

# Multimodule User's Manual

Sean Moseley, David Bogy

Department of Mechanical Engineering

University of California at Berkeley

Computer Mechanics Laboratory

June 3, 2005

# Outline

- I. Goals of This User's Manual
- II. What in the World Does This Multimodule Thing Do Anyway?
  - A. What Multimodule Does
  - B. Walkthrough of the Multimodule Window
  - C. Examples of How to Use Multimodule
- III. Multimodule Options
  - A. Sample vs. Search and Interpolation
  - B. Cross vs. Solo and Independence
  - C. The Global Parameter Type
  - D. The Monte Carlo Search Option
  - E. Subtypes
    - i. Uniform and the Boundary Option
    - ii. Median, Limit, and Unidirectional Options
  - F. Rail/Point Motion
  - G. The Yield and Results Buttons
    - i. Yield Limits
    - ii. The Results Button
- IV. Multimodule Examples
  - A. Independent Parameters: Crown & Camber
  - B. Dependent Parameters: Trailing Pad Position
- V. Summary

## I. Goals of This User's Manual

The first goal of this user's manual is to make the Multimodule portion of the CMLAir Air Bearing Design Code better understood. To this goal, it will explain in detail the multitude of options, checkboxes, and radiobuttons that grace the Multimodule window. Emphasis will be placed on first understanding what Multimodule is intended to do, and on this basis, build an understanding of the detailed options that make Multimodule powerful. After the options are explained, a few examples will be given that demonstrate the use of some of the options. It should also be noted that this user's manual assumes familiarity with the vocabulary of slider design (crown, camber, pitch, roll, fly height, etc.) as the scope includes explaining Multimodule, not the complicated theories of slider design.

## II. What in the World Does This Multimodule Thing Do Anyway?

The short answer: Multimodule is an automated way of running multiple simulations with systematic changes in the input parameters. As one might expect, the long answer is more complex, but also much more instructive.

### II.A. What Multimodule Does

In designing a slider, one specifies values for many parameters, like the slider rail lengths and positions, recess heights, crown, camber, etc. While an experienced slider designer will be able to come up with good starting values for these parameters, there might be an easy way to improve the slider's performance by making slight changes in these parameters. Also, one may be interested in how manufacturing tolerances will affect the performance of the slider. Multimodule is designed to allow easy and systematic variation of the parameters in question so that the user may find the "best" value or demonstrate how slight changes in the manufacturing will affect the flying characteristics. This is accomplished by generating sets of input parameters using the various options, called "sample cases", and sending them one-by-one to the static air bearing solver. The solver generates the results for each case and Multimodule collects the results in a nice tabular format.

It is best to think of Multimodule as an interface surrounding the CML static code, feeding it variations in the parameters (sample cases) and storing the results after those cases have been solved by the static air bearing solver. See Figure 1 below for a graphical representation. The various parts of this figure will become clear later.

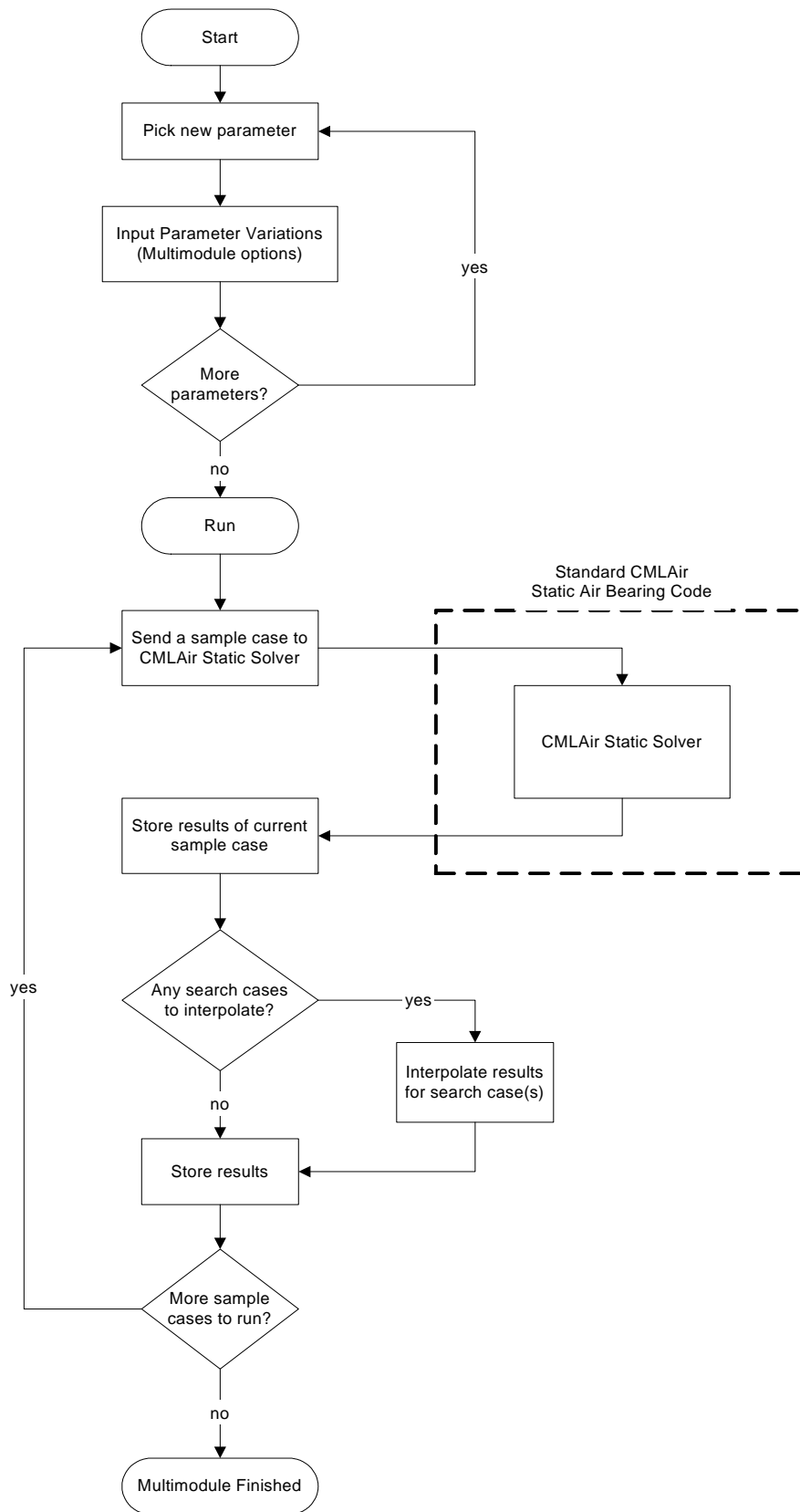


Figure 1. Flowchart for Multimodule Operation

## II.B. Walkthrough of the Multimodule Window

Before using the Multimodule, you must first design a slider as if you were going to run a regular static simulation (create rails, specify other parameters, etc.). You can also open up a slider case that has already been created. After the slider design is finished, click on the Multimodule tab.

After the Multimodule window is opened you will immediately notice a picture of the slider's rails in the upper right corner. This area is used for changing the rail and rail point positions, to be explained later. It also serves to remind you which slider you are working on. The remainder of the Multimodule window is filled with drop-down boxes, check boxes, radiobuttons, and the like. Some of these options will be grayed out, indicating that they are not available with the current options.

The first step is to identify the parameter that you want to study. The drop-down box labeled "Parameter Type" lists the available parameters that you can study: slider geometry (length, crown, twist, taper, etc.), suspension parameters (load, pitch offset/torque, etc.), slider features (rail position, rail points), manufacturing parameters (mask, recesses), and the disc RPM.

After identifying the parameter that you want to study, you will then need to specify how much you want that parameter to vary (the range). One point in this range will be the "base value" that you specified earlier when designing your slider. The start and end of the range represent the smallest and largest values the current parameter will be allowed to take on. Enter the start and end of the range, relative to the base value. Multimodule will then show you what the absolute range of the parameter will be in the second column, and the units on the right. Note that the range does not have to be symmetric. See Figure 2 below.

