulation. The circles are known as nodes and do not represent physical aspects of the system, but they do define the system's RBD structure. The start and end nodes effectively define the boundaries of the system (i.e., everything contained between is by definition the system). The node label "n2" defines the redundant relationship between string AB and string CD. Hence, Figure 1 represents four subsystems in which blocks A and B are in series as are blocks C and D. Furthermore, these two series strings are in a *1-out-of-2* redundant configuration.

Table 1 lists the RAM attributes that were utilized for each of the subsystems. Each block fails exponentially with a mean life of 90 hours and repairs lognormally with a mean of 10 hours and a standard deviation of 2 hours. The blocks have an infinite supply of spares so repairs start the instant a block fails. The blocks are listed as "independent" which in RAPTOR terminology means that these blocks continue to operate even if the system itself is down. The system is considered down whenever *both* series strings are down which implies that at least one of the blocks in each of the strings has failed.

	Failure Distribution	Mean	Repair Distribution	Mean	Stdev	Spares	Dependency
Block A	Exponential	90	Lognormal	10	2	Infinite	Independent
Block B	Exponential	90	Lognormal	10	2	Infinite	Independent
Block C	Exponential	90	Lognormal	10	2	Infinite	Independent
Block D	Exponential	90	Lognormal	10	2	Infinite	Independent

Table 1. Block's RAM Attributes

The RAPTOR simulator allows a system to be mimicked by operating the system for any length of time and number of replications (trials) desired. RAPTOR uses random numbers to represent random failure times of the blocks based on the distribution specified. The system operates until a block fails at which point two activities take place.

- The block that failed starts its random length repair cycle based on a random number drawn from its repair distribution.
- An evaluation of the system is conducted to determine if the failure of the block (or combination of blocks) resulted in the system being considered down.

This process of blocks failing and repairing continues until the end of a trial is reached, and then the process is repeated (using different random numbers of course) until the number of trials specified have been simulated. During this process, RAPTOR collects statistics on the system's up and down status and uses this information to produce system-level RAM characteristics.

Now that we understand the premise of this paper, the example system that will be used, and the underlying mechanisms of RAPTOR, we can begin the process of answering the questions of how long and how many trials for each of the four case studies.

2.2 *Case I: Endurability Simulations*

An endurability simulation seeks to determine the system's availability at a time that by definition still contains the effects of its start-up transients. These start-up transients are caused by all of the components at the beginning of the simulation being "good". All the blocks start a simulation trial as if they were brand new (i.e., no amount of life has been exhausted before the start of each trial). Although this is not relevant when using only exponential systems in series because of the memoryless property, readers are cautioned to recall that redundant systems (i.e., the system used throughout this paper) do not have failure times that are exponentially distributed.

Endurability simulations are often conducted for systems that must survive at some level of availability just up to a specified time. An example might be a satellite that must operate at 99% availability for three years after which time it is discarded for a newer version. Since by definition endurability simulations are conducted for a specified time, the question of how long to conduct a simulation trial is trivial. Analysts should conduct time-truncated trials of a length equal to the specified time the system must operate at some level of availability. For the satellite example, we would simulate the system for 24,300 hours (or three years) for n number of trials.

For this case study, we shall say that we want to know the availability of the system at a time equal to 24 hours. Thus, the only question that remains is how many trials are appropriate. To determine the number of trials (n) we recommended plotting what we call the "horn of variance" graph of availability versus the number of trials. We simulated eight different levels of n as shown in Table 2 and then plotted the plus and minus three standard deviations about the mean (i.e., standard error of the mean or $s/\sqrt{n}$) as well as the average availability observed. We plotted eight levels for consistency with the other cases, but in reality, you should only simulate enough levels to reach your personal tolerance. The horn of variance plot for this case is shown in Figure 2 and corresponding values are displayed in Table 2.

