

# Automated framework to estimate probability of mechanical failure in handheld electronic devices

Manish Maheshwari, Alberto Cavallaro

Motorola Mobility LLC – a *Lenovo* company  
Chicago, IL, 60654

**Abstract:** *An automated framework for numerical simulation is developed to estimate the probability of brittle failure for a cell-phone display subjected to accelerated life cycle testing (ALT). SIMULIA Isight and the Abaqus-Python interface are used to predict the field-failure rate (FFR). The framework's adaptable definition of inputs and output parameters can be extended to other common failures in handheld devices such as the denting and cracking of the housing walls, the cracking and delamination of the solder-joints in surface mounted devices, and the shock based criteria for packaged components.*

**Keywords:** *statistical probability, field-failure-rate, FFR, accelerated-life-cycle test, ALT, consumer electronics, cell-phones, LCD, OLED, display, brittle failure, Isight, Python, automation, statistics*

## 1. Introduction

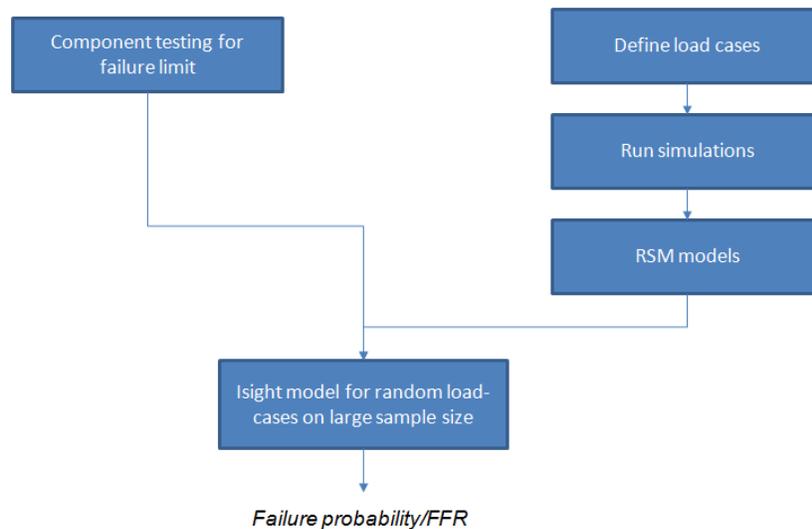
Electronic device manufacturers necessitate predicting the field-failure-rate for the life of a product to compute warranty costs. Accelerated life-cycle testing of prototypes is often used to estimate the field-failure rate and entails a sequence of tests that represent everyday use and abuse, such as dropping, scratching, bending, environmental exposure, and thermal cycling. However, the drive to reduce design cycle-time and to use fewer prototypes can result in uncertainty around these estimates. Fortunately, advancements in finite element analysis allow us to simulate most of these mechanical test cases and, with a statistical consideration for the ALT sequence, are used to estimate the FFR.

A significant shortcoming to this numerical approach is the amount of time spent by the engineer to set-up the numerous load cases as well as to post-process the large amount of output data. This study demonstrates the use of Isight and the Abaqus-Python scripting interface to significantly reduce pre and post-processing time. The inherent capabilities of Isight include tools such as design-of-experiments, survival distribution, response-surface analysis, and Monte-Carlo simulation, which facilitate defining the load conditions and sequence, as well as analyzing the output response.

This paper describes the overall framework and applies it to predicting display failure for a mobile which undergoes accelerated life-cycle testing. The framework is designed with flexibility in defining the input and output parameters so the same approach can be extended for virtually any other failure mode in the system.

## 2. Automated Framework:

The high-level flow for the full framework (Figure 1) entails the following steps: calculating the statistical distribution of the failure strength for the component or material of interest; defining and running the simulations for the load-cases to generate the response surface models for the resulting stresses in the component and material of interest; and finally run a MonteCarlo interference analysis of the imposed stress and failure strength yielding the FFR of the component or material.