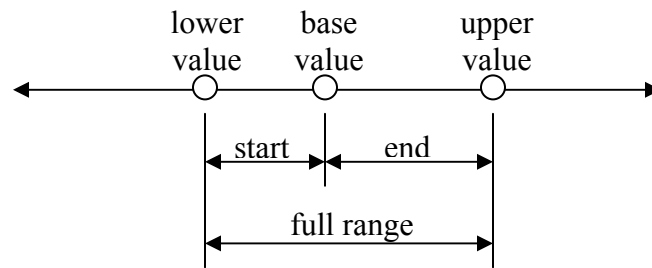


Figure 2. Specifying the Range of a Parameter

After you specify the range of the current parameter, you will choose the sample options and the search options. For now, just think of the sample and search options as a way to tell Multimodule how to pick new values for the current parameter. These options will be explained in detail in section III.

After you are satisfied with the sample and search options that have been chosen, click on the “Add” button to add the current parameter to the “Active Parameters” list. You must click the “Add” button to save a new parameter or the “Update” button to save changes to a parameter that has already been entered. The “Active Parameters” list reminds you how many parameters you are changing for this Multimodule run. After you have added or updated the current parameter, you can start over again with another parameter, select a parameter from the active parameter list for modification, or you can proceed if you are finished entering parameters.

On the right side of the Multimodule window, below the picture of the slider rails, are three buttons: Yield, Results, and Run. The Yield button can be used to specify “yield limits” for your slider and the Results button is used to view the results of the simulation. Both the Yield and Results buttons are explained in section III.G. The Run button starts the Multimodule run (beginning at the block labeled “Run” in Figure 1).

That’s it! Multimodule will begin sending sample cases to the static solver and collecting the results. A window labeled “Job Status” will appear, showing the location of the current file, where the solver is located (local or remote), the index number of the current

sample case being solved (MM #), the elapsed time for the current sample case, and the status of that simulation. Click on the filename for more information about the number of remaining and completed cases, along with an estimate for the amount of time remaining to finish this Multimodule run. You can click on the “Show” button to see the details of the current sample case, or stop the entire Multimodule run with the “Stop” button.

After the Multimodule run is finished, click on the “Results” button to see the results for your Multimodule run. You can view the results in absolute or relative form, copy everything to the clipboard, export the run files for specific cases, or view histograms for the search cases. These options will be explained in section III.G.ii.

### II.C. Examples of How to Use Multimodule

Multimodule is extremely useful for investigating the sensitivity of a slider design to manufacturing tolerances, tweaking the slider design for robust flying characteristics, or making incremental improvements of existing designs. The Monte Carlo simulation option for the search cases is well suited to performing statistical analysis of the performance of a slider design—simply enter the standard deviation and number of search cases to run, and you have an instant Monte Carlo simulation (see section III.D).

Wonder if you should lengthen the side rails by 10%? Multimodule can run cases for you with slight increases in the side rail lengths to determine if a longer rail would provide better flying characteristics.

Are you thinking of modifying your suspension, but wonder how new variations in the load and pitch torque will affect the slider? Multimodule can create cases showing how independently changing the load or pitch torque will affect the slider. Alternatively, Multimodule can create cases with variations in both the load and pitch torque at the same time.



### III. Multimodule Options

Multimodule has a multitude of options, which can be confusing and intimidating at first. This section of the user's manual seeks to explain these options and make Multimodule more friendly.

#### III.A. Sample vs. Search and Interpolation

A clear understanding of the difference between sample space and search space is critical to using Multimodule effectively. The easiest way to remember the difference is to always think: "sample" means "simulated results" while "search" means "interpolated results". Taken from the previous Help file:

"It is important to have a strong understanding of the relationship between sample and search space in order to maximize the efficiency of the MultiModule. As described in the previous section, sample space defines the nodes for which results are generated using an air bearing solver (quick4 or quick5). Search space results are generated through linear interpolation of sampled results. As you might expect, a good sample set is one which properly characterizes the result space. For a result space that is highly linear, a sparse sample distribution is sufficient. For non-linear result spaces, a dense sample distribution may be necessary."

What this means is that only the sample cases are actually sent to the air bearing solver. The results will be identical to those you would get if you had entered the sample case parameter values as your original slider design values and run a static simulation. The size of the sample space (number of sample cases) is directly related to the amount of time required to complete the Multimodule run.

In contrast, the search case results are not the result of an air bearing simulation, but simply a linear interpolation of the sample case results. Because of this, the search case results are *not* the same as if you had entered the search case parameter values as your original slider design values and run a static simulation (the size of the search space has almost no effect on the amount of time required to complete the Multimodule run). Because of this linear interpolation, the quality of the search case results is directly correlated to the quality of the sample space.

To illustrate this difference, imagine the fly height of the slider was sinusoidally dependent on one of the parameters. In Figure 3, the asterisks represent the sample cases while the circles represent the search cases.

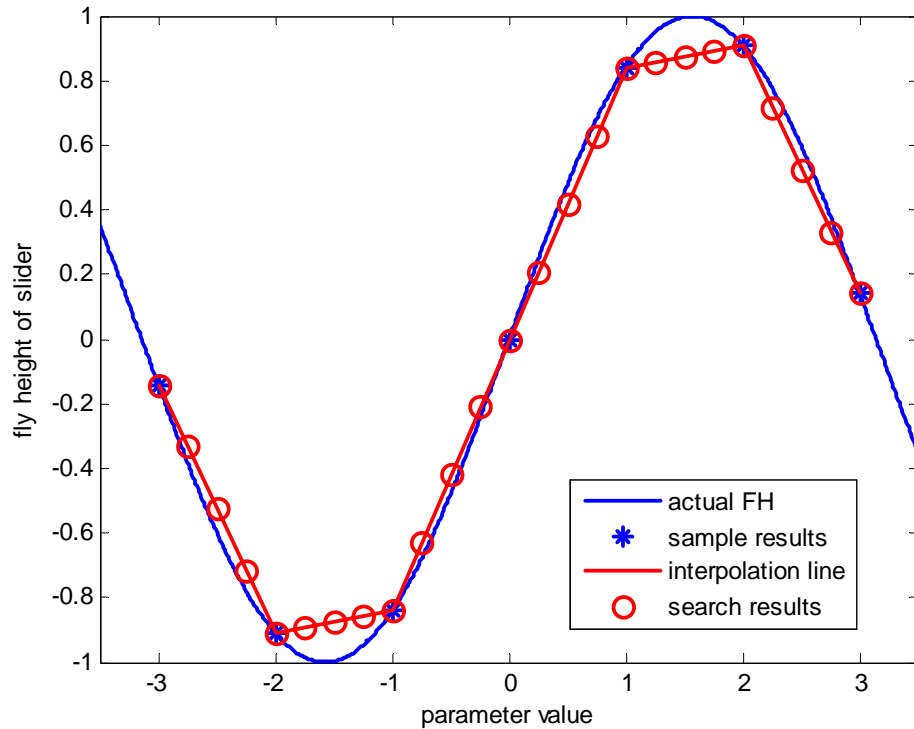


Figure 3. Illustration of Interpolation of Sample Results to get Search Results

Note that the sample results (the asterisks) are coincident with the actual FH line (solid line). This is because the sample results are obtained by running the air bearing solver for the sample cases. In contrast, the search results are based on a linear interpolation of the sample results. The interpolation line (dashed line) connects successive sample results in a linear interpolation. The search results (the circles) lie on the interpolation line. Notice that some of the search results coincide exactly with the sample results. This occurs when a search case is exactly the same as a sample case, so the sample results are simply copied to the search results. Notice that the accuracy of the search results varies from pretty good ( $-1 < \text{parameter} < 1$ ) to very bad ( $-2 < \text{parameter} < -1$ ). This is why it is important to choose the sample cases wisely in order to make the linear interpolation accurate. Imagine a case where the sample frequency was much lower; say one half of

what it was in Figure 3. While the sample results would be correct, as always, the search results would have significant error because the sample results did not adequately describe the relationship between FH and the parameter in question. Figure 4 illustrates this scenario.