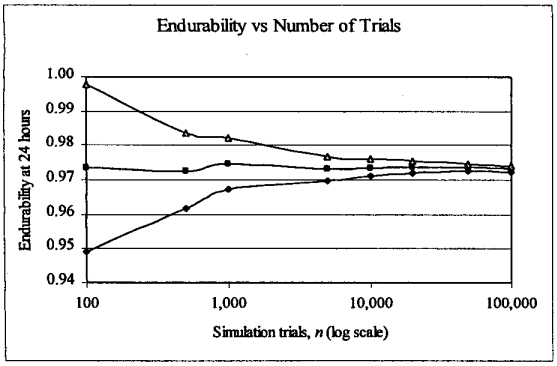

Figure 2. Endurability Horn of Variance

# of Trials	Minus 3 SEM	$Availability_{avg}$	Plus 3 SEM	Δ% Error
100	0.94888	0.97335	0.99781	2.51
500	0.96159	0.97263	0.98368	1.14
1,000	0.96700	0.97453	0.98206	0.77
5,000	0.96960	0.97315	0.97671	0.37
10,000	0.97111	0.97359	0.97607	0.25
20,000	0.97200	0.97376	0.97553	0.18
50,000	0.97251	0.97363	0.97474	0.11
100,000	0.97236	0.97315	0.97395	0.08

Table 2. Endurability Horn of Variance Values

For this case study, 1,000 trials were adequate since the plus and minus three standard deviations about the mean are within 1% of the average value. Although the answer depends on the level of error an analyst is willing to accept, a clear method for rationally determining the number of trials has been provided. Figure 3 summarizes the rule of thumb we use for endurability simulations.

Rule of Thumb: Endurability Simulations

1. How long (t) is by definition known.
2. Make a horn of variance plot of Availability vs Number of trials.
3. Progressively increase the number of trials (n) until personal tolerance achieved.

Figure 3. Endurability Rule of Thumb

2.3 *Case II: Steady State Availability Simulations*

A steady state availability simulation tries to determine the system's availability long after the effects of the start-up transients have passed. Since it often requires a long time to eliminate a system's start-up transients, running multiple trials which reintroduces the transients makes little sense. Hence, we recommend running one time-truncated trial (set $n = 1$) for a "very long" time. Violating the general simulation industry rule of never running just one simulation trial is justified in this case because we do not want to reintroduce the very forces that keep us from determining the long term steady state availability value. For this case study, the only question that remains is how long is a "very long" time. To answer this question we used RAPTOR's ability to produce an availability plot as a function of time.

Before we can begin the simulation, we must make an initial guess for an appropriate simulation length. Let us first simulate the system for the length of its useful life, which we will assume is 5 years (or approximately 44,000 hours). We are looking for the point at which time the average availability begins to level off. Figure 4 seems to indicate that around 35,000 hours the availability is stabilizing around 0.965. Thus after 35,000 hours of simulation, it appears that the start-up transients are no longer significant. Note how significant the transients are for the first 10,000 hours of simulation.

Now that we have determined the point at which the start-up transients are no longer relevant, we conducted a second simulation that discarded the statistics pertaining to the first 35,000 hours. Generally, we want the length of this second simulation to be a several times greater than the system's useful life. Thus, we conducted a time-truncated simulation of 132,000 hours in length (three times the system's useful life) while instructing RAPTOR to ignore the first 35,000 hours of results. The availability plot for the second simulation is shown in Figure 5.

