Applying Monte Carlo Simulation to Launch Vehicle Design and Requirements Verification

John M. Hanson[1]
Marshall Space Flight Center, Huntsville, Alabama, 35812

and

Bernard B. Beard[2]
ARES Corporation, Huntsville, Alabama, 35805

This paper is focused on applying Monte Carlo simulation to probabilistic launch vehicle design and requirements verification. The approaches developed in this paper can be applied to other complex design efforts as well. Typically the verification must show that requirement "x" is met for at least "y"% of cases, with, say, 10% "consumer risk" or 90% confidence. Two particular aspects of making these runs for requirements verification will be explored in this paper. First, there are several types of uncertainties that should be handled in different ways, depending on when they become known (or not). The paper describes how to handle different types of uncertainties and how to develop vehicle models that can be used to examine their characteristics. This includes items that are not known exactly during the design phase but that will be known for each assembled vehicle (can be used to determine the payload capability and overall behavior of that vehicle), other items that become known before or on flight day (can be used for flight day trajectory design and go/no go decision), and items that remain unknown on flight day. Second, this paper explains a method (order statistics) for determining whether certain probabilistic requirements are met or not and enables the user to determine how many Monte Carlo samples are required. Order statistics is not new, but may not be known in general to the GN&C community. The methods also apply to determining the design values of parameters of interest in driving the vehicle design. The paper briefly discusses when it is desirable to fit a distribution to the experimental Monte Carlo results rather than using order statistics.

Nomenclature

Symbols

σ	=	standard deviation
σ_z	=	standard deviation of z
F_{BIN}	=	cumulative binomial probability
$G(x)$	=	cumulative distribution function
k	=	number of failures
m	=	order statistic of interest, m^{th} order statistic
n	=	sigma level of the z vehicle target value
N	=	number of Monte Carlo samples
P_{BIN}	=	binomial probability density function
p	=	actual failure probability
$\hat{p}$	=	maximum likelihood estimate of p
p_A	=	allowable or acceptable failure probability (when only a consumer value is specified)
p_c	=	required failure probability for the consumer

---

[1] Aerospace Engineer, Flight Mechanics and Analysis Division, MSFC/EV40, Huntsville, Alabama 35812, AIAA Member.
[2] Senior Consultant, ARES Corporation Tennessee Valley Office, Huntsville, Alabama 35805.

p_p	=	allowed failure probability for the producer
x_i	=	input parameters that are allowed to vary for a vehicle model
z	=	desired target value of a vehicle parameter (such as payload or maximum dynamic pressure)

Acronyms

CDF	=	Cumulative Distribution Function
CR	=	Consumer Risk
OC	=	Operating Characteristic
PDF	=	Probability Density Function
PR	=	Producer Risk

I. Introduction

MONTE Carlo simulation, where uncertain inputs are randomly varied and many samples are taken in order to obtain statistical results, is widely used in simulating complex space systems. There are many things to get right in setting up and making Monte Carlo runs that will not be covered here, including making sure all the uncertainty values used are appropriate (and apply the right level of conservatism), taking care in modeling the random inputs correctly, checking all the models to make sure they correctly model reality, verifying the simulation, and others. This paper does not cover alternative approaches to Monte Carlo and alternative ways of generating the sample cases (different from randomization using the presumed distribution for each uncertain parameter).

This paper is focused on applying Monte Carlo simulation to launch vehicle design and requirements verification. NASA requirements statements typically take the form of "The vehicle shall do x." An example might be that the vehicle delivers a required mass to a specified orbit, or that certain stability margins are met, or that navigation errors do not exceed certain limits. Then the verification requirement statement says (for example) "x shall be verified through analysis. Analysis shall use Monte Carlo simulation and show that x is met for at least y% of cases, with 10% consumer risk (CR)." Consumer risk is the risk that the verification (based on a finite number of samples) says the requirement is met when it is actually not met. (Statisticians sometimes call this a "Type II" error.) 10% CR is the same as 90% confidence that the requirement is in fact met if the analysis indicates it is met. Sometimes a higher fidelity test might be used to validate that the Monte Carlo simulation truly reflects reality, but that is not the focus of this paper. Besides verifying that a requirement is met, during vehicle design, the desire is also to identify probabilistic design cases (for example, the 99.865% worst structural conditions, heat rates, actuator angles, propellant usage, etc.), typically also with 10% CR.

The reason that there is a consumer risk statement is that if we ran the Monte Carlo run a second time with different random seeds, we would get a different answer. A 10% CR means that if we ran, say, 10,000 Monte Carlo runs of 2000 samples each to measure success in meeting the requirement, 10% of the runs would result in a violation of the requirement. Without a CR (or confidence) requirement, it would be possible to run a very small number of Monte Carlo samples and declare success. The converse of CR is "producer risk" (PR), the risk that we think a product does not meet the requirement when in fact it does.

To keep the CR low, we typically over-design the system (adding PR). One way to reduce the impact of the over-conservatism is to run a very large number of Monte Carlo samples (assuming the system and all uncertainties are modeled correctly; otherwise we are just playing with incorrect statistics). But this option is limited by computer power, particularly when a large number of Monte Carlo runs are already necessary in order to capture a variety of scenarios. It is important to run some cases with a larger number of Monte Carlo runs (or run some cases with different random numbers) in order to see how much the design is driven by this numerical conservatism.

Two particular aspects of making Monte Carlo runs for requirements verification will be explored in this paper. First, there are several types of uncertainties that should be handled in different ways:

1. Items which are not known during the vehicle design phases, but will be known for a specific vehicle (before flight) and can be used in the trajectory design and in understanding margin or adjusting payload. Examples include uncertainty in axial force that will be reduced by the time the vehicle flies (through wind tunnel tests and maybe test flights) and uncertainty related to the overall ensemble of engines but for each specific engine will be known (because the engine is put on a test stand and fired prior to flight). Components will be weighed. Of course, there will be no improved information about the assembled vehicle if these preflight tests do not occur. Some of this is introduced in Ref. 1.

2. Mixture ratio variation for each engine, which is also known prior to flight (part of the variation is known prior to flight, based on the engine test, and part is still unknown on flight day) but does not affect the

trajectory design or nominal payload capability. It does affect how much of each propellant will remain, and this knowledge should be used in the flight day go/no go decision.

3. Variations of temperature (affecting solid motor performance) and winds that are known for the flight day go/no go decision and for day of launch trajectory design but would not be used to adjust payload, since the payload is decided well before flight day.

4. Items that are unknown on flight day. It is also possible to break these items up into two components: those for which the true value randomly varies from flight to flight (for example, the variation in winds between the first and second wind) and those for which there is a single value that does not vary much from flight to flight but is unknown (for example, the lift moment coefficient).

This paper will discuss ways to model these different uncertainties. Included will be a discussion of vehicle models that are based on the preflight-known variations and ways to develop these models.

The Monte Carlo method necessarily produces outputs that are statistical in nature. That is, a Monte Carlo run consisting of N sample trajectories enables an approximate estimate of the properties of the distributions of various output parameters. Most engineers are familiar with the usual measures of central tendency, such as mean and median, as well as measures of dispersion, such as the standard deviation. The mean of the sample provides an unbiased estimate of the mean of the modeled population, with an uncertainty that is given by the sample standard error (equal to the sample standard deviation divided by the square root of N). What is less well-known among engineers is how to deal with the statistics of the tails of the distributions of output parameters – which are often more important than the means when verifying vehicle requirements. That is one of the major topics of this paper.