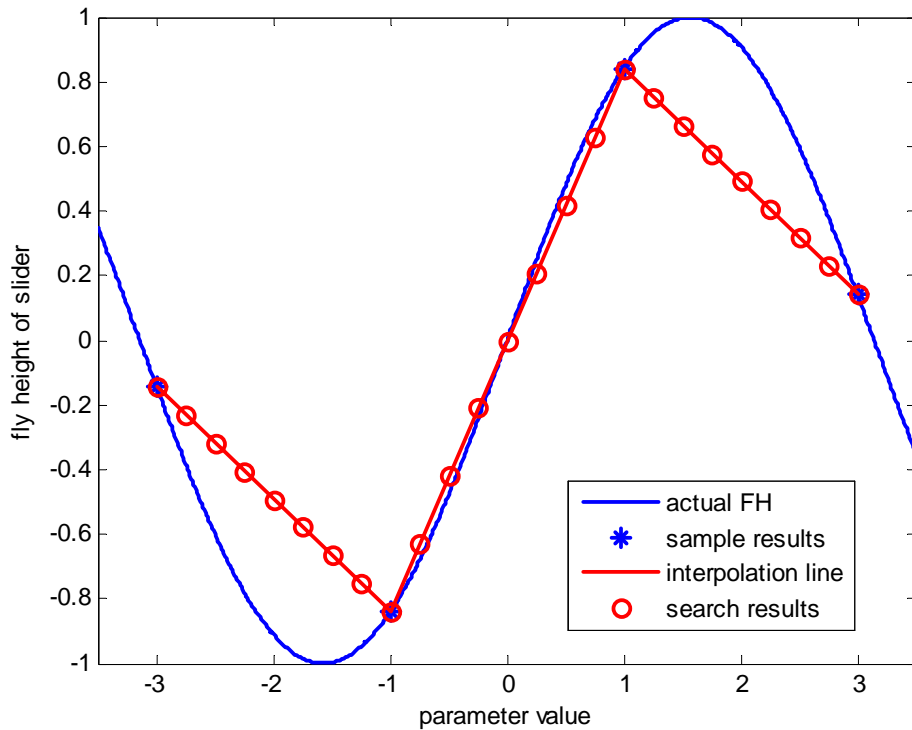


Figure 4. Illustration of Inadequate Characterization

While this example is fictional, the concept applies to real sliders in an identical manner. If you don't know beforehand what the relationship is, first run a "coarse" sample set (low frequency) with a "fine" search set (high frequency). Save the results and then run again with a "fine" sample set and the same search set as before. Compare the new search results with the previous search results. If the differences between identical cases are small, then the "coarse" sample set adequately characterizes the relationship. If the differences between identical cases are large, then the "coarse" sample set does not adequately characterize the relationship. Repeating this process will find a sample set of adequate size to characterize the relationship. These examples have been between an

arbitrary parameter and the FH, but the same things can be said about any parameter and any calculated value (result).

### III.B. Cross vs. Solo and Independence

Now that we understand the difference between Sample and Search, put it aside for a while and think about the interaction of the parameters. In Multimodule, you can change as many parameters at the same time as you want. While this allows you to gather a lot of data, running the actual simulation could take a month if you make inappropriate choices.

Imagine that we are interested in how variations in the RPM, crown, and camber affect the flying characteristics of our slider. For simplicity assume their base values are all zero, with a range of +/- 3 (acknowledging that a RPM in this range is not reasonable). We could produce the sample cases by 1) varying each parameter independently (set crown and camber to their base values while changing RPM, and vice-versa) or 2) varying each parameter dependently (change crown, camber, and RPM all at the same time). The first case describes an instance of making all three parameters “Solo” while the second case describes one of making all three parameters “Cross”.

Let’s now go to the description from the previous Help file, describing the difference between Solo and Cross:

“Solo variation of a parameter means that the parameter is varied independently of other parameters. All other parameters are set to the base amount when the solo parameter is varied. This should be used when the effect of this parameter is assumed to be independent of all other parameters.

Cross variation of a parameter means that the parameter is varied through all possible permutations of the other cross parameters. Thus, a full matrix of input variations is generated for cross parameters. This should be used when the effect of this parameter is assumed to be dependent on all other cross parameters.

The number of nodes in sample space varies linearly with the number of defined solo parameters and varies exponentially with the number of

defined cross parameters. You may construct sample and search spaces using any combination of solo or cross parameters.”

Knowledge of whether a parameter is independent of another is gained through experience. Making an independent parameter behave as a dependent parameter does no real damage, it just introduces unnecessary computation and increases the simulation time. However, making a dependent parameter behave as an independent parameter can cause problems because of the dependent interaction between parameters that has been ignored.

Let us return to our example of RPM, crown, and camber as the parameters. In case 1), we only change one parameter at a time, setting the other parameters to their base values. This situation is illustrated in Figure 5.

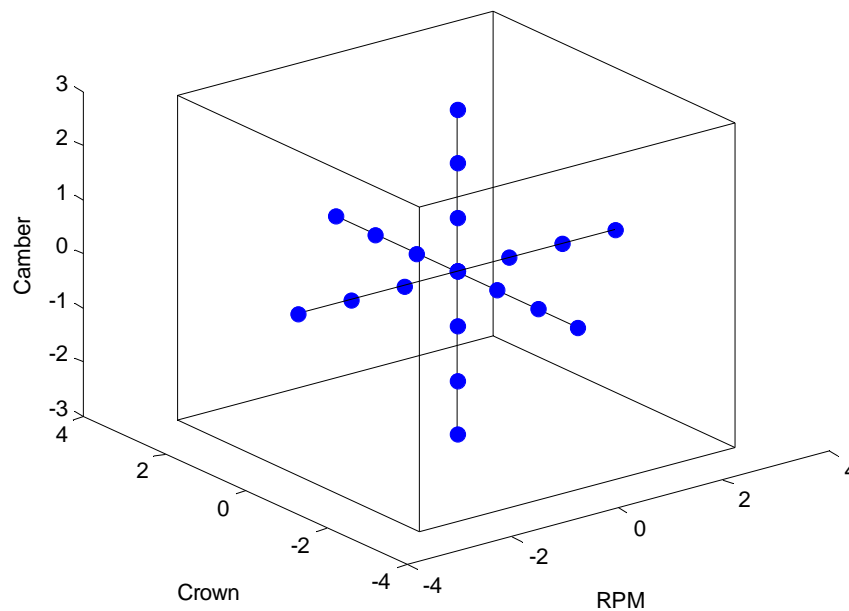


Figure 5. Three parameters, all type Solo.

In Figure 5, each parameter is visualized as a different orthogonal direction, and the space of all possible parameter values is visualized as a “cube”. Notice that as each parameter is changed, the other parameters only take on their base values. This illustrates having all three parameters independent of each other—so the type is “solo” for all three

parameters. In case 2), we change all three parameters at the same time, taking on every possible combination of the parameters. This situation is illustrated in Figure 6.