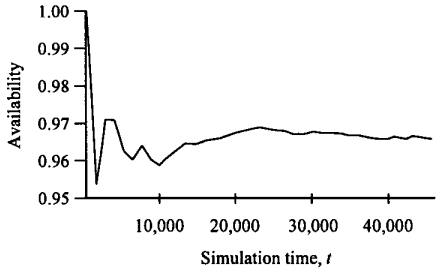

Figure 4. Availability vs Time With Start-up Transients

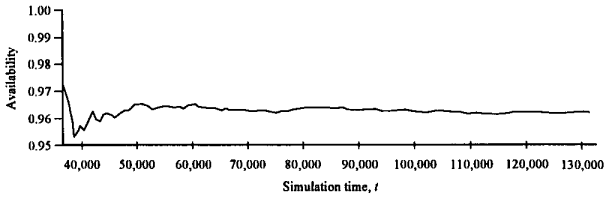

Figure 5. Availability vs Time without Start-up Transients

Figure 5 indicates that the availability of the system is tending towards 0.96. The RAPTOR results, shown in Figure 6, yield a little more precision for the availability parameter.

Results from 1 run(s):

Parameter	Minimum	Mean	Maximum	Standard Dev
Total Costs	533175.00	533175.00	533175.00	n/a
Ao	0.964100137	0.964100137	0.964100137	n/a
MTBDE	144.540515	144.540515	144.540515	n/a
MDT	5.382205	5.382205	5.382205	n/a

Total Sim Time=132000.000000, Time After Stats Reset=97000.000000

Figure 6. RAPTOR for Steady State Availability results

The true steady state availability for this system can be shown to be exactly 0.9639. Our simulation with the start-up transients produced an availability of 0.9655 or an error of 0.17%. When the start-up transients were removed, we achieve an availability of 0.9641 or an error of 0.02%.

The removal of the transients resulted in an error that was nearly nine times smaller. Both simulations resulted in an error of less than one percent but with steady state availability simulations, the error criteria usually is much more stringent. In either case, the analyst can see that simulating for a several multiples of the system's useful life with the start-up transients removed results in very accurate results.

For this case study we knew the true steady state availability for the system, but when this is not known, you should consider running a few (say five or ten) trials and use the availability variance as your error criteria. Figure 7 displays the results of 10 trials of 130,000 hours in length with a 35,000 hour delay statistics gathering implemented. As you can see, the standard deviation of 0.001 is very small especially if we are only interested in the third decimal place of the availability parameter.

Results from 10 run(s):

Parameter	Minimum	Mean	Maximum	Standard Dev
Total Costs	533118.00	533268.10	533366.00	75.89
Ao	0.961800105	0.964012905	0.966463194	0.001329539
MTBDE	132.520753	141.686922	149.041224	5.094635
MDT	5.171813	5.282817	5.398825	0.081566

Total Sim Time=132000.000000, Time After Stats Reset=97000.000000

Figure 7. Steady State Availability Results for Multiple Trials

Although the answer to what is an appropriate simulate length for this case depends on the level of error the analyst is willing to accept, a clear method for rationally determining this length has been provided. Figure 8 summarizes the rule of thumb used for steady state availability simulations.

Rule of Thumb: Steady State Availability Simulations

1. Number of trials (n) should at first be one by definition.
2. Make an availability plot for a simulation length equal to the system's useful life.
3. Determine duration of start-up transients that need to be removed.
4. Conduct a second simulation for a few trials that are a few times greater in length than the system's useful life and remove the start-up transients
5. Determine if the simulation length (t) is appropriate based on personal tolerance

Figure 8. Steady State Availability Rule of Thumb

2.4 *Case III: Mission Reliability Simulations*

A mission reliability simulation seeks to determine the system's reliability at a specified time that usually contains the effects of start-up transients. Thus, the question of how long to run the simulation is known and we only need to determine the number of trials that would be appropriate. We recommended plotting what we call the "horn of confidence" graph of mission reliability versus the number of trials. For this example, we shall say that we want to know the reliability of the system at 8 hours. We simulated eight different levels of n as shown in Table 3 and then plotted the 90% upper and lower confidence bounds and the average mission reliability observed. We plotted eight levels for consistency with the other cases, but in reality, you should only simulate enough levels to reach your personal tolerance.

