Application of Statistically Derived CPAS Parachute Parameters

Leah M. Romero[1] and Eric S. Ray[2]
Jacobs Engineering, Houston, TX, 77058

The Capsule Parachute Assembly System (CPAS) Analysis Team is responsible for determining parachute inflation parameters and dispersions that are ultimately used in verifying system requirements. A model memo is internally released semi-annually documenting parachute inflation and other key parameters reconstructed from flight test data. Dispersion probability distributions published in previous versions of the model memo were uniform because insufficient data were available for determination of statistical based distributions. Uniform distributions do not accurately represent the expected distributions since extreme parameter values are just as likely to occur as the nominal value. CPAS has taken incremental steps to move away from uniform distributions. Model Memo version 9 (MMv9) made the first use of non-uniform dispersions, but only for the reefing cutter timing, for which a large number of samples was available. In order to maximize the utility of the available flight test data, clusters of parachutes were reconstructed individually starting with Model Memo version 10. This allowed for statistical assessment for steady-state drag area (C_DS) and parachute inflation parameters such as the canopy fill distance (n), profile shape exponent (expopen), over-inflation factor (C_k), and ramp-down time (t_k) distributions. Built-in MATLAB distributions were applied to the histograms, and parameters such as scale (σ) and location (μ) were output. Engineering judgment was used to determine the "best fit" distribution based on the test data. Results include normal, log normal, and uniform (where available data remains insufficient) fits of nominal and failure (loss of parachute and skipped stage) cases for all CPAS parachutes. This paper discusses the uniform methodology that was previously used, the process and results of the statistical assessment, how the dispersions were incorporated into Monte Carlo analyses, and the application of the distributions in trajectory benchmark testing assessments with parachute inflation parameters, drag area, and reefing cutter timing used by CPAS.

Nomenclature

C_D	=	Drag coefficient
C_DS	=	Drag area
C_k	=	Over-Inflation Factor
CPAS	=	Capsule Parachute Assembly System
DSS	=	Decelerator Systems Simulation
EDU	=	Engineering Development Unit
EF	=	Engineering factor
ε	=	Reefing ratio
expopen	=	Opening profile shape exponent: < 1 concave down; = 1 linear; > 1 concave up
FAST	=	Flight Analysis and Simulation Tool
FBCP(s)	=	Forward Bay Cover Parachute(s)
GEV	=	Generalized Extreme Value distribution
Logn	=	Logarithmic normal distribution
MDL	=	Multi-Dimensional Limits
MMv	=	Model Memo version
MPCV	=	Multi-Purpose Crew Vehicle (Orion)

[1] Analysis Engineer, Aeroscience and Flight Dynamics, 2224 Bay Area Blvd, Houston, TX, AIAA Member.
[2] Analysis Engineer, Aeroscience and Flight Dynamics, 2224 Bay Area Blvd, Houston, TX, AIAA Member.

μ, mu = Mean or expected value (general)
n = Canopy fill distance, normalized to reference diameter (fill constant)
n_p = Distance (measured in reference diameters) to peak drag area (infinite mass only)
σ, sigma = Standard deviation (general)
S_o = Parachute canopy full open reference area based on constructed shape including vents and slots
t_k = Time to ramp down after stage over-inflation
T_{Max} = Maximum design dispersion
T_{Min} = Minimum design dispersion

I. Introduction

THE Capsule Parachute Assembly System (CPAS) is responsible for slowing the descent rate of the Orion capsule to safely land after re-entering the earth's atmosphere. CPAS utilizes four different parachute types: Forward Bay Cover Parachutes (FBCPs), Drogues, Pilots, and Mains, deploying in the sequence shown in Figure 1.[1]

The performance of the parachute system will be verified by analysis so it is imperative that models accurately represent the parachute dynamics.[2] The CPAS Analysis team uses drop test data to reconstruct the parachute system performance in simulations. Results are documented in a model memo released semi-annually. Since sufficient testing for statistical analysis is not practical with current cost and schedule constraints, dispersions are applied to the parameters to account for uncertainties in instrumentation, modeling limitations, and engineering judgment. In previous versions of the model memo, the inflation and drag area parameters were dispersed uniformly, which does not accurately represent the expected distributions and likely results in overly conservative pre-flight predictions. The November 2012 release of the model memo, Model Memo version 111 (MMv11), published nominal and dispersed parameter values that were statistically derived from test data. Table 1 shows the progression of dispersion data from MMv8 to MMv11. Changes included the method of reconstruction from composite to independent parachutes, dispersions from uniform to statistical distributions, description of drag

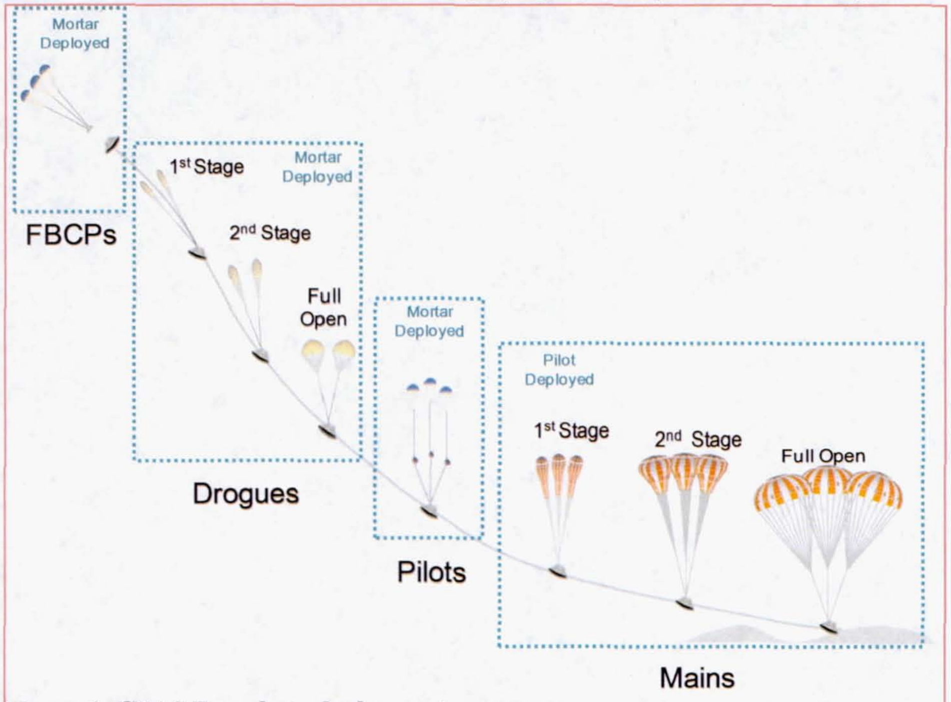


Figure 1: CPAS Parachute deployment sequence.