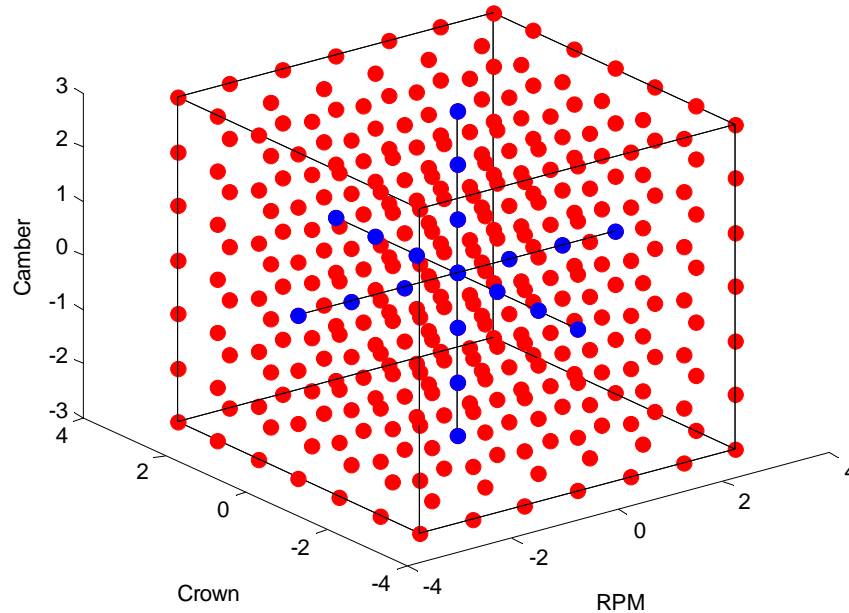


Figure 6. Three parameters, all type Cross

The points in Figure 6 are all equally spaced and equally distributed in the three directions. Here, we have highlighted the points in common between Figure 5 and Figure 6 in blue for comparison. It is clear from this figure that as each parameter is changed, the other parameters also change at the same time. This illustrates having all three parameters dependent on each other—so the type is “Cross” for all three parameters. It is also clear that having parameters of type Cross produces many more cases than when the type is Solo.

The immediately obvious question is “Do all the parameters have to be of the same type?” The answer is no. We first gave the examples of all three parameters Solo and Cross to make the distinction clear. It is actually important to think about how the illustration would look for different combinations of Solo and Cross parameters. Now imagine the illustration if RPM was Solo, but crown and camber were Cross. Recalling the definitions given earlier, we know that as RPM was changed, the other parameters

(crown and camber) would stay at their respective base values. Also, we know that both crown and camber would change at the same time, since they are both of type Cross.

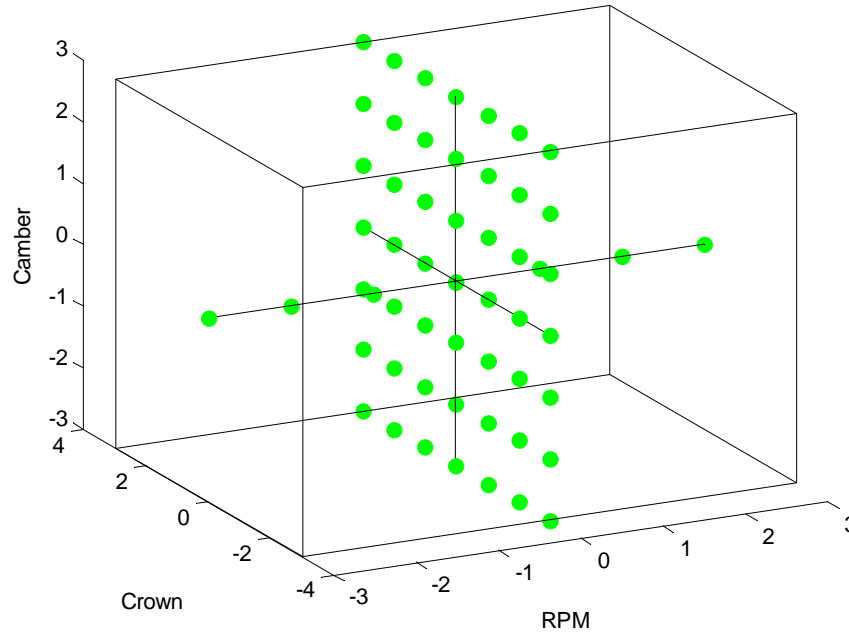


Figure 7. Three parameters, one of type Solo, two of type Cross

If Figure 7 is the illustration you were picturing, you are correct. If not, think about why this is how RPM=solo, crown=cross, and camber=cross would appear. This situation can be thought of as a combination of Figure 5 and a single “slice” of Figure 6 at RPM = 0. Again, this illustration will work for any three parameters.

Next, imagine the result for RPM=solo, crown=solo, and camber=cross. Figure 8 shows the illustration for this case.

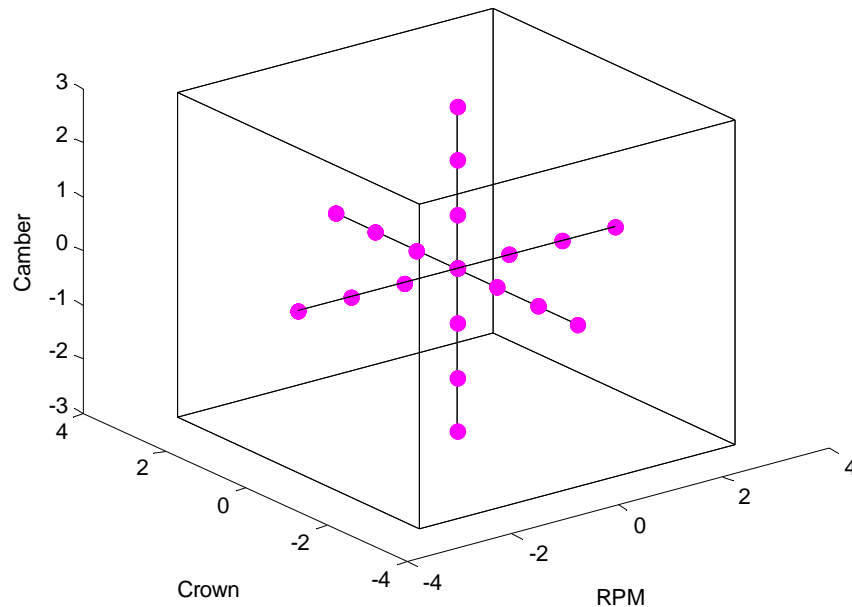


Figure 8. Three parameters, two of type Solo, one of type Cross

This is a surprising result, in that it looks exactly like the complete Solo situation shown in Figure 5. The key is in the definition of a Cross parameter—consider it again:

“Cross variation of a parameter means that the parameter is varied through all possible permutations of the *other cross parameters*.”

The clause “...of the *other cross parameters*” is a critical distinction. Since the variation of a Cross parameter is defined by having the other Cross parameters vary at the same time, there must be at least two Cross parameters for the Cross/Solo distinction to matter. In other words, a parameter can only be Cross *relative* to one or more other parameters. If there are no other parameters of type Cross, the parameter behaves as if the type was Solo.



### III.C. The Global Parameter Type

When choosing a parameter type, besides the Solo and Cross types, you also have the option of making your parameter of type “Global”. Making a parameter of type Global simply means that the behavior of that parameter will be controlled by the settings in the Global Freq/Type box, highlighted in Figure 9.

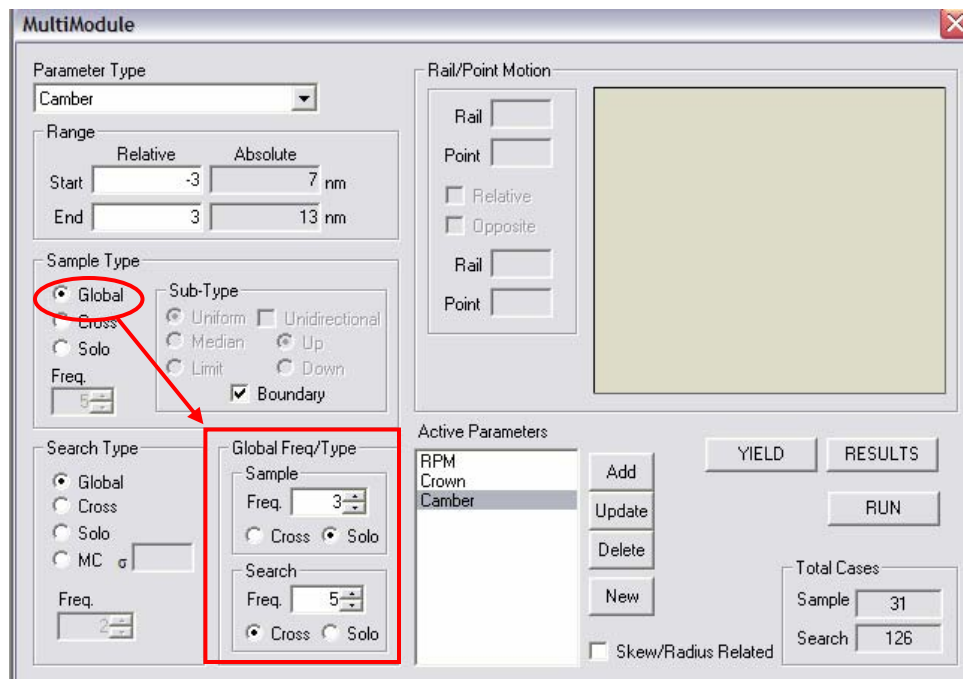


Figure 9. Global Freq/Type box

Notice that this box has two areas, one for Sample and one for Search. These areas are separate because the Sample options and Search options are completely independent of each other.