Consider a parameter that has a Gaussian distribution. The Gaussian distribution is unbounded. However, any finite set of N samples will have a largest sample. In fact the max-of-N will have a distribution that is related to the underlying Gaussian, but is not itself Gaussian. The next largest sample will have a slightly different distribution, and so on. The branch of statistics that deals with these distributions is called "order statistics" (Ref. 2), and is closely related to the field known as "sampling theory".

One of the first decisions faced by an analyst setting up Monte Carlo runs is the number of samples that are required in order to achieve a desired level of accuracy in the output parameters. Some measures, such as the sample mean, are known to improve in accuracy as the inverse of the square root of N (at least, for parameters that obey the central limit theorem). However, since the standard deviation (the numerator in the standard error) is itself a complicated function of the input parameters, it is generally not possible to predict in advance of the simulation how many samples are needed to reach a desired accuracy. It would seem at first glance that the situation in the tails of the distribution is even less tractable. However, it turns out that it is possible to relate the order statistics of the sample to the underlying distribution in a way that allows one to pick a sample size *a priori*. The sample size is chosen to provide an estimate of the value of an output parameter that bounds a required fraction of the underlying distribution, such as 99.865%, while meeting a required confidence level, such as 90% (i.e. 10% CR).

This paper explains this process and shows how to generate the information needed for verifying requirements success. This same procedure can be used to determine parameters needed for creating the vehicle design, for example the 99.865% (with 10% CR) highest acceleration.

Sometimes it is desirable to derive a distribution rather than to use order statistics, for example when the required failure percentage is extremely low (e.g., 1e-6) and the Monte Carlo results fall well within the success range (e.g., the Monte Carlo run consists of 2,000 samples). Without a huge number of Monte Carlo samples (likely requiring more than the time available to make the runs), order statistics could not be used. In this case it makes sense to fit a distribution to the data and to estimate where the 1e-6 level is reached. There will, of course, be significant uncertainty in this estimate due to the size of the sample.

In general, the good thing about order statistics is that they are non-parametric, i.e., they don't depend on an assumption or model of the underlying probability distribution. The downside is that we don't use most of the information we "bought" with our computer time, and consequently there is an undesirable tradeoff between conservatism and CR. The nice thing about fitted distributions (parametric statistics) is that they use basically all the information from the generated data, and of course the bad thing is that the uncertainty associated with the choice of distribution is difficult to control. One can test for normality, although necessarily there will be a certain number of false negatives in those tests, where we accept a distribution as being normal when it is really not. If the data are not Gaussian, it may be more appropriate to fit a distribution to the tail (say, the outlying 10%) of the sampled data rather than fitting the whole distribution (Ref. 3).

This paper will not focus on fitting distributions to data, but rather will explain the order statistics approach along with the vehicle modeling approach as a function of type of uncertainty that was discussed earlier.

II. Part 1: Setting Up Vehicle Models and Monte Carlo Runs

A. Vehicle Models

For a launch vehicle design, propulsion parameters are not well known at the start of the design effort. They will be better known by the time the vehicle is launched, but there will still be uncertainty on launch day. The design uncertainties will be gone by the time we launch and what remains is flight day uncertainty. Also, variations in the propulsion parameters for the ensemble of engines may be better known for a particular engine if it is tested prior to flight. The trajectory will be designed for the known variations. The flight performance reserve (the extra propellant needed on-board to cover for uncertainties) should be designed to cover for flight day uncertainties, not for all the uncertainty that exists early in design. Likewise, the accuracy of the orbit insertion, the structural and aerothermal loads, and other parameters, will be variations from the trajectory that include the uncertainties remaining when we fly and not the full set of early design uncertainties.

The parameters that are better known for a specific vehicle on flight day have a true value for the vehicle we will fly, but we don't know that value during design or prior to choosing a specific vehicle (this is an "epistemic" uncertainty in statistics, as opposed to an "aleatory" uncertainty that just randomly varies). A way to handle this is to devise multiple system models. For example, we choose a sluggish (heavy and slow) and a sporty (light and fast) launch vehicle model (Ref. 1) covering the range of the design uncertainties (and other uncertainties that are better known for a particular launch vehicle before it flies). Next we design trajectories for how these would be flown. Finally we run the Monte Carlo simulation for each using the estimated uncertainties remaining on flight day. If the Monte Carlo simulation had all uncertainties included as if they were all unknown, then a 99.865% value (with 10% CR) from this simulation would not cover the 99.865% value for the sluggish or sporty vehicle models; their success might be much lower.

As an example, for the Ares I launch vehicle, when a particular vehicle is assembled for launch, a small motor test estimates the First Stage solid rocket motor burn rate. The J-2X engine that powers the Upper Stage is ignited on a test stand prior to flight. Various vehicle elements are weighed. The axial force coefficient is better known by the time of first flight than it is in the early design phases. All these things influence how sporty or sluggish the vehicle will be. Assuming that it is necessary to meet requirements throughout the year, a sluggish vehicle model must meet the payload requirements in the coldest month. Likewise, the vehicle structure and thermal protection must be sufficient for a sporty vehicle model flying in the summer (when the solid motor burn rate is highest).

There are additional parameters that will likely be better known before flight than they are now, that primarily affect flight control, such as the pitch moment coefficient. However, since the flight control must be able to fly any combination of the various aerodynamic parameters with any combination of the vehicle models, a more robust design (that doesn't cause an explosion of the number of Monte Carlo runs that must be made) simply lumps the entire uncertainty for these types of parameters into the Monte Carlo runs for the varying vehicle and launch-month cases. More will be said about epistemic parameters and flight control later.

For Ares I, vehicle models are currently being used for heavy/slow (worst payload performance and lowest thrust/weight ratio at liftoff), light/fast (highest dynamic pressure, highest heat rate, highest acceleration, and highest liftoff thrust/weight ratio that leads to the fastest umbilical release needs), hybrid (sluggish First Stage and sporty Upper Stage, leading to highest acceleration in Upper Stage flight), a low liquid oxygen remaining model, and a low liquid hydrogen remaining model. For all models except for the low propellant remaining cases, nominal mixture ratio for the engine is assumed. This is because an off-nominal mixture ratio does not affect the trajectory design, the nominal payload performance, or the induced environments (max dynamic pressure, acceleration, etc.) to first order. The mixture ratio variation only affects the propellant remaining. So the off-nominal mixture ratio value, seen on the engine test stand, can be incorporated into this same process for the cases where the objective is low propellant remaining. Note that it is assumed here that the vehicle is assembled with components that are chosen randomly; that is, particular engines and other components are not chosen to assemble a vehicle with specific characteristics. "Cherry-picking" vehicle components from a warehouse in order to obtain desired vehicle characteristics would change the statistics.

There are three steps to the vehicle design process:

1. Determine the partial derivatives of the desired varying outputs with respect to each of the parameters that can vary but will be known for a given assembled vehicle (for example, the partial of dynamic pressure with respect to burn rate).
2. Determine the target values of each parameter, for example the target dynamic pressure, heat rate, acceleration, and liftoff thrust/weight ratio.
3. Determine the vehicle model that uses the total set of parameters and maximizes the statistical chance of ending up with that vehicle.