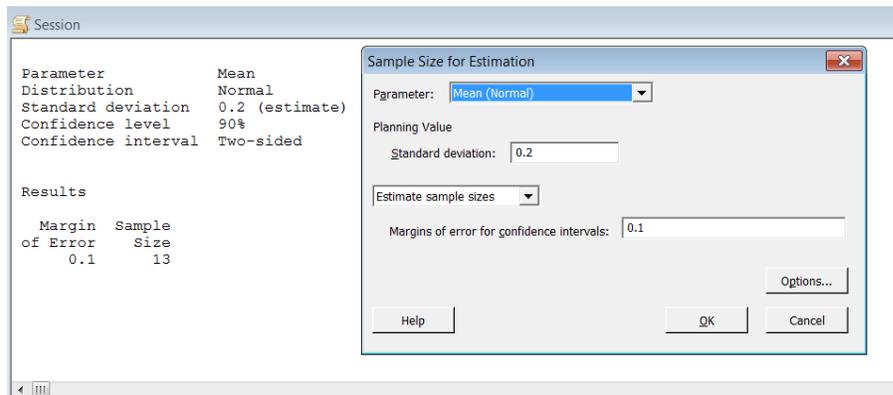


**Figure-1: High level flowchart for the framework**

The following sections highlight the step-by-step approach used to build a generic framework for predicting failures in a given component and is divided into 5 major steps. Some suggestions are included as well to yield superior accuracy from the numerical models and an example is built around modeling display failures in a mobile phone exposed to drop.

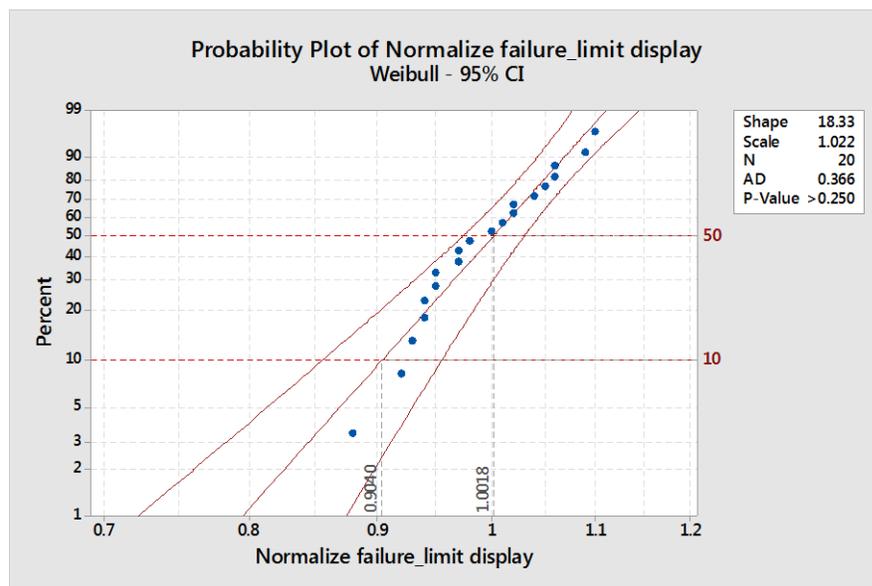
2.a. Component/material failure limit: Most material/components exhibit unit to unit variation in the failure limit for a given failure mode. Component level testing of a statistically significant sample size that exhibits the same failure mode is used to parameterize the statistical distribution of the failure strength.

Minitab or any other statistical tools, is initially used to estimate the minimum sample size. In this particular example (Figure 2) assuming a normal distribution, a 90% confidence level (e.g. 0.1 margin of error) and a standard deviation of 0.2 requires 13 units for our test. We later elect to use 20 units.



**Figure2: Estimate sample size for probability estimate for given failure mode using Minitab software**

4-point bend testing is conducted on 20 samples to extract the strength distribution for the glass in a display (Figure-3). The parameterization of the data with a Weibull fit yields the input parameters for the framework used later in steps 2-c and 2-d.