Making a parameter of type Global is helpful when you want to change several parameters at once. Since all parameters of type Global are controlled by the options in the Global Freq/Type box, changing the frequency or Cross/Solo option in this one place will automatically update the respective settings of all parameters of type Global. This makes it easy to change many options at once. The disadvantage is that you lose some customizability, which will be explained later.

### III.D. The Monte Carlo Search Option

One of the differences between the Sample Options and Search Options is the availability of selecting a “Monte Carlo” Search Type. The radiobutton that corresponds to this choice is labeled “MC” and is highlighted in Figure 10.

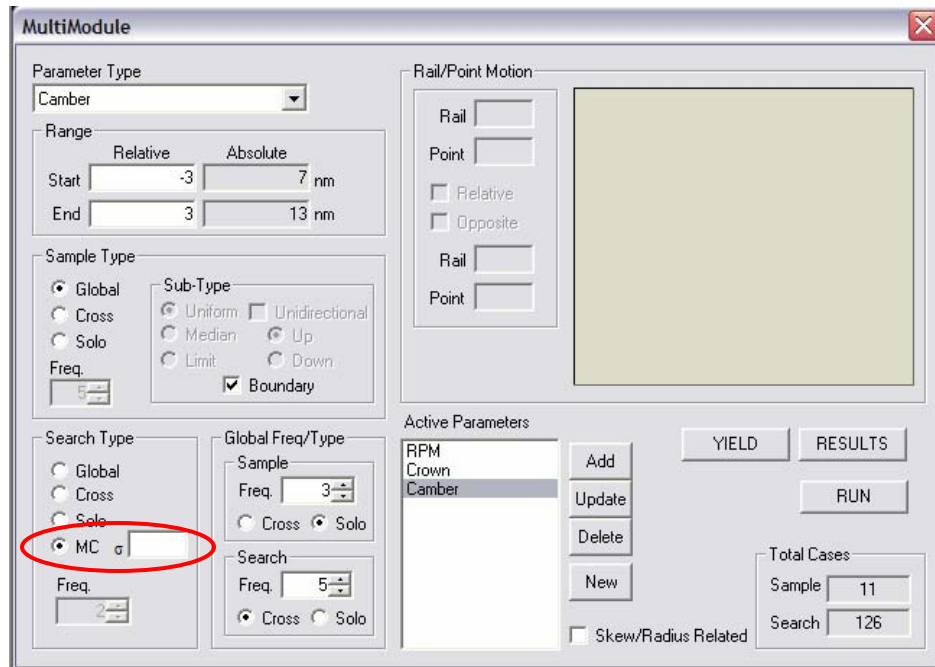


Figure 10. Monte Carlo Search Type Option

When you select a Search Type of “MC”, you are telling Multimodule to randomly select values of the current parameter for each of the Search cases. These random values are members of a normal distribution with a user-specified standard deviation given in the box next to the MC radiobutton, and a mean of zero. While the values that Multimodule picks for the current parameter are statistically random, the seed that is used to create the random values is determined by the current input parameters. Thus, two successive runs of a Multimodule set will produce the exact same random parameter values. To get new random parameter values, some of the input parameters must be changed.

When the Search type is “MC”, Multimodule produces a random value for the parameter for each of the Search cases, where the total number of Search cases is determined by the Search options of the other parameters. Thus, a Search type of MC effectively has a

frequency of 1, since it does not vary independently of the other parameters. This will become much clearer with an example.

Let us return to our example of RPM, Crown, and Camber as parameters in our Multimodule set. Let RPM and Crown have a Search type of Global, with Global Search options of Cross and a frequency of 5. Also let Camber have a Search type of MC, with a standard deviation of 0.33333. Figure 11 illustrates the Search cases for this situation.

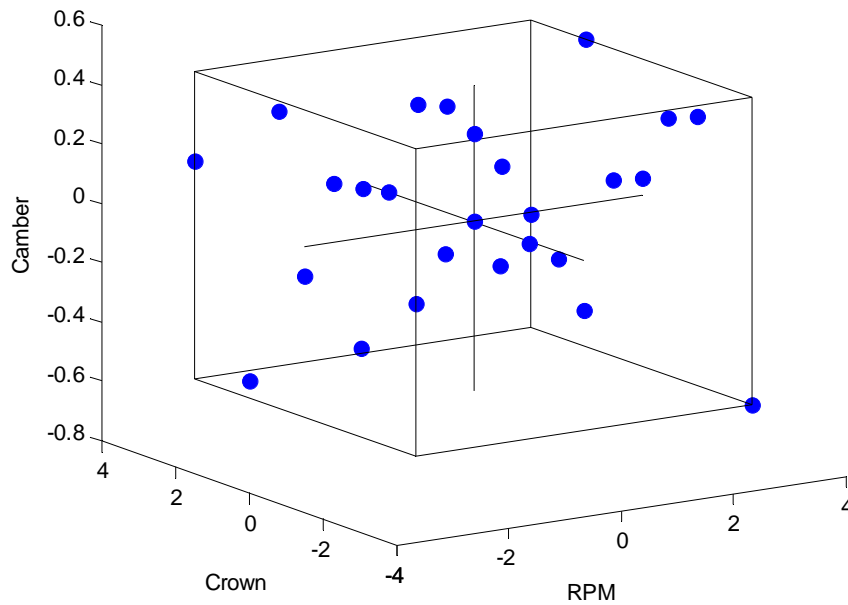


Figure 11. Search cases for one parameter of type MC

The view from this angle makes it hard to see exactly what's happening, so Figure 12 shows the same illustration, but viewed from the "top" so that the Camber axis is eliminated.

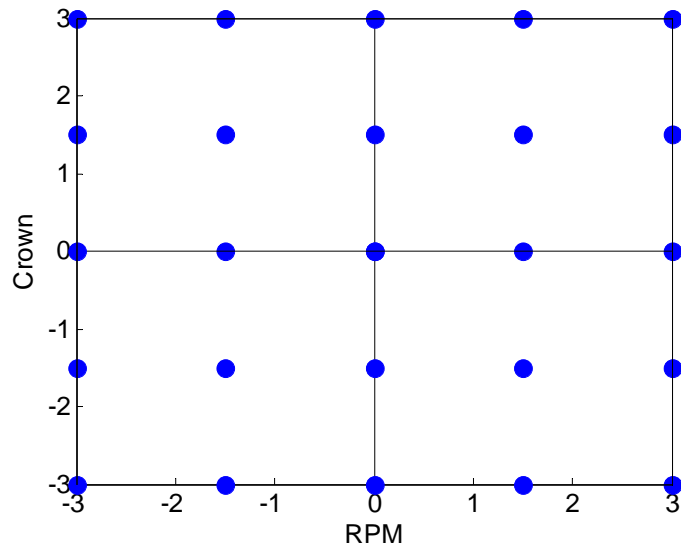


Figure 12. Alternate view of Figure 11, with Camber axis eliminated

Figure 12 shows that the RPM and Crown parameter values are what would be expected for a mutual Search type of Cross, with frequency = 5. However, each of the cases in Figure 12 have a different random value for Camber (which gives the variation along the Camber axis in Figure 11.) Thus, you can “simplify” the current situation as a two parameter study (RPM and Crown), with a Search type of Cross and a frequency of 5, but where each case takes a random value for Camber instead of the base value.