American Institute of Aeronautics and Astronautics

B. Partial Derivatives

The partial derivatives are derived by generating an optimized trajectory (using whatever trajectory design procedure is being used for the vehicle trajectories) with a slightly modified input of the parameter in question. For payload, the partials are the partial of payload with respect to each input parameter change (burn rate, axial force coefficient, First Stage dry mass, etc.). For remaining oxygen, the partials are the partial of oxygen remaining with respect to each input parameter change. All other parameters are held constant. The trajectory optimization in each case is typically performed to maximize payload to orbit. However, if the desired partial is, say, dynamic pressure with respect to Upper Stage mass, the mass trades one-for-one with payload with no change in dynamic pressure. The partial in this case must be done with fixed payload, instead maximizing remaining propellant.

C. Generating the Target Values for the Parameters

Once the partial derivatives are known, and assuming the 1-σ variations of the different input parameters are also known, we assume the effect of the different parameters is independent and generate the root-sum square (RSS) combination:

$$\sigma_z = \sqrt{\sum_{i=1}^{N}\left(\frac{\partial z}{\partial x_i}\sigma_i\right)^2} \tag{1}$$

where z is the desired output variable (payload, maximum acceleration, etc.), the x_i are the input parameters that we will choose, and the σ_i are the 1-σ variations of the input parameters. To get the target value of z, choose the sigma level corresponding to the desired probability level (assuming the combination of the various parameters behaves as a Gaussian). As an example, for a 99.73% high value, the sigma level would be 2.782. So the target value of z is then

$$z = n\,\sigma_z \tag{2}$$

where n is the necessary sigma level. The process may be extended to correlated parameters, but that will not be pursued here.

D. Determining the Vehicle Model

The problem to be solved here is to maximize the probabilistic likelihood of the vehicle model given the constraints z (target parameters, two in the heavy/slow model and four in the light/fast model in the example above). This assumes the number of targeted parameter values is less than the number of vehicle parameters that can be varied so that the vehicle model can generate the target z values for a range of possible input parameters. The optimization with constraints can be done numerically (see, for example, Ref. 4) by choosing candidate values for the x_i, determining the probability density for each x_i (no matter what its distribution type), multiplying these to get the joint probability density, and then numerically varying the x_i to maximize the joint probability density. Instead of multiplying the probability density functions (PDF), it is more numerically stable to take the natural logs of the PDF values and add the logs. Given the 1-σ impacts on z as shown in parentheses in equation 1, and assuming independence of the impacts from the x_i, the value of z for each set of candidate x_i and for each target parameter may be determined. Typically this will not satisfy the constraints. Satisfying the constraints is part of the numerical optimization. One way to do this is to introduce a penalty function into the objective function such that the optimization procedure will force the constraints to be approximately satisfied. The user can control how closely the constraints are satisfied by varying the penalty for each constraint.

Note that in the case of a uniform distribution, all values are equally likely. Thus if we are trying to get a worst payload case, choosing the maximum aerodynamic axial force coefficient gives the biggest payload impact from this uniformly-distributed parameter so that other (Gaussian) parameters don't have to contribute as much to the impact. So, other things being equal, uniformly distributed parameters should be chosen at their extreme values in order to maximize the overall vehicle likelihood. This assumes, though, that the results are feasible with the uniformly distributed parameter at its endpoint. If there are multiple parameter targets, it is possible that the solution requires an intermediate value for the uniformly-distributed variable.

E. Parameters Known on Flight Day

Suppose we are using this process and have a vehicle model that results from using a known engine, weighed components, and a specific value of the axial force coefficient at each flight condition (in design and for requirements verification we use the models like the heavy/slow and light/fast models to cover the range of cases). The trajectory is designed with this model in mind, and the payload can be adjusted, if desired. On flight day, the winds are normally measured to help with the go/no go decision. The temperature is also known (particularly important for solid rocket motor performance). So most of the wind variation and temperature variation becomes a known quantity and is no longer a flight day unknown. These variations can be used for the trajectory design if day-of-launch trajectory design is used. Evaluation of this set of vehicle model and environmental parameters allows for a more-informed decision as to whether the vehicle meets all its requirements at the time of launch. The flight performance reserve, that part of the propellant set aside to cover for the flight day uncertainties, can be a bit less since these variations are no longer unknown.

The remaining flight day uncertainties are randomly chosen in the Monte Carlo simulation when doing the vehicle design work for worst performance, structural loads, etc. While the vehicle is still in the design phases, the practice (for payload performance studies) is to postulate a sluggish vehicle, then to determine the expected performance for the vehicle with challenging winds and temperatures, and finally to show through the Monte Carlo simulation that the flight performance reserve is sufficient to achieve orbit. The sluggish vehicle model choice represents the manifesting likelihood (probability of being able to launch a vehicle that has been assembled). The challenging winds and temperatures represent the ability to launch in a certain percentage of natural environments. The result of the Monte Carlo simulation represents the probability of being able to take the sluggish vehicle on the challenging day and get it to orbit. The percentage of success for this last item is probably the highest, because it represents whether the vehicle, once launched, can reach orbit, whereas the earlier analyses represent cases where the choice of not launching is still available.

As mentioned earlier, aerodynamic parameters, such as pitching moment, should be better known when the vehicle flies than they are early in design. Also, there is an actual value of these parameters that likely does not change much from flight to flight. Other parameters that primarily affect flight control and have this same characteristic include the vibrational modes of the vehicle and the fuel slosh parameters. All of these are "epistemic" parameters as opposed to the truly randomly varying "aleatory" parameters. Since most of the parameters that have a significant effect on flight control are epistemic, these can be left in the Monte Carlo simulation and various combinations of these parameters are investigated for their effect on the flight control. This is important because the flight control impacts of various parameters is typically very nonlinear. It would also be advisable to remove these parameters from the Monte Carlo simulation and test them individually in order to determine which ones are highly sensitive. If parameters for which the vehicle is particularly sensitive were to end up at bad values, it is important that the vehicle design still works.

F. Summary of Procedure to Develop Models and Perform Monte Carlo Runs
1. Generate vehicle models that represent the desired level of coverage for what vehicle might be assembled. This is a "manifesting" model, that is, it determines the likelihood that it will be possible to use the randomly-chosen combination of parts for an actual launch.
2. Choose the Monte Carlo cases that will be needed. These are combinations of the vehicle models, launch months, mission destinations, and any other choices that are cases we will have to be able to launch with, in order to find the worst cases of desired end results (worst structural loads, worst fuel usage, etc.)
3. If the time of launch, within a month, can be a driver in the launch go/no go decision (for example, a low predicted propellant temperature with a headwind impacting payload capability), then run a Monte Carlo simulation where only these parameters are varied and the trajectories are designed as would be done on launch day and choose the case for the desired percentage of coverage (e.g., a cold temperature and a headwind).
4. Run the Monte Carlo simulations with the flight day unknowns randomly varying in order to determine the worst-case and probabilistic values of interest. For Ares I, there are hundreds of output values of interest.

III. Part 2: How Many Monte Carlo Runs, and How to Obtain the Required Percentage Numbers?

A. Consumer and Producer Risk
Imagine a Monte Carlo run that is used to estimate the failure rate of some component or system. To be concrete, suppose $N = 2000$ identical widgets are subjected to a stressful duty cycle, and 5 of them fail. What can be said of the failure rate of the widgets? The immediate temptations are to say that the empirical failure rate is 5/2000 or 0.25%, and to assume that this sample is typical. But if we run a second Monte Carlo simulation, a different

number might randomly fail. What confidence interval can be put on this estimate? In order to be "90% confident" that a quoted failure rate is an upper bound on the actual failure rate, what rate should be quoted?