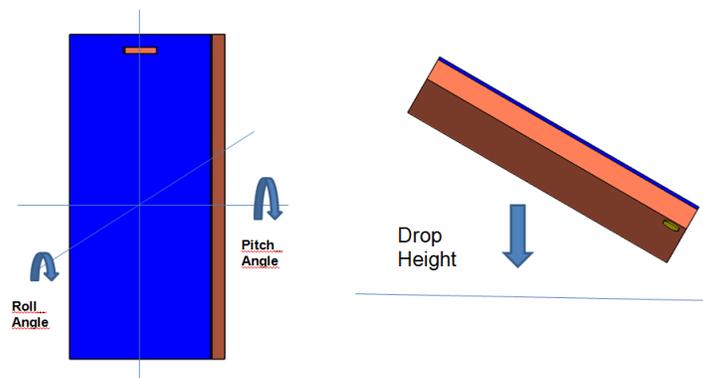


**Figure-3: probability plot showing the statistical distribution of normalized failure limit for tested samples**

2.b. Define Load cases: The load case is defined by the mechanical load and boundary conditions on the component or material. The best suited problems for the framework are the one where loads and boundary condition can be parameterized.

For a display, more than 90% of mechanical failures had been associated with system drop from a certain height. The stress on the display glass obviously varies with drop-height and drop-orientation, which can be defined by rotation angle in roll and pitch direction from base orientation (Figure-4). Therefore we have three input parameters to define a load case:

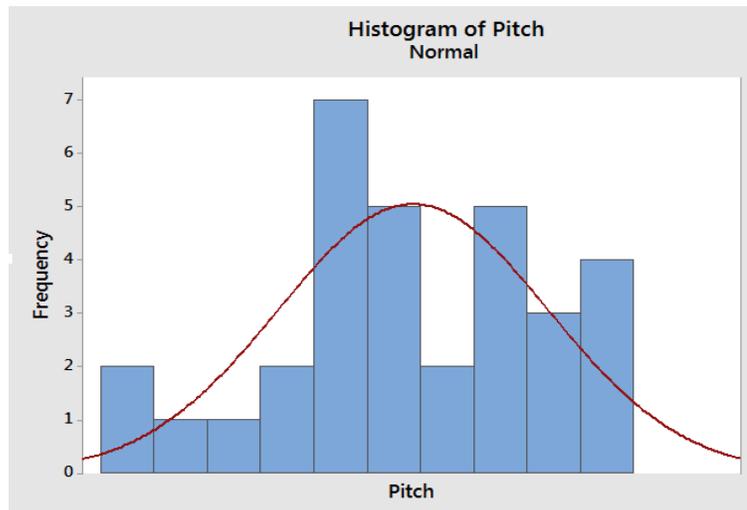
- i) Height
- ii) Roll\_angle
- iii) Pitch\_angle



**Figure-4: Load case for phone drop model**

When a phone is dropped accidentally by the end-user, the above parameters will be random and uniformly distributed over certain bounds. However, during accelerated life testing in a lab, the release drop angles are targeted to particular orientations (say face, back, side or corner drops) to simplify the test sequence and insure that most regions of interest are evaluated. This procedure yields a higher frequency of impacts close to targeted angles.

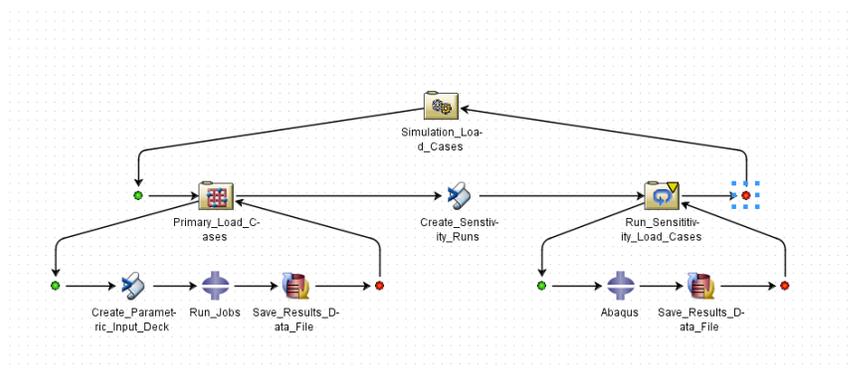
To mimic the lab scenario the distribution of drop angles and height is weighted creating a distribution for specific conditions that more likely occur in the controlled accelerated life test (Figure-5). This distribution is fitted from experimental characterization of 60+ drops for 3 phones which was recorded and analyzed with a high speed camera. This distribution becomes the input function for our framework as described in steps 2.c and 2.d.