Also, if all parameters have a search type of MC, the total number of search cases must be manually entered in the box in the lower right corner of the Multimodule window. This is because when all parameters are chosen randomly, there is no “simplified” situation that will determine the number of Search cases. Thus, you must specify the total number of Search cases, and Multimodule will pick random parameter values for each of these cases.

Of particular interest is having a distribution of random parameter values that covers the entire range of the random parameter. Unfortunately, the random number generator for the MC Search option only considers the base value for the parameter and the given standard deviation when producing the random values. Thus, the random values for the

parameter will not reflect the range specified when you first pick the parameter. In the previous situation, Camber was made a random parameter with a base value of zero and a standard deviation of 0.33333, but it had a range of +/- 3. The random variation of Camber did not take up the entire range, as the edges of the range occur at +/- 9 standard deviations, or 18 sigma. Statistically, this will never be realized. If the random parameter is to take on the entire range, the standard deviation must be adjusted based on the range specified when the parameter was first chosen. If we use a 6 sigma rule to account for 99.7% of the random values, we need to make the standard deviation equal to one-sixth of the full range:

$$\sigma = 1/6 * (\text{full range}),$$

where the “full range” is the difference between the maximum allowed value and the minimum allowed value for the random parameter. For 1000 search cases, statistically only three will fall outside the range specified. Note that this only works for symmetric ranges, since the random parameter values will have a mean equal to the parameter’s base value.

### III.E. Subtypes

The Cross and Solo parameter types have a few Sub-Type options that further increase the customizability of the Sample spaces. These options are available in the Sub-Type Options box, seen in Figure 13, and are explained in detail below.

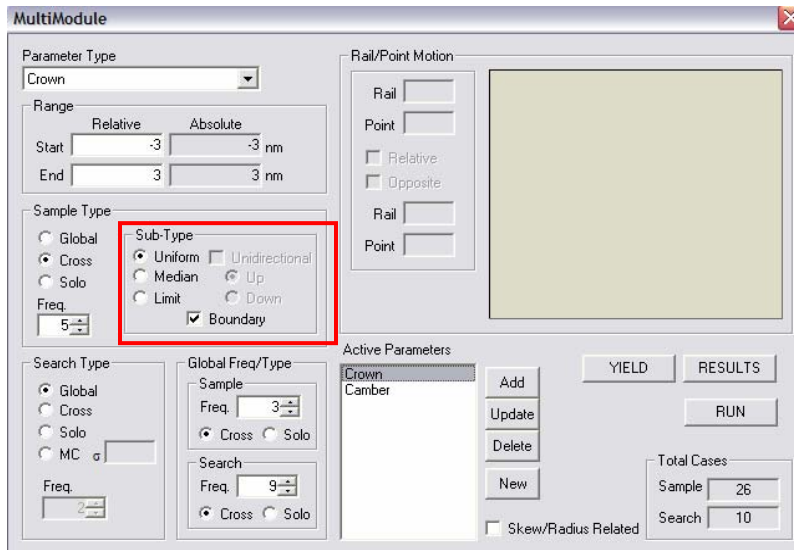


Figure 13. Sub-Type Options Box

#### III.E.i. Uniform and the Boundary Option

The first sub-type is Uniform. Uniform is the default sub-type as well as the implied sub-type when the sub-type options are not available (Global and MC). Uniform means that the cases are spaced evenly through the range of the parameter, as one would expect. Figure 14 illustrates a two-parameter study where both parameters are of subtype Uniform, with a frequency of 5 and range +/- 3.

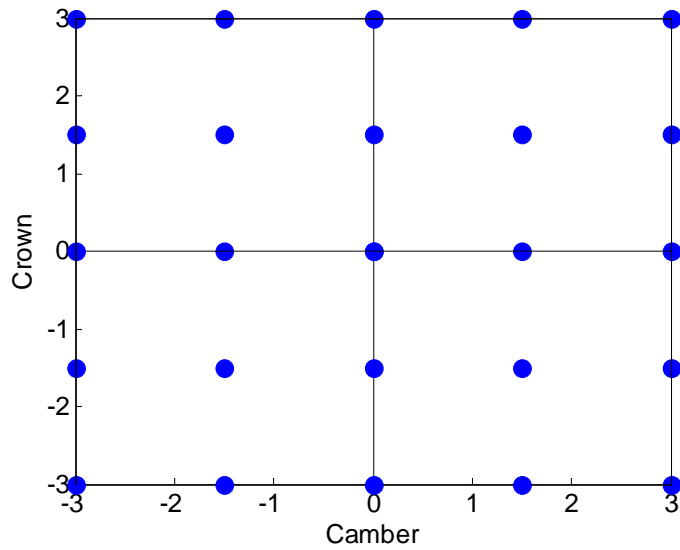


Figure 14. Subtype Uniform

Intimately related to the Uniform subtype is the Boundary option. When Boundary is checked (default), it means that the uniform distribution of cases should occur in a way that the most extreme cases fall precisely on the maximum and minimum of the range specified earlier for the current parameter. Figure 14 has Boundary checked, and Figure 15 shows the exact same options, except for Boundary being unchecked for the Crown parameter.

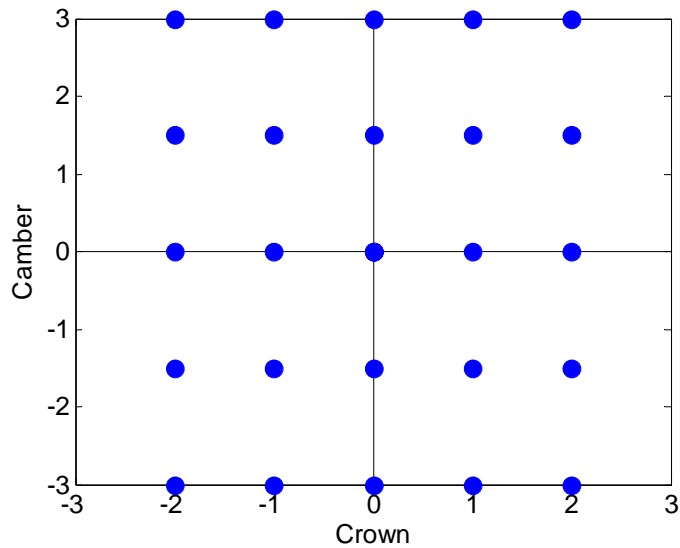


Figure 15. Subtype Uniform with Boundary unchecked for Crown, checked for Camber

In comparing Figure 15 to Figure 14, it is clear that the Crown parameter values are more closely spaced for Figure 15, and they do not extend to the minimum and maximum of the range (+/- 3). A formula can easily be derived to calculate the spacing between parameter values for the Uniform subtype for Boundary checked and unchecked:

with Boundary checked:  $\text{spacing} = (\text{full range}) / (\text{frequency} - 1)$

with Boundary unchecked:  $\text{spacing} = (\text{full range}) / (\text{frequency} + 1)$

The Boundary option is always available, except when the subtype is Median or Limit, as explained next.

### III.E.ii. Median and Limit Options

The other main subtype options are Median and Limit. As before with the Uniform subtype, their names are suggestive of their function. In the Median subtype the parameter only takes on the median values, i.e. halfway between the base value and the maximum and halfway between the base value and the minimum. Similarly, in the Limit subtype the parameter only takes on values at the maximum and minimum. Figure 16 illustrates the Median and Limit subtypes.