The rigorous approach, called binomial failure analysis, assumes that there is some actual failure probability p, and that the 2000 widgets form one Monte Carlo run out of an infinite number. (To be careful about language, the individual test of each of the widgets will be called a "sample" and the collection of N samples will be called a "run".) For any given p, the probability of seeing exactly k failures in a run of size N is given by the binomial probability formula (Ref. 5)

$$P_{\mathrm{BIN}}\left(k\,|\,p,N\right) = \binom{N}{k} p^k \left(1-p\right)^{N-k} . \tag{3}$$

Given the parameters, it is easy to plot this probability. Figure 1 shows the results for three failure rates near the intuitive result. Notice that any of these actual failure rates gives nearly the same result: around $P_{\mathrm{BIN}} = 18\%$, which means that the expectation is that about 18 out of every 100 samples of 2000 will exhibit exactly 5 failures. Also notice that ks in the interval from about 2 to 8 have non-negligible probabilities P_{BIN}. Conversely, this means that it would be mildly coincidental if the actual failure rate were really 0.25% – and that if the actual failure rate were really 0.25%, one should bet against getting exactly 5 failures out of 2000 samples.
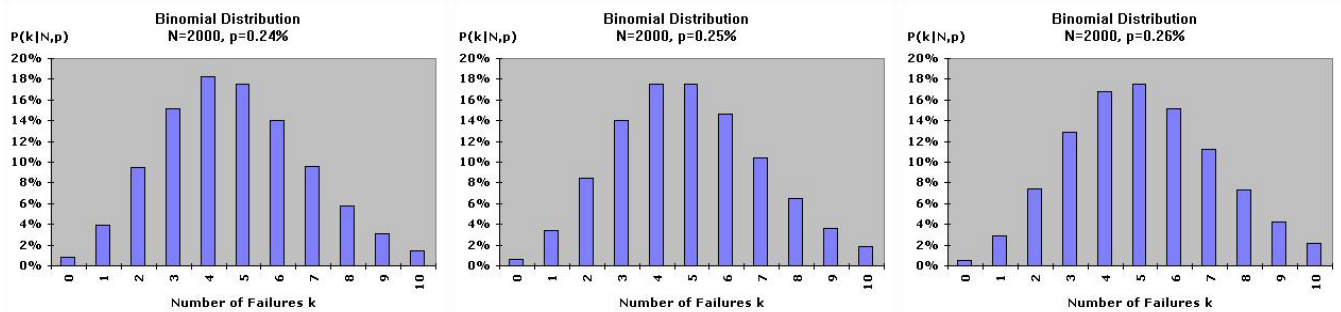

Figure 1. Probabilities $P_{\mathrm{BIN}}(k|p,N)$ for different k for three different actual failure rates.

However, the significance of the intuitive result can be easily demonstrated. Suppose the probability P_{BIN} of getting 5 failures out of 2000 samples is calculated as the actual probability is varied. The result is given in Fig. 2 – the probability P_{BIN} peaks at the intuitive value of failure probability, $\hat{p} = k/N$.
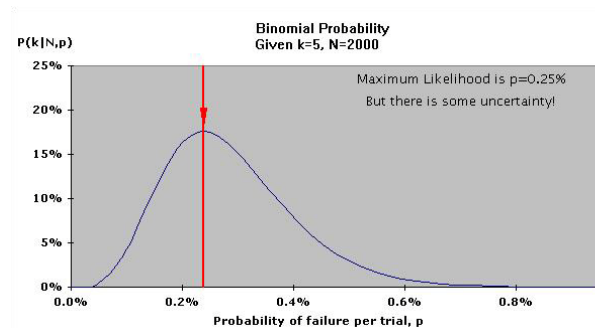

Figure 2. Probability of seeing exactly k failures as a function of actual failure rate.

Thus $\hat{p} = k/N$ is what is called a *maximum likelihood estimator* for the actual failure rate p. The wording here is important: $\hat{p}$ is not the "most likely" value of p. In fact, p has some exact unknown value, and there is not a PDF associated with different values of p. However, $\hat{p}$ is the value of p that maximizes the probability $P_{\mathrm{BIN}}\left(k\,|\,p,N\right)$ — and "maximum likelihood estimator" is the agreed terminology.

So what does this mean in terms of a confidence interval on p? There appears to be a range of actual failure rates consistent with the observed k/N. But p is *not* a random variable (although k is). If it were, its PDF could be integrated to give its cumulative distribution function (CDF), then the CDF could be inverted at 0.10 and 0.90 to get the 10%−90% confidence interval. Instead, the standard approach to binomial failure analysis is to compute, for a range of possible failure rates p, the probability of getting k *or fewer* failures, and define the confidence in terms of the resulting function of

p. For example, for *k* = 5, then for each of the three subplots in Fig. 1, imagine adding up the heights of the columns for 0,1,…5 failures. The result is the cumulative binomial probability and is a decreasing function of *p*.

The conventional way to put confidence limits on empirical binomial failures is to compute upper and lower bounds on the probability *p*, not using a cumulative distribution of the "probability of the probability", but by answering the following questions:

- What is the largest probability p_{lower} that will result in no more than a chance β_L of generating *k* or more failures out of *N* samples?
- What is the smallest probability p_{upper} that will result in no more than a chance β_U of generating *k* or fewer failures out of *N* samples?

The probabilities β_L and β_U are acceptable probabilities of error, typically taken as 5% or 10% for practical calculations. Answering these questions requires the solution of the equations (Ref. 6)

$$\sum_{j=k}^{N}\binom{N}{j}p_{lower}^{j}\left(1-p_{lower}\right)^{N-j}=\beta_L \tag{4}$$

and

$$\sum_{j=0}^{k}\binom{N}{j}p_{upper}^{j}\left(1-p_{upper}\right)^{N-j}=\beta_U \ . \tag{5}$$

Figure 3 shows an example of this upper- and lower-bound calculation. The monotonically decreasing line is the cumulative distribution function (0 to *k*) for the binomial distribution function, the left side of Eq. 5, while the increasing line is the overlap complement (*k* to *N*), the left side of Eq. 4, for the case *k* = 2, *N* = 2000. The dotted line is the 10%−90% confidence interval.
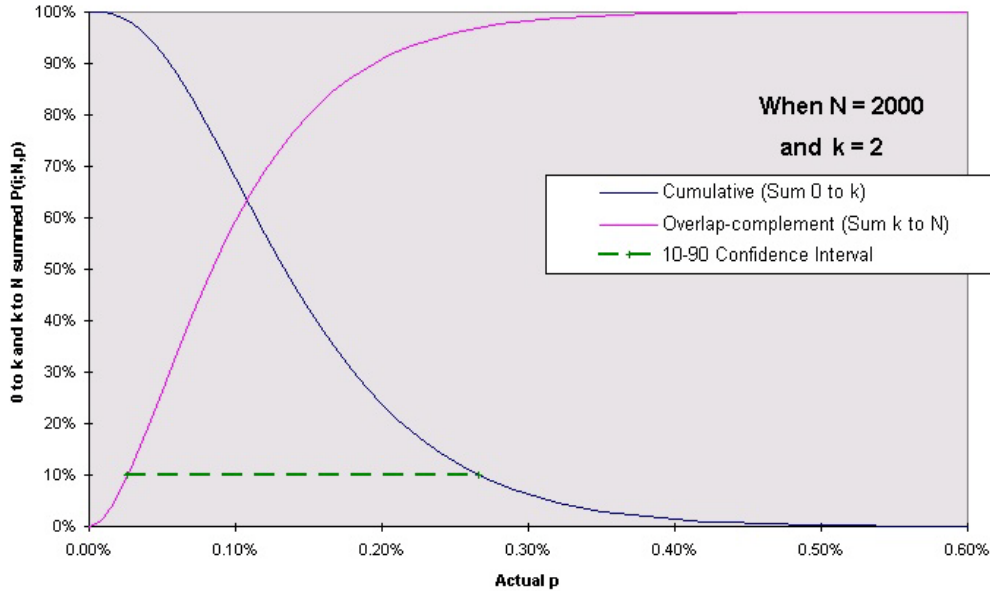

Figure 3. Confidence interval for binomial samples.