Table 1: Progression of Model Memo dispersion data.

MMv	Reconstruction Type	Parachute Parameter Dispersions	C_D or C_DS	Reefing Cutter Dispersions
8	Composite	Uniform	C_D only	None
9	Limited Independent for Skipped Stages Only			Statistical - GEV
9A			Both	
10*		Statistical	C_DS only	
11				Statistical - normal

*Internal release only

from reconstructing and dispersing the drag coefficient (C_D) only to the drag area (C_DS), and finally from a uniform rule of thumb to a descriptive method of statistically dispersing the reefing cutter timing.

A. Parachute Parameters

There are five parameters that describe parachute performance and directly affect the drag area curve: canopy fill constant (n), profile shape exponent (expopen), over-inflation factor (C_k), ramp down time (t_k), and drag area (C_DS). For a detailed explanation and equations of each parameter, see Ref. 3. All five parameters are necessary to describe infinite mass inflations in which the system deceleration is negligible, such as with small parachutes like the CPAS FBCPs, Drogues, and Pilots. Only a subset of the parameters (n, expopen, and C_DS) is necessary for the larger CPAS Main parachutes, which are finite mass inflations where the system deceleration is significant. The inflation parameters are determined through drop test reconstructions. Each stage of each parachute type has a different set of inflation parameters.

B. Reefing Cutter Time

The Drogues and Mains each have three stages which are controlled by timed pyrotechnic reefing cutters. The cutters for the Drogues are 14 and 28 seconds and the Mains are eight and 16 seconds. Each parachute canopy has redundant cutters for each stage. There are a total of 20 cutters on each nominal test (where stages are not skipped or removed). It is important to understand the potential variation in the actual cutter firing times, as early or delayed cuts could cause parachutes in a cluster to lead or lag their neighbors, possibly resulting in excessive loading or even parachute failure. Cutter tolerance values were not provided by the vendor and therefore must be determined from drop test data.

II. Model Memo 9 and Previous: Traditional Methods

A. Composite to Multi-Parachute Reconstructions

Prior to Model Memo version 9 (MMv9), the inflation and drag area parameters were reconstructed and simulated as composite parachutes, meaning parachutes in a cluster are treated as if they were a single parachute. Modeling the inflations and disreefs in this manner neglected cluster effects such as lead-lag, which are evident in flight tests. Though a multi-parachute reconstruction technique was implemented about the same time as MMv9 was released, the memo included a mix of composite and multi-chute parameters. The drag area curves for infinite mass skipped stage cases were too complex to be accurately modeled with a composite simulation; therefore they were reconstructed as individual parachutes. The assumption that most users of the data had a composite simulation drove the desire to publish the inflation parameters as such. For those users who had an independent parachute simulation, MMv9 instructed the use of simultaneous reefing cut times between parachutes in a cluster for congruity across simulations. The Decelerator Systems Simulation (DSS) has the ability to model individual parachutes, but it is unable to output individual loads traces. This resulted in the need for continued use of composite data. As the CPAS community continued to use the MMv9 composite parameters, progress on development of multi-parachute reconstructions increased the number of data points per test, allowing parameters to be statistically derived.

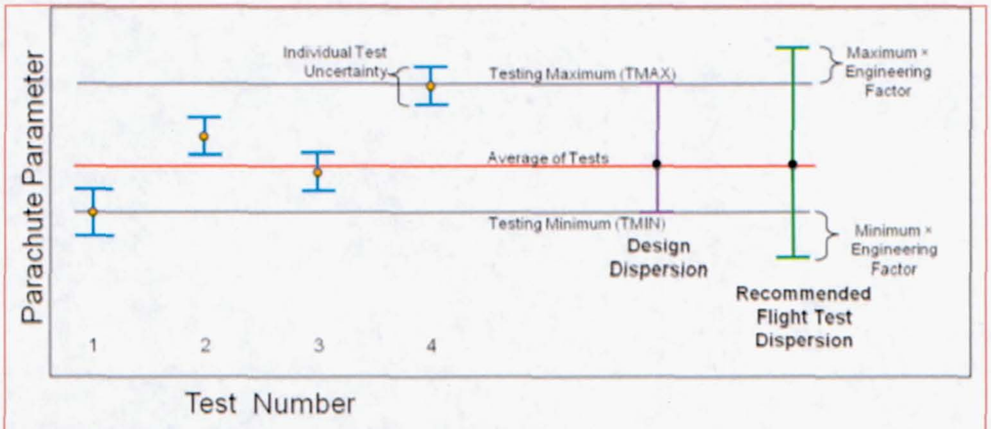

Parachute Parameter
Individual Test Uncertainty
Testing Maximum (TMAX)
Average of Tests
Testing Minimum (TMIN)
Design Dispersion
Maximum × Engineering Factor
Minimum × Engineering Factor
Recommended Flight Test Dispersion
1
2
3
4
Test Number

Figure 2: MMv9A Design and flight test dispersions.

B. Uniform Inflation Parameters

MMv9A and previous versions were limited by the smaller number of cluster tests earlier in the program; as a result, two different sets of parameters were published: design and flight test dispersions. Design dispersions were based on the range of values reconstructed from drop test data as shown as purple in Figure 2. These dispersions were bounded by the highest and lowest data points reconstructed from drop tests. The recommended flight test dispersions (green bar) were the design dispersions with an engineering factor of ±10% for inflation parameters applied to each limit. This engineering factor, EF, is applied to the minimum design dispersion, T_{Min}, and the maximum design dispersion, T_{Max}, to account for extreme cases that may be seen on future tests. The EF was based on the judgment of engineers with significant reconstruction experience.

Test preflight predictions used the flight test dispersions to bound previous test experience, and also account for test measurement, modeling, and subjective reconstruction uncertainties. Design dispersions were used in CPAS benchmark test cases to assess the system requirements against the latest model memo release. It was expected that as CPAS flight test experience grows, the spread in test data would approach the flight test dispersion values defined early in the test program.For cases where only a single data point was available, an EF of ±10% of the nominal value was used. Therefore, flight test and design dispersion were identical.

C. Transition from C_D to C_DS

An update to the technique for dispersing drag area necessitated the MMv9A revision. In the preceding model memos, the drag coefficient (C_D) was uniformly dispersed ±5% using the same method as described in the preceding section. The reefing ratio (ε) was also dispersed by ±10% in the dispersion calculation for reefed stages. The equation used, $C_D \cdot S_o \cdot \varepsilon = (C_DS)_R$, results in the triangular distribution shown in Figure 3a. However, the full open drag area distribution was determined by multiplying the uniformly dispersed drag coefficient by a reefing ratio always exactly unity (undispersed), resulting in a uniform distribution. This incongruity between the reefed drag performance and the full open performance was determined to be unacceptable. A more consistent approach is to characterize the drag performance of both types of stages (reefed and full open) in terms of drag area, using $(C_DS)_R / C_D / S_o = \varepsilon$. The effective reefing is determined from test data.