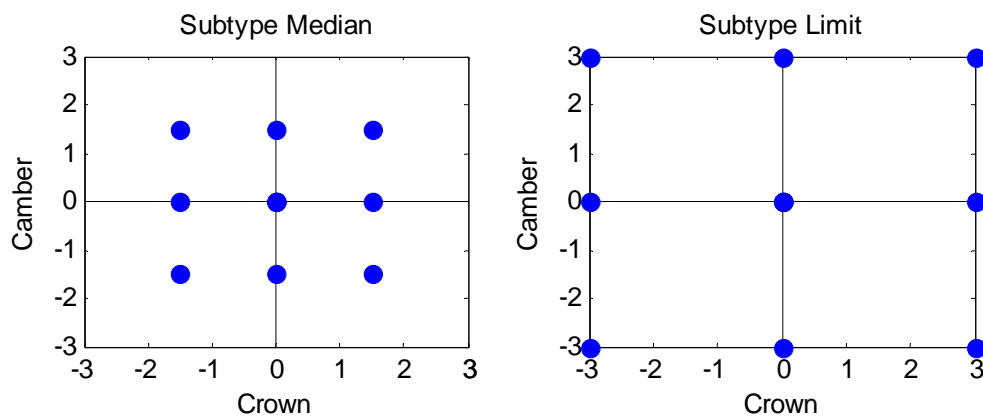


Figure 16. Illustration of Subtypes Median and Limit

The Unidirectional option is available for both Median and Limit subtypes, with options of Unidirectional Up and Unidirectional Down. Again, these options behave just as expected in that the Median or Limit points are only taken in the positive (up) or negative



(down) increment. Figure 17 illustrates the Unidirectional Up option applied to the Crown parameter in Figure 16.

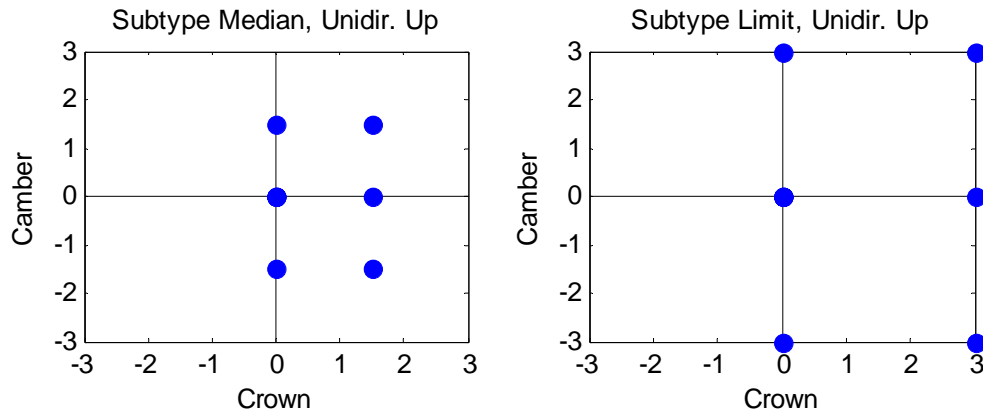


Figure 17. Illustration of Unidirectional Up option for Median and Limit Subtypes

### III.F. Rail/Point Motion

The top right corner of Multimodule contains the Rail/Point Motion options. These options are used when the current parameter is Rail Point X/Y or Rail X/Y. In Figure 18, the current parameter is Rail X.

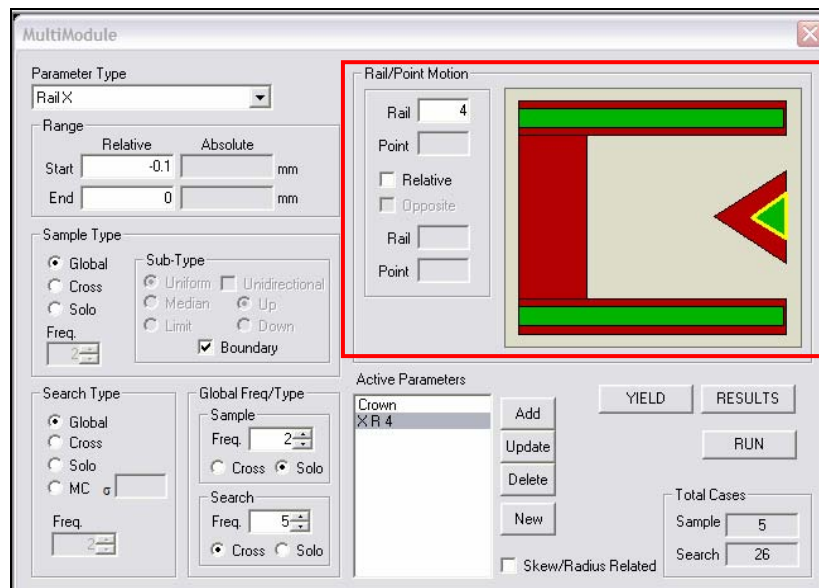


Figure 18. Rail/Point Motion Options

As with other the options, we enter the Range in the boxes below the Parameter Type selection box. Notice that the units for Rail/Point motion are in mm, and adjust the range accordingly. In Figure 18, the slider is only 1.25mm long, so a range of 0.1mm is a large range. Also notice that Multimodule does not show the absolute range of the parameter because it is not really defined for a feature made up of many points. If the parameter were Rail *Point* X/Y, then the absolute range values would appear.

After entering the range and other options for the parameter, one must select which rail or which point to manipulate. In Figure 18 above, rail number 4 is selected as our current parameter. This can be done by entering the rail number in the “Rail” box or by double-clicking on the slider picture. Notice that the rail is highlighted in the slider picture to the right. This helps to remember the numbering of the slider rails or points. Additionally, if the parameter Rail *Point* X/Y is selected, the point on that rail to be manipulated must be selected by entering the point number in the “Point” box. The Rail *Point* will then be highlighted similar to the Rail.

To only move a single Rail or Rail *Point* around, leave the “Relative” box unchecked. This box is used to tie the motion of one Rail or Rail *Point* to the motion of another Rail or Rail *Point*. The use of the Relative box is best seen with an example. Figure 19 shows an example slider with the center pads circled.

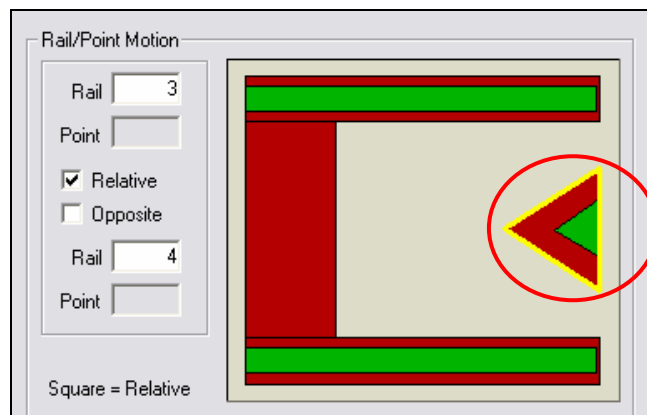


Figure 19. Example Slider

The Relative checkbox is useful if we want two or more rails to move together. Let's say that we want the center trailing pads (circled) to move towards the center of the slider as one feature. Since the Rail X/Y parameter option only lets us select one rail at a time, we must use the Relative checkbox to tie the motion of the two rails together. We have already shown how to get the upper rail (number 4) to move in Figure 18. We now want to make the lower rail (number 3, highlighted) move with the upper rail. We add a new parameter of type Rail X and enter the rail number to be moved (3). We then check the Relative checkbox and enter the rail number (4) to which we want to tie the current rail's motion. Now whenever rail 4 moves, rail 3 will move with it, moving the entire trailing pad around together instead of one rail at a time. You can use the Relative checkbox to "daisy-chain" many rails together by making rail C follow rail B, and rail B follow rail A, etc. Then rails A, B, and C will all move together.

The Opposite checkbox is used if you want one rail or rail point to move in the opposite direction of another rail or rail point. This option functions as explained in the previous paragraph, except when rail 4 moves in the positive direction, rail 3 will move in the negative direction.