Note these expressions are related to the cumulative probability of seeing *k* or fewer failures, which is

$$F_{\text{BIN}}\left(k\mid p,N\right)=\sum_{j=0}^{k}\binom{N}{j}p^{j}\left(1-p\right)^{N-j}.\tag{6}$$

If this function of p is plotted for a given k and N, the result is the curve shown in Fig. 4. The curve is called the *operating characteristic (OC) for the sampling plan (k,N)*. The terminology comes from the discipline of quality
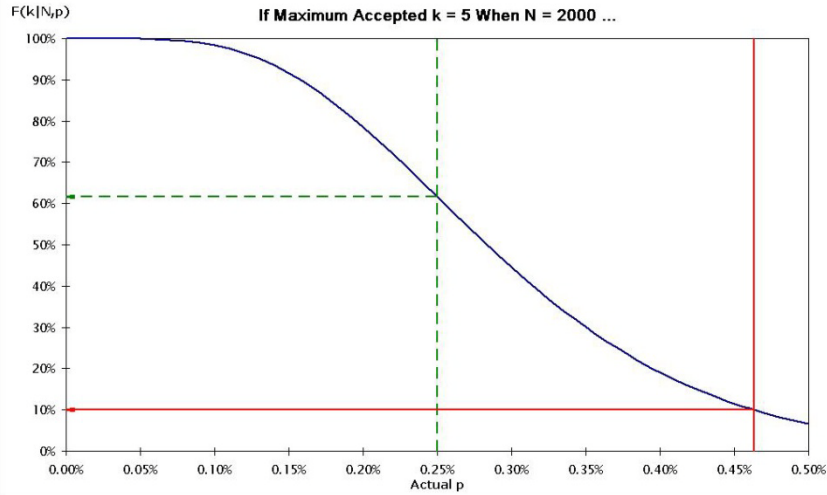
Figure 4. Cumulative probability $F(k|p,N)$ as a function of actual failure rate.

control.

Note that if the actual failure rate were 0.25%, the plot shows that about 62% of 2000 sample runs exhibit 5 or fewer failures. If there were a strict requirement for the widget failure rate not to exceed 0.25%, then a single run of 2000 samples with 5 failures would not give much confidence that the widgets meet the requirement. There is still a 58% chance of acceptance if $p = 0.26\%$, about 50% chance if $p = 0.28\%$, and so on. In fact, the probability of getting 5 or fewer failures in 2000 samples drops below 10% only when p increases to 0.463%. (Note this is equivalent to saying that $p = 0.463\%$ is the solution to the equation $F_{\text{BIN}}\left(k\mid p,N\right)=10\%$ for the given k and N.) This suggests that a (5, 2000) sampling plan is likely to be unsatisfactory to verify a requirement like $p < 0.25\%$. This risk of accepting an invalid design is called *consumer risk* (CR).

The traditional approach to choosing a sampling plan exploits the fact that two independent parameters (k and N) can be specified. This approach also reflects the fact that the producer and the consumer of the widgets have distinct concerns about the transaction. In this paradigm, the consumer has a requirement that the widgets have a specified not-to-exceed failure rate. The producer, however, would like not to waste good products because of inadequate sampling. It is in the interest of the producer therefore to specify an allowable failure rate. In order to satisfy both parties, though, the producer-allowable failure rate should be less than or equal to the consumer-specified requirement.

In this context, the CR for a given sampling plan (k,N) is the upper bound on the probability that widgets will be accepted that have greater than the consumer-required failure rate. The producer risk (PR) for a given sampling plan (k,N) is the upper bound on the probability that widgets will be rejected that have less than the producer-allowable failure rate. As tersely as possible:

- CR is the risk of accepting a bad product.
- PR is the risk of rejecting a good product.

Figure 5 shows the traditional setup of an OC that provides 10% CR and 10% PR for consumer and producer required/allowable failure rates of $p_C = 0.250\%$ and $p_P = 0.125\%$, respectively. The parameters k and N, in this case (13, 7579), are selected to be the minimum values that meet the goals of the sampling plan. Note that if the actual failure rate is less than p_P, there is a finite probability that the widget will be rejected, and if the actual failure rate is greater than p_C, there is a finite probability that the widget will be accepted.

American Institute of Aeronautics and Astronautics

However, without a separate PR requirement, the PR is the complement of the CR, PR = 1 − CR. Failure to specify or to account for PR leads to a verification framework with a very large PR – that is, there is a significant probability that the resulting design will be over-conservative because acceptable designs will be rejected. Nonetheless, sampling plans accounting only for CR can be designed.

So what is the best way to choose a sampling plan for a given consumer-required failure rate? It may be
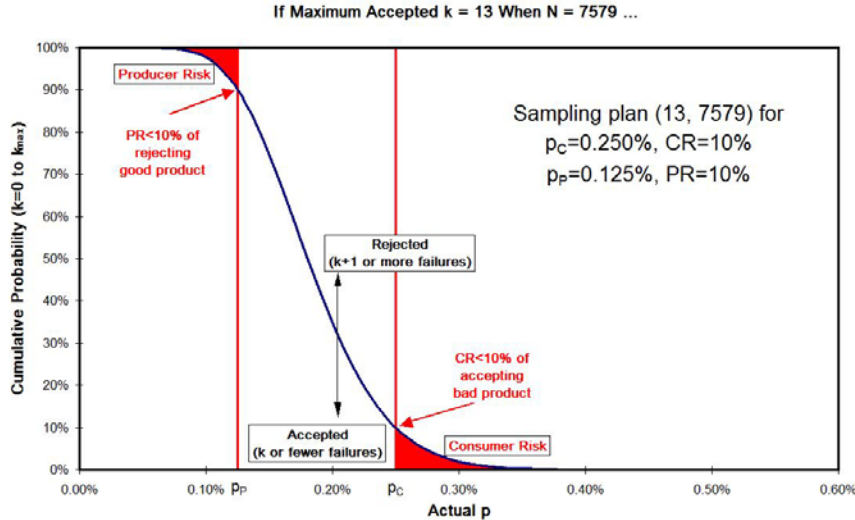
Figure 5. Example of an operating characteristic for a sampling plan devised to provide 10% Consumer Risk and 10% Producer Risk.

worthwhile to backtrack a bit and approach the problem from a slightly different direction.

The essential process involves the following steps:
1) Specify an acceptable failure rate (a not-to-exceed requirement) p_A,
2) Choose a sampling plan (k,N), in which a design is accepted if, in a run of N samples, no more than k failures are encountered, and
3) Compute the upper bound of the probability of accepting the design if the actual failure rate exceeds the requirement (i.e., if $p > p_A$).