**Figure-5: Distribution of pitch\_angle when dropped from various heights**

2.c. Run Drop simulations: By defining bottom drop as the zero-angle, the pitch and roll angles of  $-180^{\circ}$  to  $+180^{\circ}$  represent all of the possible drop orientations. We use a DOE and response-surface-model to predict stress values for any random orientation in this space.

The Isight Sim-flow (Figure-6) generates the simulation load-cases, creates the input files, runs the simulation jobs, reads the results, and saves them into a data file. The data file will be later used to create the response-surface-model.



**Figure-6: Isight Sim-flow showing framework to automatically generate load-cases and submit the simulation jobs on cluster**

Furthermore, the functions used of this flow entail the following:

- i) DOE: It generates 50 random drop angles and heights within the specified bounds. Optionally a latin-hypercube approach can be used in this specific case but any other approach with sufficient number of cases will also work.

*Tip: If part of design space is safe for a given problem – that it can be skipped and results can be made zero or lower than the failure limit using calculator function without running the simulation jobs to save time.*

- ii) Create\_parametric\_input\_Deck: A python script is used to read the DOE parameters as input for the orientation and height, and to modify the template for the simulation input deck. Here, the drop angles and height are specified as the parameters for the drop simulation model. These parameters will be used to rotate the model with \*NCOPY command.

```
****
*PARAMETER
*PITCH_ANGLE = <pitch_angle_from_script>
*ROLL_ANGLE = <roll_angle_from_script>
*DROP_HEIGHT = <drop_height_from_script>
****
```

- iii) Run\_Jobs: The Abaqus function in Isight reads the input deck created by the script, runs the simulation jobs, and reads the peak stress on the display from simulation output database (\*.odb) file.

*Tip: We use the simulation grid engine to submit jobs for this display problem since the simulation jobs are very complex and run for a long time. The python script creates the input decks in Isight while another script that runs when the job is complete on the simulation grid engine generates the results.*

The display strain is written as part of the history output with the “ANTIALIASING” filter and at a 20 kHz frequency. The peak strain is read using the python script on a pre-specified set and then stored in the data file against the matching load case.

```
***
*OUTPUT, HISTORY, FILTER=ANTIALIASING, TIME INTERVAL=0.005
*ELEMENT OUTPUT, ELSET=DISPLAY-GLASS
LEP2, LEP1, LE11, LE22
***
```

- iv) Data exchanger: The data exchanger function writes the results in text file in the CSV (comma separated values) format where first three columns write the pitch\_angle, roll\_angle and height and the fourth column writes the peak strain data for display (Figure-7)

Pitch_Angle,	Roll_Angle,	height,	Peak Strain
60.8,	-45.2,	3.2,	0.98
-121.3,	176.2,	4.8,	0.67
22.4,	-2.0,	5.0,	0.39
105.1,	26.4,	2.7,	1.67

**Figure-7: Results stored in CSV text file**

- v) Script\_Create\_Sensitivity\_Runs: Another python script reads the results from the data file and compares the stress/strain value against nominal failure limit. If the maximum stress/strain value is >30% of the nominal failure value, it creates an additional 2 to 4 simulation jobs around that orientation to capture the sensitivity in pitch and roll directions.