It can be shown that the confidence bounds of a binomial point estimate (e.g., mission reliability) can be determined exactly using the following equations based on Fisher's distribution (Grosh).

$$R(t)_L = \frac{1}{1+\frac{r+1}{n-r}F_{\alpha/2,(2r+2,2n-2r)}} \le \hat{R}(t) \le \frac{F_{\alpha/2,(2n-2r+2,2r)}}{F_{\alpha/2,(2n-2r+2,2r)}+\frac{r}{(n-r+1)}} = R(t)_U$$

Equation 1

Where r is the number of failures, n is the number of trials, and α is the confidence criteria specified (α = 1- confidence, or 0.10 for this example). The horn of confidence plot is shown in Figure 9 and the actual values are listed in Table 3. The simulation trial variable was plotted on a log scale to increase visual understanding and does not change the conclusion drawn from Figure 9.

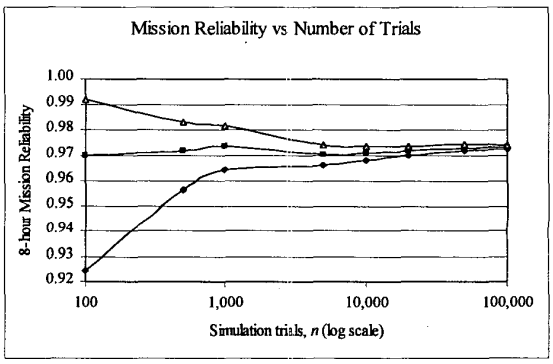

Figure 9. Mission Reliability Horn of Confidence

# of Trials	Lower Bound	R(8)	Upper Bound	Δ% Error$_{avg}$
100	0.92429	0.97000	0.99177	3.48
500	0.95657	0.97200	0.98299	1.36
1,000	0.96411	0.97400	0.98172	0.90
5,000	0.96615	0.97040	0.97424	0.42
10,000	0.96809	0.97100	0.97371	0.29
20,000	0.96995	0.97195	0.97384	0.20
50,000	0.97204	0.97326	0.97444	0.12
100,000	0.97272	0.97357	0.97440	0.09

Table 3. Mission Reliability Values

For this case study, 1,000 trials were adequate since the plus and minus three standard deviations about the mean are within 1% of the average value. Again, the answer depends on the level of error the analyst is willing to accept. Figure 10

summarizes the rule of thumb we use for mission reliability simulations.

Rule of Thumb: Mission Reliability Simulations

1. How long (t) is by definition known.
2. Make a confidence plot of Mission Reliability vs Number of trials.
3. Progressively increase the number of trials (n) until personal tolerance achieved.

Figure 10. Mission Reliability Rule of Thumb

2.5 *Case IV: MTBDE or MDT Simulations*

An MTBDE or MDT simulation seeks to determine the system's mean time between downing event and mean down time, respectively. Of the four cases considered in this paper, these are the most time-consuming simulations. The reason for these protracted simulations is that failure and repair distributions are often represented by statistical distributions that have very heavy tails (e.g., exponential, lognormal, etc.). A significant amount of simulation time is necessary to smooth out the extreme values that result in very long failure or repair times.

The best approach for determining a system-level MTBDE or MDT is for a large number of downing events to occur. This can be accomplished by conducting failure-truncated simulations as opposed to the time-truncated simulations implemented in the previous three cases. Failure-truncated simulations operate until a pre-specified number of downing events has occurred. Contrast this to time-truncated simulations that operate until a pre-specified time has elapsed.

The question of how long to simulate has been replaced by the question of how many downing events should be simulated. There are two approaches to answer this question. First, if the analyst is concerned with the steady-state MTBDE or MDT, then a single simulation trial with a "large" number of downing events should be simulated with delayed statistics gathering on the first "few" events. The effects of the first few downing events should be suppressed because these events are often larger on average than the subsequent downing events because the system starts with all components in a "good" state (i.e., start-up transients). The key to this approach is a single trial with enough downing events to overcome the potential bias of the first few events and to gain confidence in the steady-state value of interest.