The probability in step 3 is the CR. If the CR is unacceptable for a given (k,N), it is necessary to revise the sampling plan (go to a smaller k or a larger N).

Typically requirements are written with CR limited to 10% or less, that is, the verification of various aspects of system reliability and performance is to allow no more than 10% chance that an unsatisfactory design will be accepted because of statistical errors.

For any given acceptable failure rate p_A and number of samples N, specifying an upper bound on CR of 10% is equivalent to putting an upper bound on the allowable number of failures k. Conversely, for any given acceptable failure rate p_A and allowable number of failures k, specifying CR is equivalent to putting a lower bound on the number of samples N. Using Eq. 6, Table 1 shows some typical results for $p_A = 0.27\%$ and $p_A = 0.135\%$. Notice how, in order to capture the required CR, the allowable failure fraction k/N is much smaller than the target failure fraction (0.0027 and 0.00135).

Table 1. Combinations of number of allowable failures and required number of samples.

N	k	p_A	CR	k/N
852	0	0.27%	10%	0
1440	1	0.27%	10%	0.000694
1970	2	0.27%	10%	0.001015
2473	3	0.27%	10%	0.001213
1705	0	0.135%	10%	0
2880	1	0.135%	10%	0.000347

3941	2	0.135%	10%	0.000507
11410	10	0.135%	10%	0.000876
514	0	0.135%	50%	0
1243	1	0.135%	50%	0.000805
1981	2	0.135%	50%	0.001010

Note that the allowable number of failures is noticeably less than what one gets using an intuitive rule like $k \cong Np_A$.

Consider the operating characteristic for a sampling plan based on this table. Figure 6 is a typical result. (For this figure, $N = 1970$ has been rounded up to $N = 2000$). This figure clarifies the meaning of Type I and Type II errors (PR and CR). Designs actually meet or fail to meet the requirement if they reside to the left or to the right of the vertical line at $p = p_A = 0.27\%$. However, they are accepted or rejected according to whether the results of an "experiment" (or simulation, in the case of trajectory dispersions) lie below or above the OC. The red area to the right and below the OC represents CR, i.e., the acceptance of a design that actually fails to meet requirements (note the upper bound on CR is 10%, as promised). The red area to the left and above the OC represents the PR for this situation where there is no separate specification for the producer-allowable failure rate.
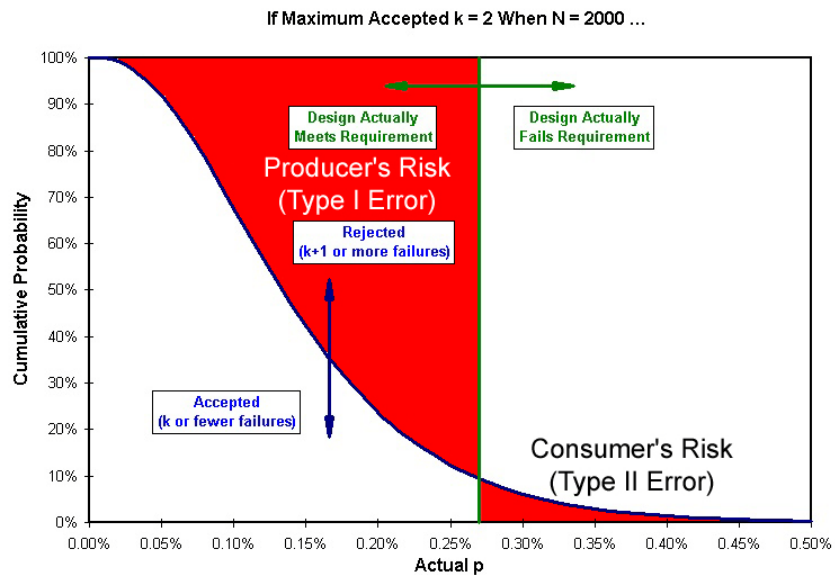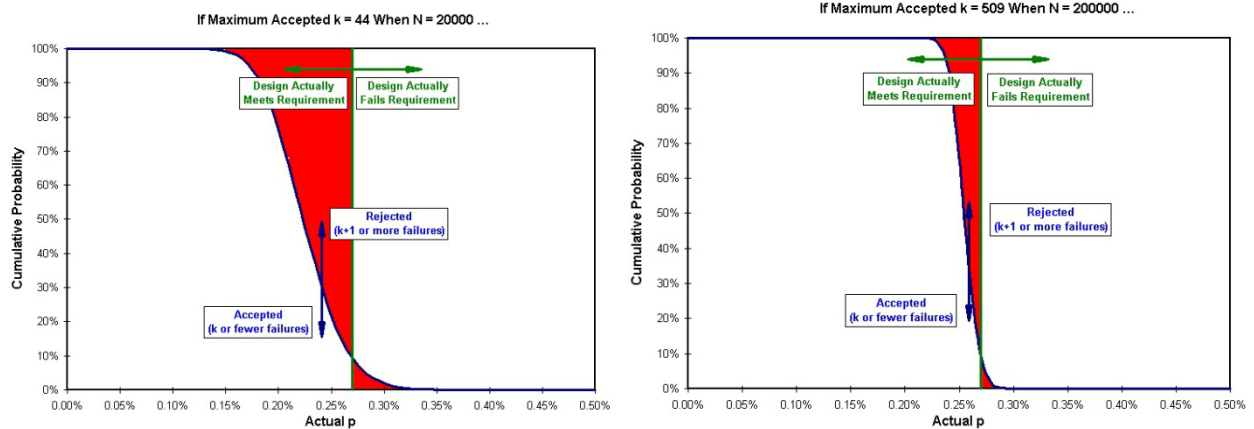

Figure 6. Operating characteristic for the (2,2000) sampling plan.

Note that PR is not specified, so any of these sampling plans has 90% PR at the specified failure rate p_A. However, the advantage of increasing N is that the PR at failure rates less than the specified failure rate becomes significantly less. Suppose, hypothetically, that the producer set a maximum allowable failure rate of $p_P = 0.20\%$ (less than the consumer specified rate $p_C = 0.27\%$). For the (2, 2000) sampling plan, the PR is around 75%. But the OC shifts as N is increased. Figure 7 shows the (44, 20000) and (509, 200000) sampling plans, for which k was selected so that CR is 10%. Thus,

- For the (2, 2000) sampling plan, the PR at 0.20% is around 75%,
- For the (44, 20000) sampling plan, the PR at 0.20% is down to about 25%,
- For the (509, 200000) sampling plan, the PR at 0.20% is essentially zero.

Note that as the run size N is increased, the test aligns more and more closely with the hypothesis.

Figure 7. Operating characteristics for the (44, 20000) and (509, 200000) sampling plans.

Specifying "producer allowables" distinct from "consumer allowables" enables the establishment of a minimum run size, but more importantly, ensures that the design is not overly conservative, that is, that acceptable designs are not tossed aside unnecessarily.

B. From Binomial Failures to Order Statistics

Trajectory dispersions are frequently used to establish extreme values for launch vehicle operating parameters (e.g., maximum dynamic pressure, maximum pitch attitude error, maximum axial acceleration, etc.). Simply using the PERCENTILE function from Excel® (or equivalent) to compute the 99.865 percentile value from a run of 2000 trajectories will usually underestimate the population "3-sigma-equivalent" value, however. More importantly, it will also involve considerable CR.