The sensitivity runs shows how fast the high stress/strain changes when we change the drop angle by few degrees around a high stress condition.

The script creates the input decks for sensitivity runs and updates the loop parameter to adjust additional jobs that might be needed.

- vi) Loop function: It runs all the sensitivity runs by reading the input deck and appending the results into the results file.

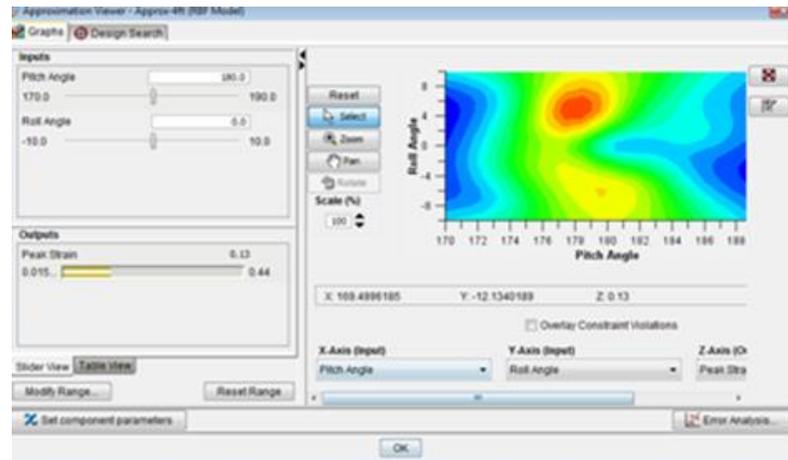
2.d. Response-surface-model: For this display problem, we run between 200 to 300 simulations at various drop angles, drop heights and sensitivity angles around the high-risk orientations. The results are then stored in the data file in CSV format.

*Tip: Most RSM models try to minimize error over the whole design space and this practice can often average and thus ignore highly localized peak values. Based on our experience we have divided our design space in multiple segments (4 in this display problem) to get higher accuracy for each of the smaller design spaces.*

Steps:

- i) The Isight Approximation function is used to create the response-surface-model.
- ii) The RBF model is best suited for display problems since there are localized peaks at specific angles.
- iii) The results data file is used as an input to the Approximation function.
- iv) The “Dynamic file” is used as the data input file so that the program can update the input file as more simulation jobs complete in the background. Therefore, the final results are up-to-date with the current data.
- v) The elliptical basis function is best suited to minimize the error in display problem but this is very specific to type of runs and should be verified for each new program and data
- vi) Cross-validation is used for error analysis accepting an RMS error less than 10% for the model.

In Figure-8 the RSM model is shown for a smaller design space around a high stress area: the red color envelopes the high risk region.



**Figure-8: one of the RSM model showing hotspot near high risk simulation**

- 2.e. Estimate probability of failure: Once we have the response-surface-model to predict the simulation results for all parameters, we can run the Monte-Carlo simulation to create a large number of load-cases without running each finite-element simulation.

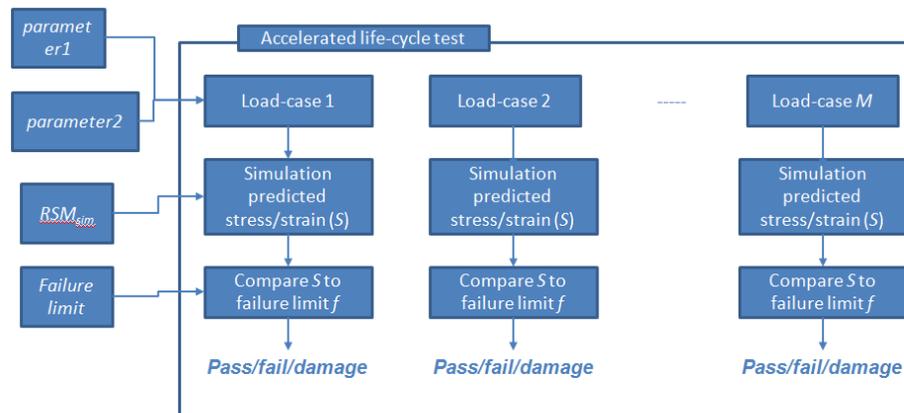
*Tip: The model accuracy is highly dependent upon number of simulations run over design space and using FMEA or experience to capture most of the high risk orientation in design-space.*