A second approach, which we will focus on in this paper, is to be concerned with the mean time to the *first* downing event or repair. Thus, many trials of a single downing event failure-truncated simulation should be conducted. We recommended plotting a horn of variance graph of MTBDE and MDT versus the number of trials. We simulated eight different levels of n as shown in Tables 4 and 5. We plotted the plus and minus three standard deviations about the mean as well as the average MTBDE and MDT observed. We plotted eight levels for consistency with the other cases, but in reality, you should only simulate enough levels to reach your personal tolerance.

The horn of variance plots for this case study are shown in Figures 11 and 12 and their corresponding values are displayed in Tables 4 and 5.

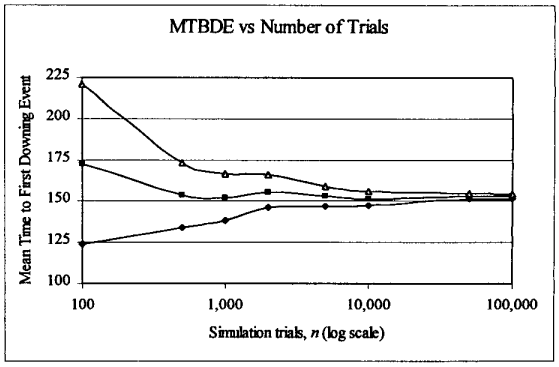

Figure 11. MTBDE Horn of Variance

# of Trials	Minus 3 SEM	$MTBDE_{avg}$	Plus 3 SEM	Δ% Error
100	123.775	172.226	220.677	28.13
500	133.883	153.537	173.191	12.80
1,000	138.145	152.182	166.219	9.22
2,000	145.880	155.839	165.798	6.39
5,000	146.702	152.987	159.271	4.11
10,000	147.173	151.647	156.121	2.95
50,000	151.156	153.188	155.219	1.33
100,000	151.721	153.142	154.564	0.93

Table 4. MTBDE Values

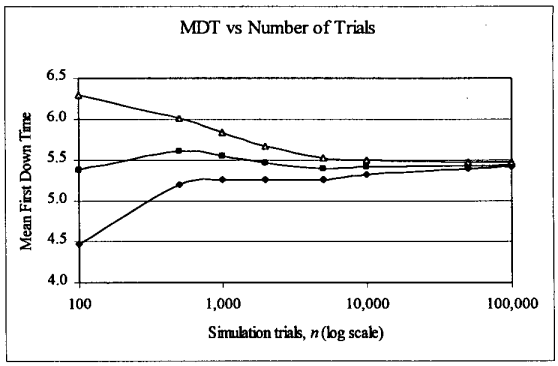

Figure 12. MDT Horn of Variance

# of Trials	Minus 3 SEM	MDT_{avg}	Plus 3 SEM	Δ% Error
100	4.469	5.384	6.299	16.99
500	5.202	5.605	6.008	7.19
1,000	5.264	5.552	5.840	5.19
2,000	5.261	5.467	5.674	3.78
5,000	5.265	5.395	5.525	2.40
10,000	5.323	5.415	5.507	1.71
50,000	5.390	5.432	5.474	0.77
100,000	5.416	5.446	5.475	0.54

Table 5. MDT Values

For this case study, 100,000 trials was adequate for the MTBDE parameter since the plus and minus three standard deviations about the mean are within 1% of the average value. For the MDT parameter, 50,000 trials are sufficient. Again, the answer depends on the level of error the analyst is willing to accept. It should be noted that this case study required significantly more trials to achieve an error of 1% compared to the other three cases. Figure 13 summarizes the rule of thumb we use for mission reliability simulations.

Rule of Thumb: MTBDE or MDT Simulations

1. How long (t) is not relevant.
2. How many failures per trial (k) is known for failure truncated simulations.
3. Make a horn of variance plot of MTBDE or MDT vs Number of trials.
4. Progressively increase the number of trials (n) until personal tolerance achieved.