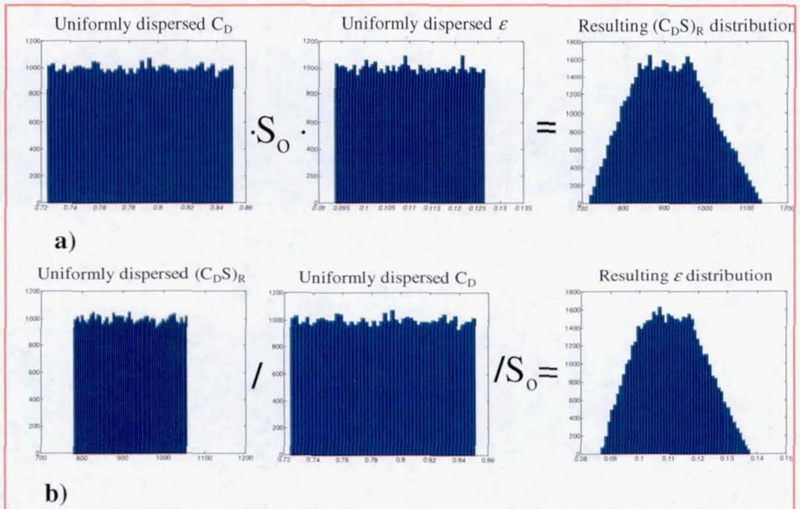


Figure 3: Reefed drag area dispersion distribution a) traditional for reefed stages only, and b) updated, to be applied to both reefed and full open stages.

Since DSS (the simulation used at the time) accepted only drag coefficient and reefing ratio as inputs, not a drag area, both C_D and C_DS were published and the analyst was required to pre-compute the reefing ratio by dividing the C_DS by the product of C_D and the reference area (S_o) (Figure 3b).

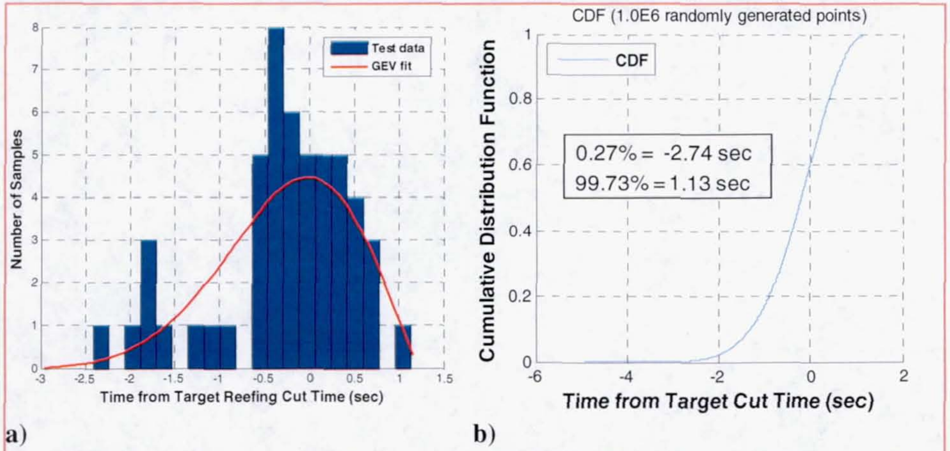


Figure 4: Reefing cutter a) histogram, best fit GEV distribution, and b) associated CDF.

D. First Statistical Assessment of Reefing Cutters

Prior to release of MMv9, reefing cutter timing dispersions were not included in the Model Memo documentation. The rule of thumb was to disperse the design cut time by ±10%.[4] This caused longer nominal times to have proportionally larger dispersions without a physical basis; thereby warranting a more in-depth understanding of flight reefing cutter dispersions.

Actual reefing cutter times were evaluated using CPAS test video timelines. Each cutter time was computed as the difference in time between the apparent cut event and the skirt exposure for the given canopy. There is some error in this approach based on the camera frame rate and subjectivity of when the events occur. Although redundant cutters are used for each disreef, only the earliest cut is visible (determined by when the parachute skirt begins to expand).

A total of 51 cut events were examined from CPAS testing for nominal cut times of 8, 12, 14, 15, 16, 20, and 28 seconds. The deviation from each nominal was calculated. A histogram of the offsets was constructed from these data to determine expected reefing cutter statistics, as shown in **Error! Reference source not found.**a. The data have a negative skew, with a longer tail toward shorter cut times. This is because for any given cut event, only the first of the two redundant reefing cutter needs to be considered, so data on most longer cutters are not gathered. A Generalized Extreme Value (GEV) distribution, shown in red, fit the data best. Several candidate methods were evaluated to determine a suitable probability distribution function. For example, a Gaussian fit failed a Kolmogorov-Smirnov (K-S) test, but the GEV fit passed.

The cumulative distribution function based on the GEV parameters is shown in **Error! Reference source not found.**b. Because this is not a Gaussian distribution, the standard deviation (σ) is not relevant. Therefore, instead of $\pm 3\sigma$, the upper and lower values of equivalent probability (0.27% and 99.73%) are displayed.

The expected and actual cutter times are shown in Figure 5. The deviation from the expected cut times can be seen by the vertical offsets from the target line. It can be seen that the test data are well bounded by the parallel lines based of -2.74 and +1.13 seconds. Absolute values were chosen over a percentage of the nominal cutter time because the latter's bounds would expand with time resulting in the same issue encountered with the legacy ±10% dispersion.

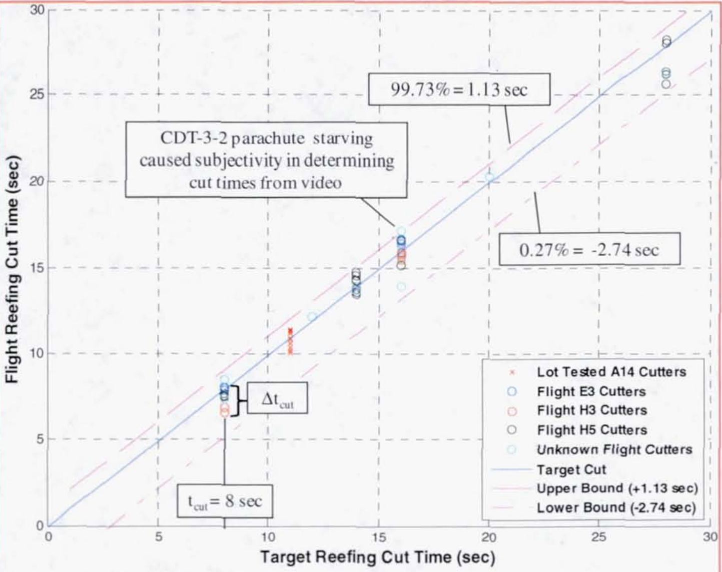


Figure 5: Target vs. flight actual cut time.