There is a close relationship between binomial failure analysis and order statistics. This enables a non-parametric estimate of population extremes, that is, an estimate that does not depend on the specifics of the underlying distribution, be it normal, log-normal, β, γ, etc. It also enables accounting for CR.

To see the connection, suppose the PDF of some continuous variable x is $g(x)$, with CDF

$$G(x) = \int_{-\infty}^{x} g(x')dx' . \tag{7}$$

Consider a run of N independent samples taken from $g(x)$. Order the run from least to greatest x value, numbering them 1,2…N. Then the mth order statistic is defined to be the mth smallest value in the list, and is a random variable with a computable PDF and CDF. (Note that $m = 1$ corresponds to the minimum, $m = N$ corresponds to the maximum, and, if N is even, $m = N/2$ is the median.) The mth order statistic is denoted $x_{(m)}$. The CDF for $x_{(m)}$ is given by (Ref. 7)

$$F_{(m)}(x \mid N) = \sum_{j=m}^{N} \binom{N}{j} \left[G(x) \right]^{j} \left[1 - G(x) \right]^{N-j} . \tag{8}$$

From the binomial theorem

$$\sum_{j=0}^{N} \binom{N}{j} \left[G(x) \right]^{j} \left[1 - G(x) \right]^{N-j} = 1 \Rightarrow \sum_{j=m}^{N} \binom{N}{j} \left[G(x) \right]^{j} \left[1 - G(x) \right]^{N-j} = 1 - \sum_{j=0}^{m-1} \binom{N}{j} \left[G(x) \right]^{j} \left[1 - G(x) \right]^{N-j} . \tag{9}$$

The last summation is simply the cumulative binomial probability for $p = G(x)$, $k = m - 1$, hence

$$F_{(m)}(x \mid N) = 1 - F_{\text{BIN}} \left((m-1) \mid G(x), N \right) . \tag{10}$$

Because of the symmetries of the binomial formula, this is equivalent to

$$F_{(m)}(x \mid N) = F_{\text{BIN}}\left((N-m)\middle|(1-G(x)), N\right),$$ (11)

which is more computationally friendly than the other form when N is large and $N - m$ is small.[3]

For reference, the PDF for the mth order statistic is the derivative of the CDF[4] and is given by

$$f_{(m)}(x \mid N) = N\binom{N-1}{m-1}\left[G(x)\right]^{m-1}\left[1-G(x)\right]^{N-m} g(x).$$ (12)

Estimators can be constructed for any desired level of CR and success rate. The CDF for the mth order statistic can be inverted to establish confidence bounds. But first, a discussion of the non-parametric nature of order statistics is called for. So far, the examples shown here have been tied to a standard Gaussian random variable x. However, it should be noted that the CDF $F_{(m)}$ for the mth order statistic is a function of the CDF $G(x)$ of the underlying distribution, and is not tied to x directly. This means that order statistics are "non-parametric", that is, they do not depend on the parameters or type of the underlying distribution. This is because the statistics are tied to the binomial calculations of success and failure, not to the distribution that leads to the successes and failures. Figure 8 shows a CDF and illustrates the CR as a function of probability level desired if we examine point 1997 out of 2000 samples, irrespective of the underlying probability distribution. The resulting curve is valid for any underlying probability density.

From this point, order statistics will be discussed in terms of the CDF $G(x)$, which makes the discussion independent of the type of the underlying probability density. It is desired to choose order statistics that satisfy a given benchmark success rate, such as $G(x) = 99.865\%$, while maintaining an upper bound on CR. That is, it is necessary to choose N and m that satisfy the inequality

$$F_{(m)} = F_{\text{BIN}}\left((N-m)\middle|(1-99.865\%), N\right) < 10\%.$$

However, these combinations are already provided in the calculations leading to tables equivalent to Table 1, e.g., $F_{\text{BIN}}(k \mid 0.135\%, N) < 10\%$.

---

[3] Excel® has a convenient worksheet function, BINOMDIST. This expression is equivalent to '=BINOMDIST(N-m,N,1-G,TRUE)' where N, m, and G refer to the appropriate cells. However, large N or large $N - m$ can produce #NUM! errors.

[4] The sum "telescopes" (i.e., successive terms) cancel, leaving only the first.
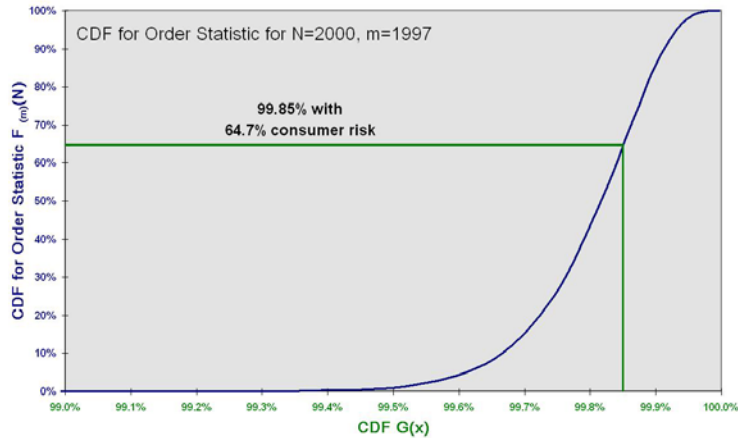
Figure 8. The CDF for an order statistic does not depend directly on the underlying distribution, only on the cumulative $G(x)$.

The equivalent order statistic is thus $m = N - k$. For example, the tables for $p_A = 0.135\%$ show that the sampling plans $(k, N) = (0, 1705)$, $(1, 2880)$, $(2, 3941)$, and so on, all satisfy the CR = 10% constraint. Figure 9 shows the CDFs for the first two of these cases, illustrating how they provide 90% confidence that the 99.865% level of performance will be estimated.
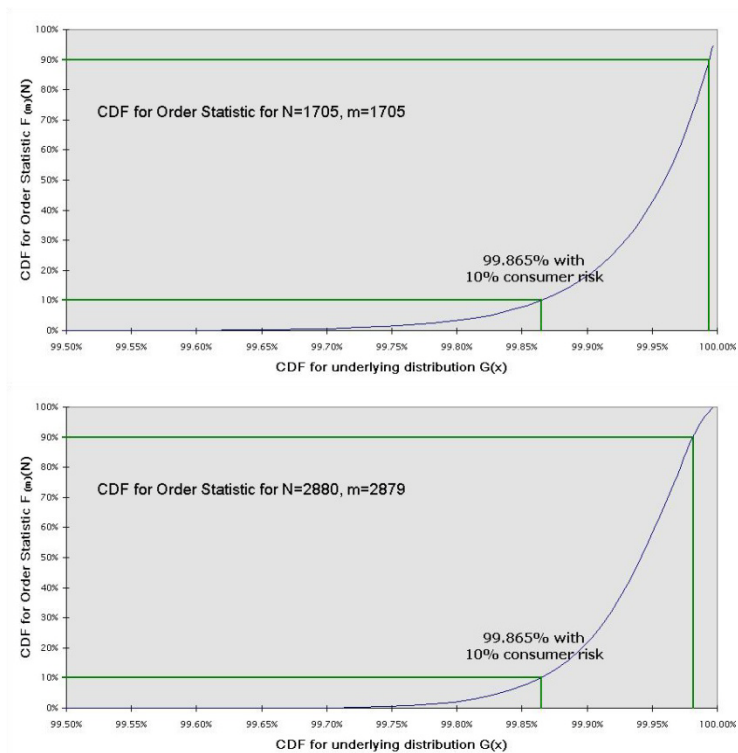
Figure 9. The CDFs for the $(m,N) = (1705, 1705)$ and $(2879, 2880)$ order statistics, showing that they both provide CR = 10% for 99.865% success.