A model is built around the actual accelerated-life-cycle test (ALT). The test sequence is replicated into Isight using Monte-Carlo simulation, RSM and standard math operations. Figure-9 shows the flow-chart for a generic ALT and Figure-10 shows the flow-chart for an Isight model that runs the ALT with  $n_{total}$  number of units.

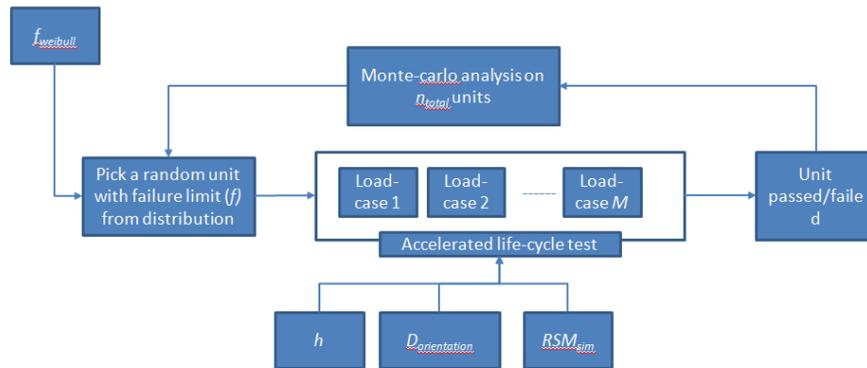
While the general ALT includes a sequence of mechanical, thermal and environmental loads, we focus on the mechanical loads as they are the ones that apply to the anticipated display failures. Using simulation data and the Isight model, we can run the sequence of load-cases on several hundred or several thousand units instead of the 20 to 30 units used during physical testing. Steps in modeling the ALT flow to calculate probability of display failures follow.

- i) ALT loop is defined using MonteCarlo loop with input parameters as pitch\_angle, roll\_angle and height. The distribution for each parameter is defined using weibull distribution captured from testing.
- ii)  $n_{total}$  units are number of units used that goes through sequence of load cases.

- iii) For each unit, a Monte-Carlo loop will randomly select failure limit from the Weibull distribution captured in step 2.c
- iv) For each load-case (defined by randomly selected pitch, roll angles and height), we calculate the peak-stress from the response-surface-model associated with that load-case
- v) The peak-stress is now compared against the failure limit to decide whether the unit survived load-case or not. If survived it is subjected to next load case
- vi) If the failure is defined as accumulated damage then the damage\_parameter is updated at the end of each load
- vii) Once the unit has been through all the load-cases, we can assess pass/fail/total-damage for the unit.
- viii) The process is repeated for each unit and we count the total number of failure units  $n_{fail}$
- ix) Probability of failure of a given component is the ratio  $n_{fail} / n_{total}$



**Figure-9: Flow-chart showing accelerated-life-cycle-test case model**

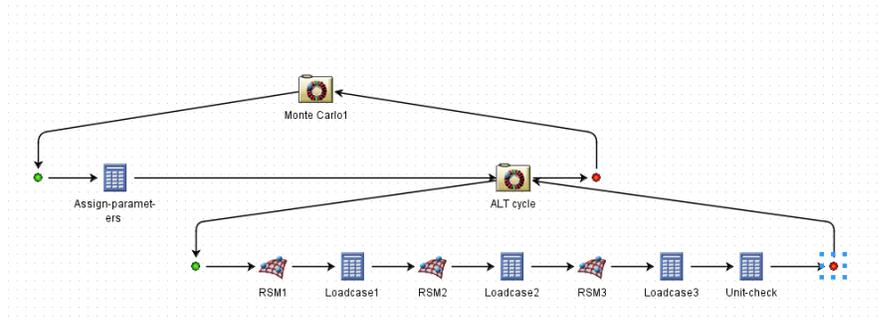


If  $n_{fail}$  units failed from  $n_{total}$  units, then probability of failure  $P_{fail}$  is:

$$P_{fail} = n_{fail} / n_{total}$$

**Figure-10: Monte-Carlo simulation model to assess probability of failures on large number of units**

Figure-11 shows the snapshot from the I-sight model showing 2 Monte-Carlo loops. The outer loop picks a random unit with a failure limit from the Weibull distribution. The nested inner loop runs the ALT for a random load-case and captures if unit passed/failed.



**Figure-11: I-sight model snapshot for probability of failure calculation**

### 3. Extensions of the study

- 3.a. Calculate Field-Failure-Rate: While the lab testing for prototypes generally have predefined sequence of loads and targeted height and orientations for drop testing, in the real-world the loads and orientations are random.

As we used a normal distribution for pitch\_angle, roll\_angle and height about specific orientations and heights for ALT, we can change the distribution to uniform to predict the actual field-failures which are random about the orientation. If we employ a large number of units (i.e. 10,000 for the display failure example) in the analysis, we can quantify the related field-failure rate.

- 3.b. Ductile failures: Section-2 was heavily focused on brittle failures in display. However, the Isight model is flexible to define ductile failure or damage to component as well. In section 2.e we can use the peak-stress/strain to calculate total damage to the material/component. Or the damage model can be defined into an Abaqus-material model and we can read the damage variable instead of peak-stress and move down the sequence of loads to capture total damage to the component.

### 4. Advantages and limitation of the approach

- 4.a. Averaging in response-surface model: Most response-surface models or statistics tend to ignore/average the outliers to get accuracy in maximum design space of input parameters. Take the example of a random drop for a hand-held device. In the continuous space of roll and pitch angles from  $-180^\circ$  to  $180^\circ$ , only few orientations may result in failure for a given height and most of the design space may remain safe. If we try to capture the whole continuous space in one RSM model, the model may totally miss the high risk zone(s). Therefore selectively dividing the RSM model in multiple spaces using FMEA or past experience gives us choice to look at the broader orientations without missing local peaks.
- 4.b. Damage to other components is ignored in sequence of load cycle: since we are focusing on one-component during our study here, we are ignoring damage to other components and assembly during the sequence of load cycles. Several times, the damage to other components may lead to higher probability of failure in our component of interest. Based upon heuristic models, additional correction in stress/strain can be added to the sequence of loads address this issue.
- 4.c. Validation: Receiving FFR (field failure rates) data from field is challenging unless we warranty the product for given failure. Therefore our model focused on validating a total of 4 products with 100+ sample size as physically tested in our lab. When a good correlation was found with these products, we were able to build confidence that the FFR data should be accurate for field and start using that to assess warranty cost.

## 5. Summary

This study demonstrates the capability of an automated numerical framework to predict the probability of brittle failure in a display during a targeted load cases like ALT for predicting field failure rates. The approach is generic and can be used for modeling most mechanical failure including but not limited to brittle failure of glass and ceramics as well as defining the damage accumulation in ductile materials. Like most numerical models, the design is sensitive to the numerical inputs, and the number of simulation jobs run to build the RSM models. Very precise steps should be taken leveraging the experience from past failure modes when defining these parameters.

### Trademarks:

- 1) Isight and Abaqus are trademarks of Dassault Systèmes Simulia Corp.
- 2) Minitab is a trademark of Minitab Inc.

### Acknowledgement

This work was performed at the Motorola Mobility LLC laboratories

### Disclaimer

**This communication may contain information and data that has been normalized. Any opinions or information expressed regarding 3rd party products or services do not constitute an endorsement from Motorola Mobility LLC. Neither Motorola Mobility nor the Author shall be liable for any unauthorized use, or amendment, or the consequences thereof.**