E. MMv9 Monte Carlo Assessment

After a model memo is released, CPAS benchmark testing and analysis are completed to examine the effects of the new dispersions and any model updates. There are eight different benchmark cases including a nominal configuration, parachute failures, and skipped stages. The MMv9 assessment was completed by dispersing only the parachute inflation and drag area parameters. The benchmarks did not show a need to modify the dispersions.

Implementing the uniform distributions for benchmark assessment was a simple process in all simulations. The user would refer to the latest model memo for the nominal, and upper and lower bounds of each parameter, depending on if they wished to use flight test or design dispersions. They would then create a set of dispersions unique to the simulation and, if desired, case type (one or two Drogues, two or three Mains, etc.). This caused some difficulties in simulation comparisons because the inputs were not identical.

III. Model Memo v10: Multi-Parachute Modeling

Model Memo version 10 (MMv10) was internally released in a short form only in August 2012[5]. It included statistical parachute parameter dispersions and a re-parameterization of the fill constant, but did not contain an update to the reefing cutter dispersion. Accompanying the memo was a dispersion rules spreadsheet which described how to disperse the parachutes and a set of text files (case-type dependent) of 3000 dispersed parameter values.

The primary purpose of MMv10 was to identify and resolve potential issues the new dispersions or with the distributions themselves. It also was the first memo to be assessed with benchmarks conducted using the Flight Analysis and Simulation Tool (FAST), which is to eventually replace DSS as the primary CPAS analysis simulation.

FAST has multi-body simulation capability, contains the high-fidelity parachute model that was developed in the Lockheed Martin Osiris simulation, and is capable of modeling parachutes independently.

A. Inflation Parameter Histograms

As previously stated, individual parachute test data were reconstructed for MMv10, resulting in significantly more data than when clusters were reconstructed as a single composite parachute. The reconstructed test data were plotted as histograms and built-in MATLAB functions were used to fit distribution curves to the data. Plotting test data histograms is subjective because the quantity of bins in each histogram could be varied so that all the data appeared to have a uniform distribution. For many of the parameters, the MATLAB default of 20 bins was used. Once the test data was plotted, distribution curves were fit, introducing more subjectivity.

As **Error! Reference source not found.** shows, a few different distributions could appropriately describe the test data. However, the uniform distribution does not accurately match the potential tails of this data, and the normal has a tail that includes negative values which is not physically possible for this parameter, C_DS. That leaves the Generalized Extreme Value (GEV) and Logarithmic Normal (logn) distributions as potential choices. Random numbers were generated based on both curves, which showed that the GEV distribution had a longer right hand tail causing unrealistic parachute parameters. This method was employed for each of the parameters of each parachute and stage, resulting in 38 different distributions.

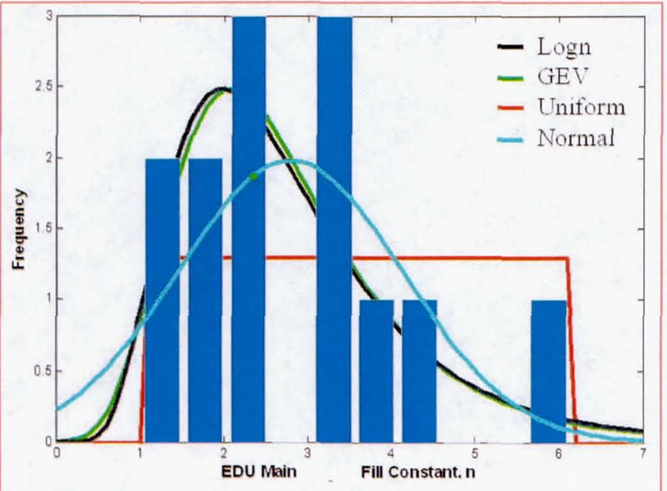


Figure 6: Subjectivity of a distribution fit.

B. Re-parameterization of Fill Constant

During assessment of the distributions curves, theoretical drag area growth curves were generated providing a rapid evaluation of effects of potential distributions on characteristics such as fill time and peak drag areas. These curves showed that a few cases lagged significantly behind the majority as shown in Figure 7. Upon further examination, the value of the fill constant, n, was found to be the cause. For infinite mass inflations (e.g., Main parachutes), the peak drag area is based on a combination of parameters, some of which are coupled. Dispersing these parameters independently can cause the timing of the peak load to be unrealistic. This was overcome by re-parameterizing[6] n to a new peak fill constant (n_p) during reconstruction, and then converting n_p back to an n based on the other inflation parameters. Note that the converted n is not the same as an n determined through a reconstruction that uses the MMv9 process. The converted n eliminated many of the unrealistic inflation cases. For

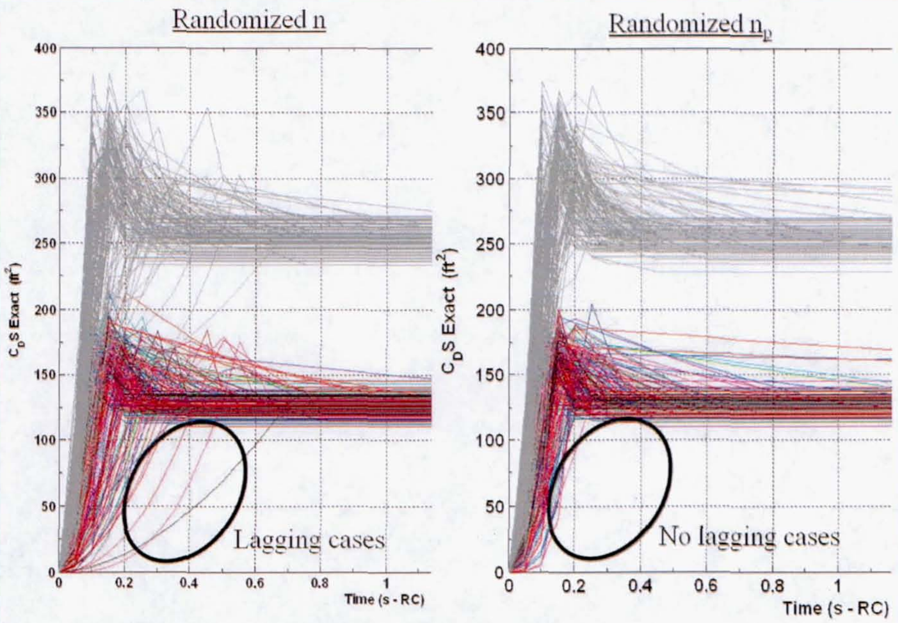


Figure 7: Effect of n re-parameterization on C_DS.

a more detailed discussion refer to Ref. 6.