It is possible to define PR for order statistics in several ways. The approach that most closely resembles the approach of sampling theory for quality control is to specify a producer-acceptable success rate (greater than the consumer-acceptable rate 99.865%) and to compute the probability that the order statistic will overpredict that level.

A similar approach is to compute the success rate for which there is a 10% chance that the order statistic will overpredict, that is, the value of $G(x)$ at which the CDF $F_{(m)}$ is equal to 90%. In Fig. 9, these levels are illustrated along with the 10% CR levels. These levels are 99.994% and 99.981% for the $N = 1705$ and $N = 2880$ sampling plans, respectively. Another way of saying this is that the PR for the (1705, 1705) plan is 10% if the acceptable success rate is 99.994% (about "3.85 sigma equivalent").

Finally, the sampling plans in tables like Table 1 are spaced closely enough that interpolation between entries is feasible. A Monte Carlo simulation might produce a run of $N = 2000$ Monte Carlo samples, and need a 10% CR estimator for some required success probability. Figure 10 illustrates the philosophy behind the interpolation. The figure shows 10% CR contours versus N for constant $k = N - m$ values. For every N and k, the 10% CR point is computed, using the relations developed above. Thus, the vertical axis is the 10% CR value from the CDF $G(x)$ for the underlying distribution (the quantity on the horizontal axis of Figs. 8 and 9). The horizontal lines are the benchmark levels 99.73% and 99.865% success, corresponding to two-tailed and one-tailed "3 sigma" levels for a standard Gaussian. Values from computed binomial failure tables are marked on the plot, e.g., $(k,N) = (1,440)$.

The interpolation problem can be posed as follows: suppose a run of $N = 2000$ samples is generated. It is desired to use high-end order statistics to estimate the 99.865% success value of a parameter, with no more than 10% CR. This is ostensibly the point "X" in Fig. 10, which lies on the line "AB" at $N = 2000$ connecting the $k = 0$ and $k = 1$ contours. How should the $m = 1999$ and $m = 2000$ data points (corresponding to $N = 2000$ and $k = 1, 0$), denoted x_{1999} and x_{2000}, be used to formulate this estimate? The answer is that BXC and AXD are almost triangles, and are nearly similar, which means that

$$\frac{AX}{AB} \cong \frac{DX}{DC} = \frac{2880 - 2000}{2880 - 1705} = 0.749 \,.$$

Thus one can use

$$\hat{x}_{99.865\%} \cong x_{1999} + 0.749 \left(x_{2000} - x_{1999} \right)$$

as an estimator for the 99.865% success value of the parameter x.[5]

Similar interpolations can be done for other success values (e.g., one can interpolate between $(k, N) = (2, 1970)$ and $(3, 2473)$), that is, between x_{1997} and x_{1998}, for the 99.73% success value for $N = 2000$.

---

[5] In this case, explicit computation of the ordinates at points A and B shows that there is less than 0.05% error in the coefficient 0.749 associated with the similar-triangle approximation.
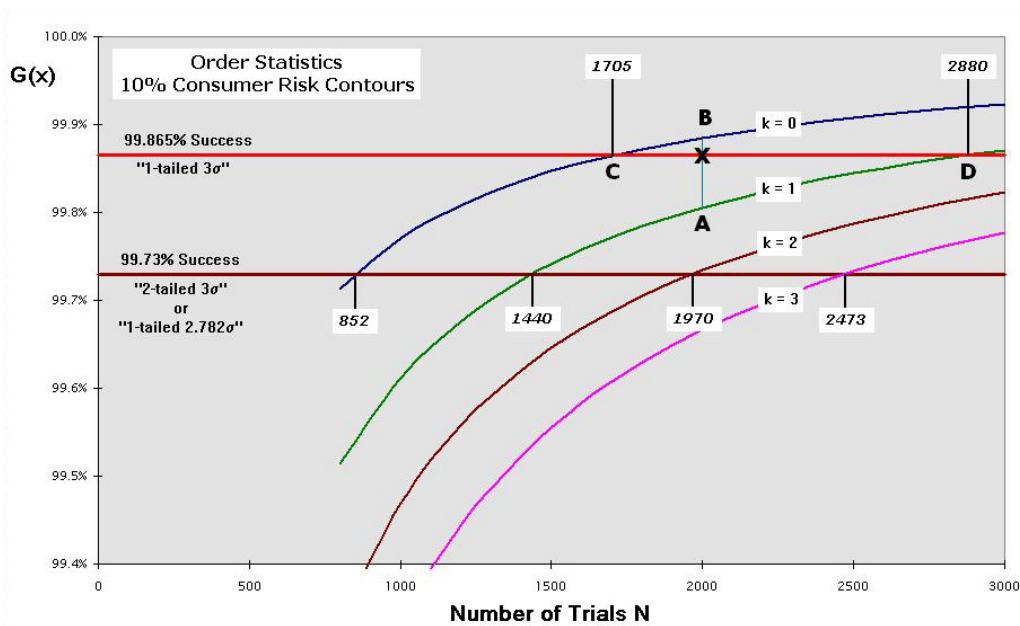
Figure 10.   10% Consumer Risk contours for various sampling plans.

IV.   Conclusions

This paper covers two specific aspects of performing Monte Carlo simulation for launch vehicle design and requirements verification. The first is how to handle different types of uncertainties: those which will become known when a particular vehicle is assembled, those that will become known on flight day, and those that will remain unknown on flight day. The method involves defining challenging vehicle models that must meet the requirements, then defining challenging conditions for when they may need to launch, and then running the Monte Carlo simulations for remaining flight day uncertainties.

The second topic discussed in this paper involves determining how many Monte Carlo samples are necessary when trying to meet a required value of success along with a required confidence level. If results of each Monte Carlo sample are viewed as being successes or failures, then order statistics provides a method of determining how many samples are needed and how many may be allowed to fail. This procedure can also be used to determine appropriate percentile settings for various parameters of interest in performing the vehicle design.

The techniques developed in this paper should be applicable to other complex system designs.

References

[1]John M. Hanson and Charles E. Hall, "Learning About Ares I from Monte Carlo Simulation," AIAA-2008-6622, *AIAA Guidance, Navigation, and Control Conference*, Honolulu, HI, August 19, 2008.

[2]H.A. David and H.N. Nagaraja, *Order Statistics*, 3rd Ed., Wiley-Interscience, 2003.

[3]Stuart Coles, *An Introduction to Statistical Modeling of Extreme Values*, Springer-Verlag, London, 2001.

[4]Dimitri P. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*, Athena Scientific, 1996.

[5]William Feller, *An Introduction to Probability Theory and Its Applications*, Volume 1, Third Edition, J. Wiley & Sons, New York, Chichester, Brisbane, Toronto, Singapore, 1968, p. 148.

[6]A. Birolini, *Reliability Engineering: Theory and Practice*, Fourth Edition, SpringeVerlag, Berlin Heidelberg, 2004.

[7]This is a basic result from order statistics; see e.g., the notes by Dr. Dennis Walsh, URL: http://frank.mtsu.edu/~dwalsh/4370/437ORDER.pdf [cited May 7, 2010].

American Institute of Aeronautics and Astronautics