Figure 13. MTBDE or MDT Rule of Thumb

3.0 SYNOPSIS

In each case study, one of the two questions was reasoned to be known. The remaining question was resolved by providing a rule of thumb that generated a solution space that contained the answer with respect to the level of error that can be tolerated. While these rules of thumb focused on the error associated with an output parameter, analysts should not forget to compare their results to the error of their inputs. What sense does it make to simulate a model that achieves an MDT that has an error of a few seconds when the error in the data collection process is at best five minutes. Analysts should always confirm that their answers are consistent with the errors associated with a simulation's input values. For example, simulation results might indicate that 100,000 trials are needed to acquire an MTBDE with a 1% error, but 10,000 trials yields an error of 3%. This 3% error may be good enough compared to the error associated with the input values.

Our main goal of this paper was to provide practical methodologies that yield solutions to the difficult simulation questions of how long and how many trials. By using a simplistic system and a free simulator, we have provided the readers the ability to repeat our work and thus solidify their understanding of this subject.

REFERENCES

1. RAPTOR Software, Version 5.0, ARINC

2. Doris L. Grosh, *A Primer of Reliability Theory*, John Wiley & Sons, New York, 1989, pp 239-244.

BIOGRAPHIES

Kenneth E. Murphy
ARINC
2309 Renard Place SE, Suite 200
Albuquerque, NM, 87106, USA

kmurphy@arinc.com

Kenneth E. Murphy graduated from the University of Colorado with a Bachelor of Science degree in Aerospace Engineering in 1989. In 1990 he graduated from the University of Alabama with a Masters of Science in Systems Engineering and a minor in Reliability Engineering. Ken was a reliability engineer for the United States Air Force for eleven years where he introduced his visions of the way reliability analysis and testing should be conducted. Ken developed the theory of a quick but robust reliability simulator in 1992 and three years later his idea became a reality when the free reliability software tool called RAPTOR was distributed to the world. In 1996 Ken became the chief reliability engineer for the Air Force's operational testing agency where he continued to lead the development of RAPTOR 3.0 and in 1999, the development of version 4.0. Ken is currently a principal reliability engineer with the ARINC Corporation where he is building a reliability skunk-works shop responsible for the development of RAPTOR version 5.0, analysis consulting, and RAPTOR training.

Charles M. Carter
ARINC
2309 Renard Place SE, Suite 200
Albuquerque, NM, 87106, USA

ccarter@arinc.com

Charles M. Carter has a B.S. in Aeronautical and Astronautical Engineering from the University of Illinois and a M.S. in Systems Engineering from the Air Force Institute of Technology. As a commissioned officer in the USAF, he worked as a Titan satellite launch engineer at Vandenberg AFB, CA and as a reliability engineer at Air Force Operational Test & Evaluation Center at Kirtland AFB, New Mexico. Chuck was a payload safety engineer evaluating space shuttle and International Space Station payloads while working for SAIC at Johnson Space Center. Chuck is currently a principal reliability engineer with the ARINC Corporation where he is responsible for the development of the RAPTOR simulation engine.

Larry H. Wolfe
ARINC
2309 Renard Place SE, Suite 200
Albuquerque, NM, 87106, USA

lwolfe@arinc.com

Larry Wolfe is currently employed as a Staff Principal Analyst with ARINC in Albuquerque, New Mexico. He holds a bachelors degree in Chemistry from Oregon State University and a masters degree in Business Administration from Central Michigan University. Mr. Wolfe is market lead for the new RAPTOR+ with ARINC. He has considerable RAM experience after 24 years with the United States Air Force and Chief of the Air Force Operational Test Center's Reliability Division. Mr. Wolfe is the ARINC corporate lead for Society of Logistics Engineers (SOLE) and a SOLE member.