C. Dispersion Implementation

As stated previously, implementing the uniform distributions in various simulations was straightforward, but tended to result in slightly different dispersed values. To eliminate this inconsistency between simulations, MMv10 and subsequent memos include a set of pre-dispersed parachute parameter values, one for each parachute in a cluster.

Dispersions are created through feeding MATLAB-created distribution rules (mu and sigma) into a Python script. First, MATLAB outputs a .csv file that includes flags for items such as number of parachutes in the cluster, whether the parachute is skipping a stage or is a "lagger", and to which stage the distribution applies (Table 2).

Table 2: MATLAB output of distribution rules

MMv10 Drogues							
CLUSTER	SKIP	STAGE	Parameter	Dist. Type	Nominal	Param(1)	Param(2)
0	0	1	CdSR	normal	115.7	115.68	5.2357
0	0	2	CdSR	normal	168.7	168.69	5.2357
0	0	3	CdSR	normal	252.1	251.34	10.51
0	1	1	CdSR	normal	168.7	168.69	5.2357
0	1	2	CdSR	normal	168.7	168.69	5.2357
0	1	3	CdSR	normal	252.1	251.34	10.51
0	2	1	CdSR	normal	115.7	115.68	5.2357

Second, the Python script uses a random number generator with a different seed for each parachute to create the dispersions. For a nominal cluster, each parachute uses the same distribution, but since a different seed is used to generate the dispersions, the resulting values are different. This reflects how parachutes perform in flight. Since n is no longer directly reconstructed, Python first disperses n_p according to its distribution and then uses algebra to convert each n_p to an n value that the simulations can use.

Third, scatter plots and histograms (Figure 8) are used to verify that the generated dispersions (blue) fall within the distributions (black line) and match test data (green bars). Finally, the script saves the dispersed values as a text file (Figure 9). The text files are used to distribute the data to interested parties. The files include all necessary parameters for each stage of that parachute.

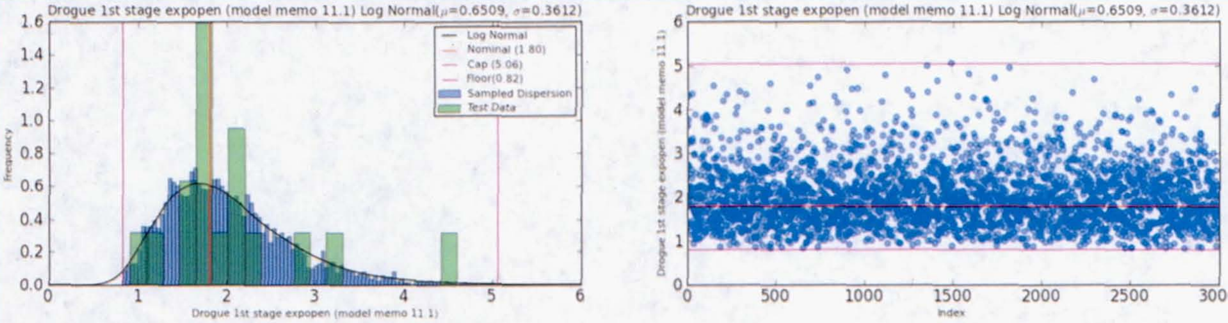
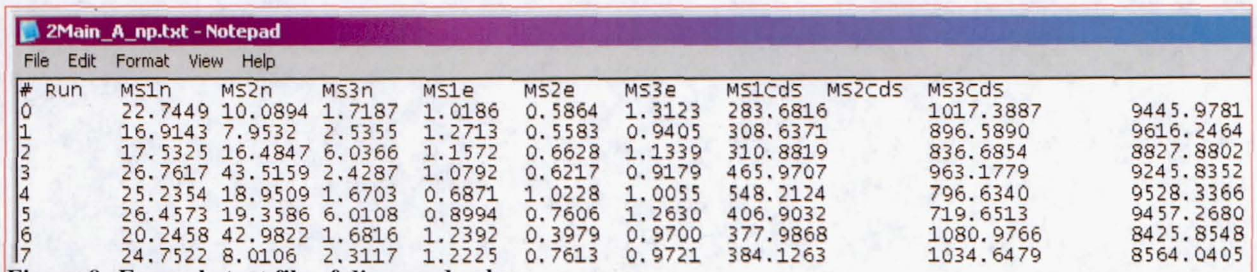


Figure 8: Example histograms and scatter plots of dispersions



```
# Run   MS1n    MS2n    MS3n    MS1e    MS2e    MS3e    MS1Cds  MS2Cds  MS3Cds
0       22.7449 10.0894 1.7187  1.0186  0.5864  1.3123  283.6816        1017.3887       9445.9781
1       16.9143 7.9532  2.5355  1.2713  0.5583  0.9405  308.6371        896.5890        9616.2464
2       17.5325 16.4847 6.0366  1.1572  0.6628  1.1339  310.8819        836.6854        8827.8802
3       26.7617 43.5159 2.4287  1.0792  0.6217  0.9179  465.9707        963.1779        9245.8352
4       25.2354 18.9509 1.6703  0.6871  1.0228  1.0055  548.2124        796.6340        9528.3366
5       26.4573 19.3586 6.0108  0.8994  0.7606  1.2630  406.9032        719.6513        9457.2680
6       20.2458 42.9822 1.6816  1.2392  0.3979  0.9700  377.9868        1080.9766       8425.8548
7       24.7522 8.0106  2.3117  1.2225  0.7613  0.9721  384.1263        1034.6479       8564.0405
```

Figure 9: Example text file of dispersed values.

The simulations then read the text file in which each row corresponds to a different Monte Carlo Cycle. This method has worked successfully in several independent parachute simulations, and makes simulation comparisons easier since a particular cycle has identical parachute inputs.

The user creating the dispersions has the ability to generate any number of values though the default is currently 3000. There are known requests for future dispersion files containing up to 10,000 values.

D. MMv10 Monte Carlo Assessment

MMv10 CPAS benchmarks were preliminarily assessed. However, input from other users of the dispersions provided insight into necessary dispersion updates before the CPAS benchmarks were completed. Updates included the need to cap the distribution tails and update the reefing cutter dispersion. It was decided to update the dispersions and continue running CPAS benchmarks as part of MMv11, which would publish the final versions.

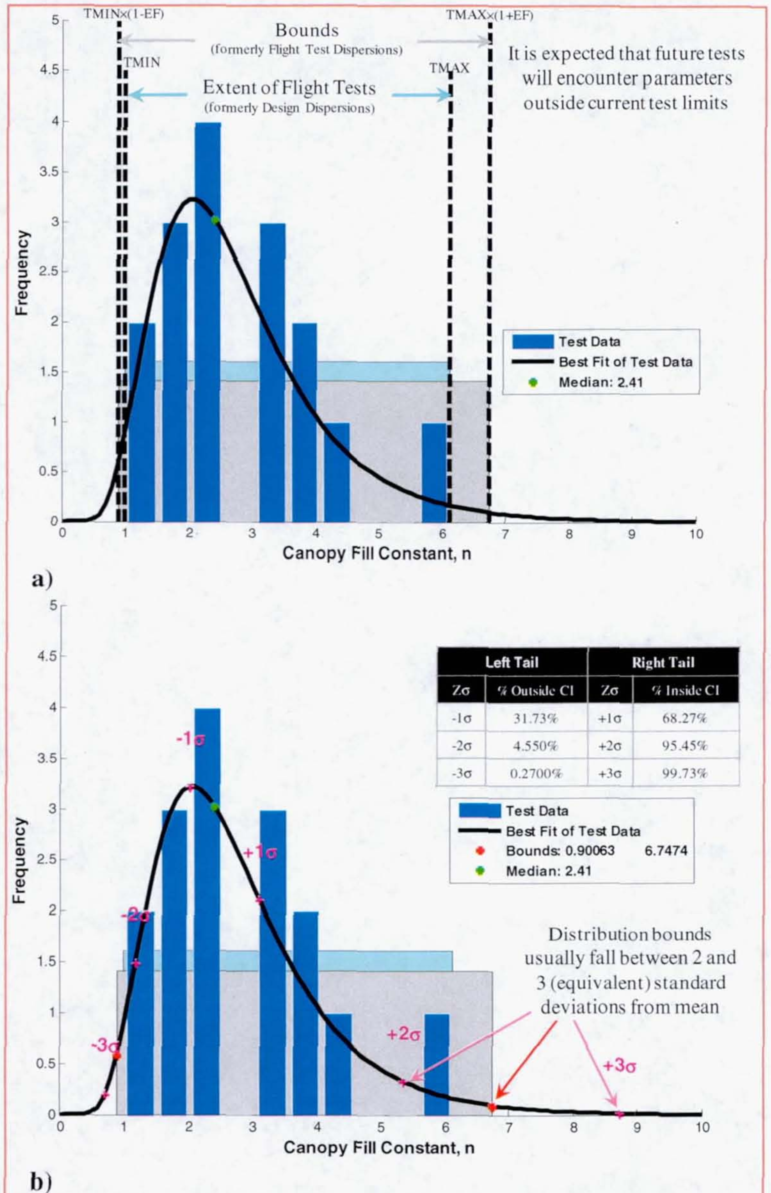
IV. Model Memo v11: Incremental Refinements

A. Capped Distributions

During the first iteration of Monte Carlos run with the statically dispersed parachute parameters, MMv10 CPAS benchmark testing, a handful of cycles resulted in excessively large loads or long inflation times. To mitigate this, the distributions were capped, though the amount by which to cap was contested. The options were to cap with an EF applied to the minimum and maximum reconstructed test values or via standard deviation (σ) levels. Note that for Gaussian distributions, standard deviations (σ) are associated with probability distributions, and percentages of data contained within standard deviation intervals are defined. For non-Gaussian distributions, although the standard deviation cannot be associated with a probability, determination of the percentage of data outside a number of standard deviations gives useful insight into the distribution.

Figure 10a and b show capping based on an EF and sigma levels, respectively. Since CPAS has historically used an EF of 5% for the drag coefficient and 10% for all other performance parameters, this method was preferred over using a sigma level. Interestingly, the engineering factor bounds generally fall between the second and third standard deviations as seen in Figure 10b giving credence to the EF implementation.

The MATLAB script was updated to include additional rules for calculation and output of floor and cap limits (also known as lower and upper bounds). Subsequently, the Python script was updated to disperse between the limits. Problems arose when dispersing the values because values that fell outside the limits were originally set to the floor or cap value causing a large cluster of data at the limits, thereby corrupting the



Left Tail		Right Tail	
Zσ	% Outside CI	Zσ	% Inside CI
-1σ	31.73%	+1σ	68.27%
-2σ	4.550%	+2σ	95.45%
-3σ	0.2700%	+3σ	99.73%

Figure 10: Bounding based on a) an engineering factor and b) sigma percentage levels.

intended distribution. Logic was added to force a random redraw.

B. Reefing Cutter Data

As more drop tests were completed, the reefing cutter data was pulled from the video timelines and increased the number of data points from 51 to 90. The same methodology was used as discussed in Section II. DFirst Statistical Assessment of Reefing CuttersFirst Statistical Assessment of Reefing Cutters. When plotted as a histogram, the data began to have more of a Gaussian distribution as seen in Figure 11. This was expected, but a lot hypothesis test was conducted with a pseudo-random number generator to determine the resulting distribution if thousands of data points were used.

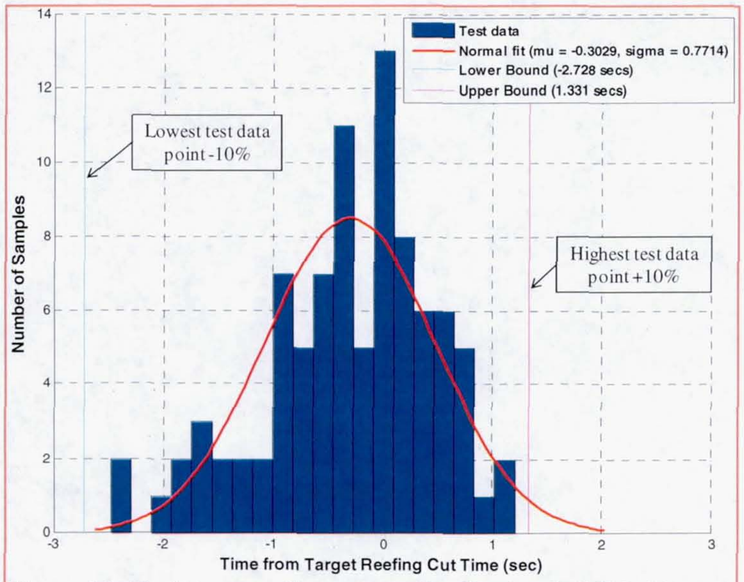


Figure 11: Update to reefing cutter distribution in MMv11.

The lot hypothesis test began with a Gaussian distribution created with a sigma of 0.5 seconds and mean of 8.0 seconds. Other sigma and mean values were tested with the same result. Then two values were chosen at random from the original lot (Figure 12 top). These two values represented the redundant cutters on a single parachute disreef event. Since the disreef occurs when the first cutter fires, the earlier of the two values was kept while the other was discarded. This was done thousands of times and a second lot and histogram were created using the earlier cutter data. A Gaussian distribution fit this new histogram, though the mean was shifted to the left (Figure 12 bottom). Therefore, if sufficient test data was available, the cutter time distribution should be Gaussian. This provided more confidence in the decision to use a Gaussian distribution on the 90 CPAS data points.

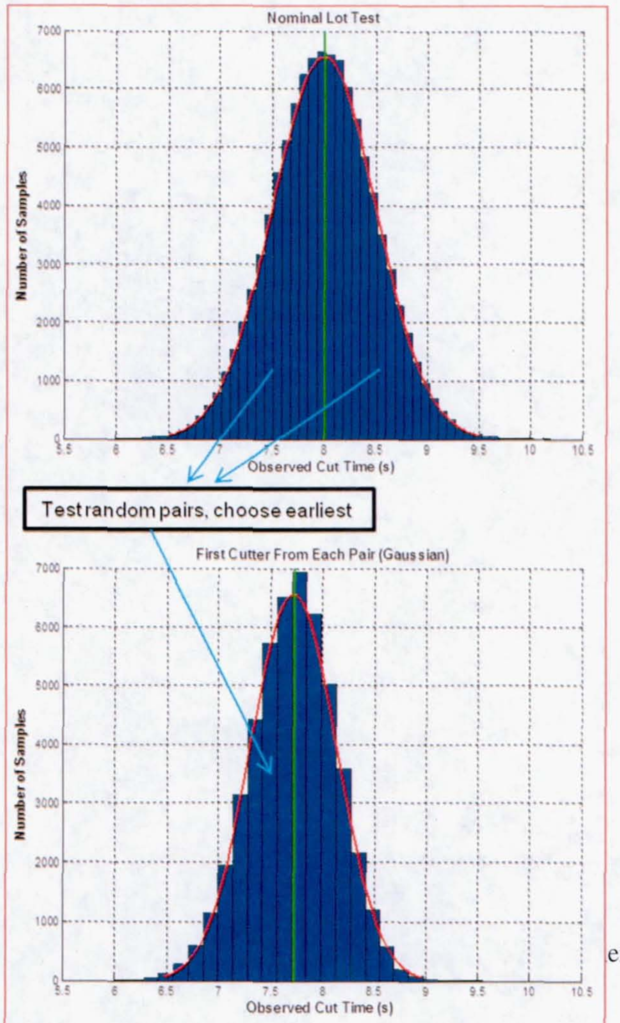


Figure 12: Reefing cutter lot hypothesis test.

To be consistent with the inflation dispersions, a cap and floor of ±10% above and below the target value were applied as seen in Figure 11. As the aforementioned figure shows, the floor and cap correspond to times of 2.728 and 1.331 seconds before and after the target, respectively. This means that for any cutter, whether eight or 40 seconds, the floor is the target minus 2.728 and the cap is the target plus 1.331.

It is known that the mean nominal reefing cutter time can be biased by temperature and the age of the pyrotechnics. Future analyses may define cutter dispersions that include the affect of scenario type (e.g. nominal reentry, pad abort) to account for the expected temperature effects.

C. MMv11 Monte Carlo Assessment

A complete assessment of the dispersions was conducted in conjunction with MMv11, similar to that done for MMv10. The only difference was that the reefing cutter time dispersions were included in the delivered text files. Each parachute had its own cutter time and therefore, using the high fidelity parachute model in the FAST simulation with its multi-parachute modeling capability, lead-lag dynamics could be examined. Results of this study are covered in the following section.

V. Further Improvements

A. Multi-Dimensional Limits Method

After assessing the MMv11 dispersions through CPAS Benchmark analysis, many cycles in the two-Main cases showed exceedances of load requirements during the Main full open inflation. Upon further investigation, it was found that unphysical combinations of inflation parameters n and expopen drove the loads to artificially high values.

To eliminate the unphysical combinations, reconstructed n and expopen values were plotted as the blue dots in Figure 13. A polygon (cyan curve) was drawn around the extreme data points using a convex hull algorithm. Then a slightly larger polygon (black curve) was generated to bound the data based on an EF of ±10%. This method was called the *multi-dimensional limits (MDL) method*. As discussed in the preceding section, the only bounding of the distributions for the MMv11 release was through an EF applied to the largest and smallest test reconstructed value (shown as red lines). Application of the MDL method will force (through re-draws as necessary) dispersions to fall within the black polygon limits.

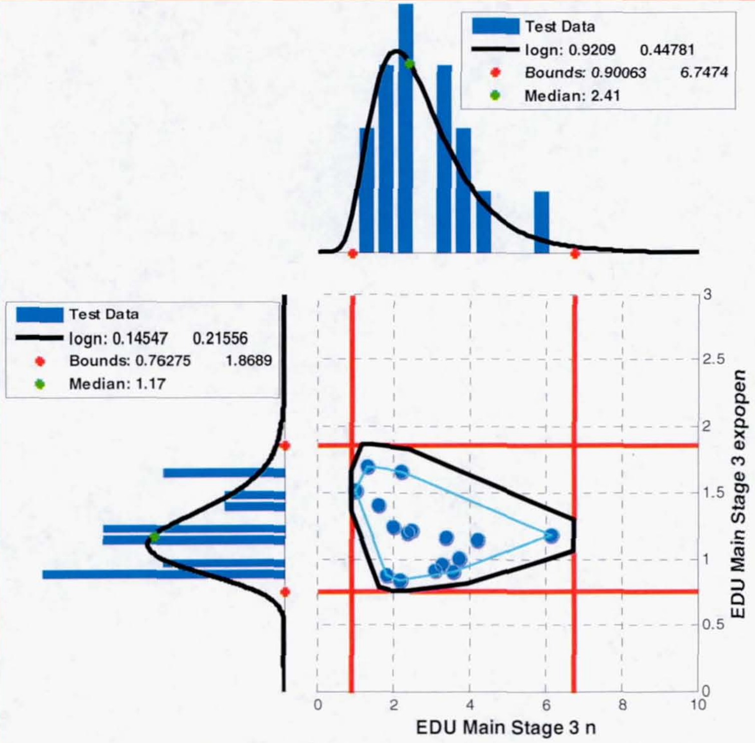
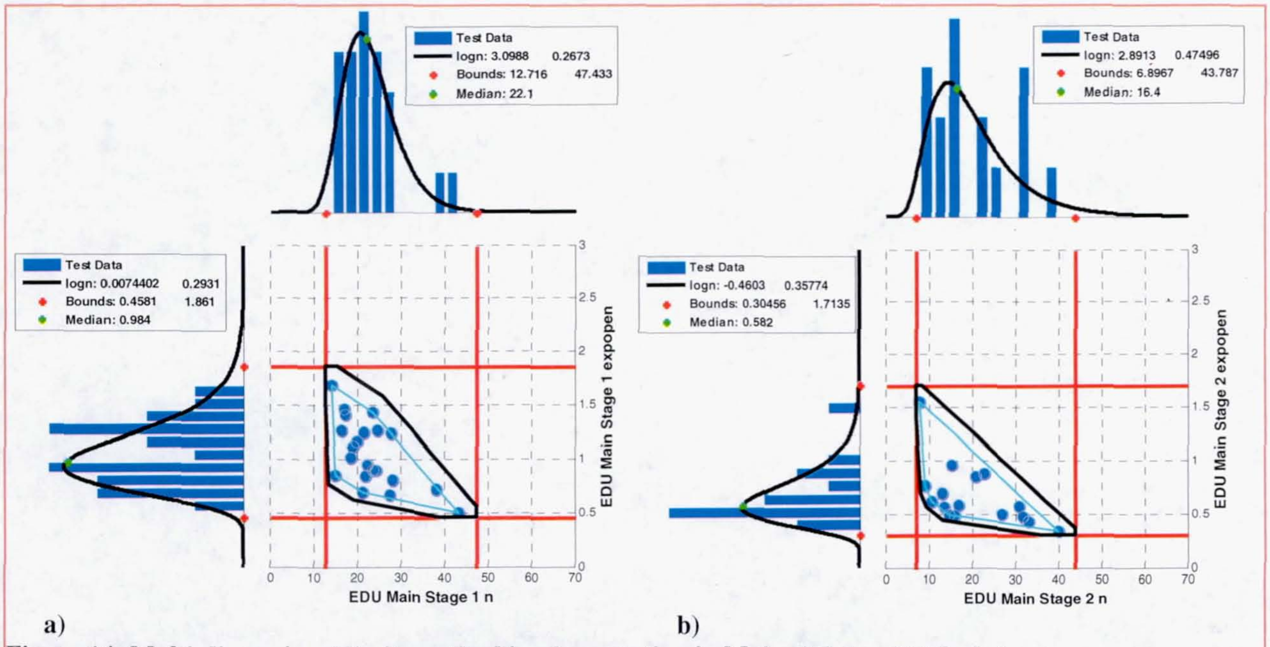


Figure 13: Multi-dimensional limits method of bounding dispersions in main full open stage.



a)

b)

Figure 14: Multi-dimensional limits method implementation in Main a) 1st and b) 2nd stages.

Though this method came about due to an issue of peak load exceedances in the disreef to full open, it can be applied to the other Main inflation stages as well. Figure 14 shows the reconstructed test data for stages 1 and 2 with the MDL method implemented.

This methodology is also being examined for use on the Drogue parachutes. Due to the additional inflation parameters, C_k and t_k, the number of combinations makes the application more complicated. Preliminary studies show that the C_k and n parameters drive the loads.

The MCL method has not yet been implemented in the Python scripts that randomly disperse parameters within the distribution, cap, and floor rules. The current plan is to incorporate this method into MMv12 assessments scheduled to be released in April 2013.

B. Deficiency in Current Dispersion Method

As previously mentioned, recall that a single parameter for a particular parachute type and stage is dispersed using the same distribution rules but a different seed. Therefore, each parachute in a cluster has a different value for that particular input. Though this allows each parachute to inflate differently, it does not account for interplay between parachutes. As seen in flight test, there is usually a lead and a lag parachute. In the current method of generating the dispersions, each parachute in a cluster has inflation parameters that can cause them to open early, as though they are all leaders, with no laggers, or vice versa. There is no logic preventing this from occurring, but the benchmarks have not shown that is has been a problem in peak loads or C_DS curves.

VI. Conclusion

The use of statistically derived parachute parameters will allow CPAS to better predict parachute dynamics that are outside the realm of testing. It will also be used in verifying the parachute system for human flight. Inflation parameters are reconstructed from test data to which a distribution curve is fit. Floor, cap, and multi-dimensional limits prevent the dispersion from being more than 10% beyond reconstructed test data point extremes. A Python script generates dispersions using the distribution rules and bounds, and ensures that each parachute uses unique parameter values. This method does not take into account cluster effects between the parachutes, though additional rules may be incorporated as the need arises. As the test program continues, the distributions, and therefore dispersions, will be updated and refined with each release of the model memo.

Acknowledgements

The authors wish to acknowledge the contributions of James Moore, previously of the ESCG CPAS Analysis Team, for writing Python scripts to operate on MATLAB-created distributions from and automating the dispersion creation process. Special thanks also go to James Dearman of Lockheed Martin for assessing the dispersions with integrated vehicle simulations and for assisting with the implementation of dispersions into FAST.

References

[1] Ray, E. S, "Engineering Development Unit Operating Modeling Parameters Version 11," JSC-65914 Rev F, Houston, Texas, November 2012.

[2] Morris, A. L., and Olson, L., "Verification and Validation of Flight Performance Requirements for Human Crewed Spacecraft Parachute Recovery Systems," AIAA-2011-2560-193, 21st AIAA Aerodynamic Decelerator Systems Technology Conference and Seminar, Dublin, Ireland, May 2011.

[3] Schulte, P., Moore, J., and Morris, A. L., "Verification and Validation of Requirements on the CEV Parachute Assembly System Using Design of Experiments," AIAA-2011-2558-415, 21st AIAA Aerodynamic Decelerator Systems Technology Conference and Seminar, Dublin, Ireland, May 2011.

[4] Barlog, S. J., "Practical Aspects of Reefing Cutter Design," 6th AIAA Aerodynamic Decelerator and Balloon Technology Conference, Houston, TX, March 1979, AIAA paper 1979-0418.

[5] Ray, E. S., "CPAS Parchute Parameter Model Memo 10 Data Delivery", ESCG-8400-12-CPAS-MEMO-0096, August 2012, Jacobs Sverdrup Engineering and Science Contract Group (unpublished).

[6] Ray, E., "Reconstruction of Orion EDU Parachute Inflation Loads," 22nd AIAA Aerodynamic Decelerator Systems Technology Conference, Daytona Beach, Florida, March 2013 (submitted for publication).