### III.G. The Yield and Results Buttons

The Yield and Results buttons provide additional ways to interpret and view the results. The Yield button allows you to count the cases that fail certain user-specified criteria while the Results button lets you view the results in tabular or Histogram form.

#### III.G.i. Yield Limits

As mentioned earlier, the Yield button brings up the Yield Limits box, where you can specify maximum and minimum allowable limits for the slider's performance. The Yield Limits box allows you to specify ranges for the Nominal FH, Pitch, Roll, Min FH, and FH at the four points of interest (specified earlier under the General tab). See Figure 20 for the Yield Limits box.

The image shows a software dialog box titled "Yield Limits". It is organized into two columns and four rows of control panels. Each panel includes a checked checkbox, a "Low Limit" field with a numerical value (all are "0") and a unit, and a "High Limit" field with a numerical value (all are "0") and a unit. The units are "nm" for fly height and "μrad" for pitch and roll. The right column panels also include a range specification in brackets. At the bottom center are "OK" and "Cancel" buttons.

Section	Low Limit	High Limit	Unit
Nominal Fly Height	0	0	nm
Pitch	0	0	μrad
Roll	0	0	μrad
Min FH	0	0	nm
FH POI1 [0.805, 0.33]	0	0	nm
FH POI2 [0.805, 0.44]	0	0	nm
FH POI3 [0.15, 0.6]	0	0	nm
FH POI4 [0.15, 0.05]	0	0	nm

Figure 20. The Yield Limits Box

The Yield Limits box is best understood with an example. Say we are designing a slider to fly at 10nm with a range of +/- 2nm. This means we want our slider to fly between 8 and 12nm. If we are studying how manufacturing tolerances affect our FH, we want to count the number of cases that fall outside this range. Thus, we bring up the Yield Limits box and enter 8nm for the “Low Limit” and 12nm for the “High Limit” for the Min FH. Say, for another reason, we want to make sure the pitch stays below 250 microradians. We then enter 250 microradians for the “High Limit” for the Pitch. After we enter all the Yield Limits, we must make sure that the check boxes are selected for the limits we have specified and we must uncheck the boxes for all the other limits. We then click “OK” and run the simulation.

After the simulation is finished, we use the Results button to scroll all the way to the right. A new column has appeared called “Yield Failures” that flags when a case has failed the Yield criteria specified in the Yield Limits box. We also have the total number of cases that pass the Yield criteria (out of the total number of cases), as well as a percentage. See Figure 21.

n Loc (mm)	Pos Force (g)	Neg Force (g)	Con Force (g)	X Shear (g)	Y Shear (g)	Yield Failures
.25,0.416)	4.0362	-2.5363	0	0.068276	-0.000433	0
.25,0.416)	4.0398	-2.5388	0	0.068299	-0.000432	0
.25,0.416)	4.0362	-2.5363	0	0.068276	-0.000433	0
.25,0.416)	4.0335	-2.5337	0	0.068255	-0.000435	0
.25,0.411)	4.0353	-2.5349	0	0.068261	-0.000433	1
.25,0.416)	4.0362	-2.5363	0	0.068276	-0.000433	0
.25,0.426)	4.0371	-2.5377	0	0.06829	-0.000434	1
.25,0.416)	4.0362	-2.5363	0	0.068276	-0.000433	0

Figure 21. Yield Information in the Results Box

The new column “Yield Failures” shows a 0 when a case passes the Yield criteria and a 1 when a case fails. As can be seen in Figure 21, two cases failed the Yield criteria out of 7 cases (remember the first row is the base case, which is not counted) for a total yield of 5/7, or 71.4%. The Yield Failures column and Yield percentage is given for both the Sample and Search results.

### III.G.ii. The Results Button

The Results button has previously been seen, but we will now explain it in full. Recall that the Results button can show the results (in relative or absolute form) and yield information. It can also be used to copy the results to the clipboard for exporting to Excel or Matlab, export the case files to a folder, or view the Search results in histogram form.

To copy the results to the clipboard, simply click the “Copy to Clipboard” button and the results for every case will copy to the clipboard (Sample and Search cases must be copied separately). The titles of the columns are also copied to remind you what each column represents.

The “Export Selected” and “Export All” buttons export the run files of “rail.dat”, “result.dat”, “run.dat”, “trigrd.dat”, “x.dat”, and “y.dat” for either the selected case(s) or for all the cases. These files will be put into a new folder titled “*slidername\_Export\_Case\_#*” under the directory of the slider, where *slidername* is the filename of the current slider and # is the case number from the far left column. Using the “Export All” button creates a separate folder for each case and “Export Selected” will only create folders for the selected cases.

Figure 22 shows the Histograms window (only available for the Search cases). The drop-down box in the upper left corner allows you to choose which result to view (everything that appears in the Results window in tabular form can be viewed in histogram form here). The slider to the right of the drop-down box allows you to select how many bars to use in the histogram, from 2 to 15 in steps of one, with a default of 8 bars. In the upper right corner, you can also see the standard deviation and mean for the selected result.

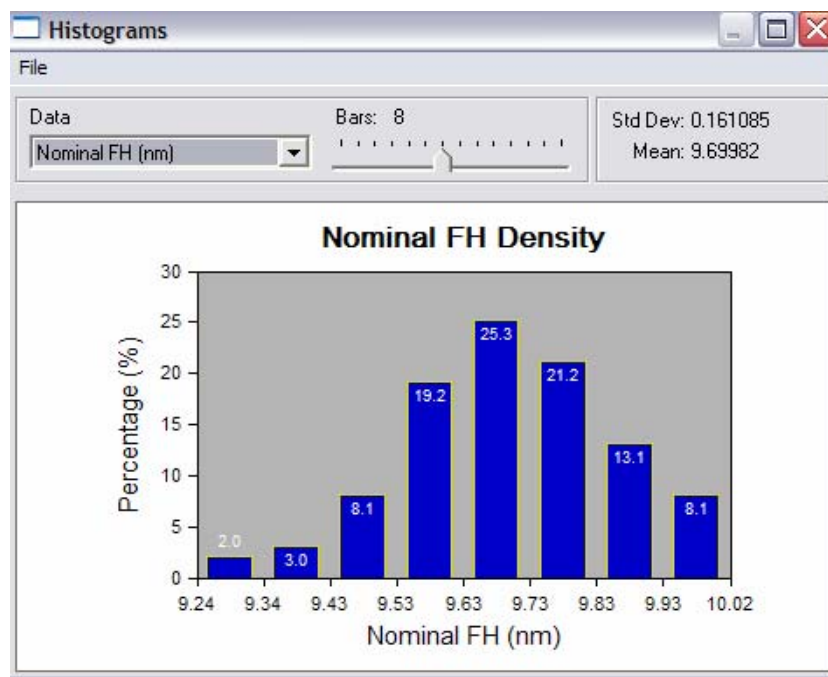


Figure 22. Histograms window

#### IV. Multimodule Examples

This section of the Multimodule User's Manual gives a few complete examples of a Multimodule study, focusing on the determination of parameter dependence or independence. We will use the NSIC slider for our examples. The first example investigates the effects of Crown and Camber on the Nominal FH of the example slider. The second example investigates the effect of changing the location of the central trailing pad.

##### IV.A. Independent Parameters: Crown and Camber

The Crown and Camber of the slider are considered independent parameters in that they both affect the slider performance independently of the value of the other parameter. From the discussion in section III.B we know that since these parameters are independent of each other, the Sample type should be Solo. However, if we didn't know that Crown and Camber are independent of each other *a priori*, we would have to investigate the relationship to determine which Sample type should be used.

The most basic way to determine if a parameter set is independent or dependent is to perform a series of Multimodule sets, first with a Sample type of Solo and then with a Sample type of Cross. Comparing the errors between the Search results for the two sets will show which Sample type is appropriate for the current parameters.

In order to compare the errors between a Solo and Cross Sample type, we must first find the "true" values by making the Sample cases coincide with the Search cases. This will give us an exact value at each Search case with which to compare the interpolated results (when the Sample cases and Search cases do not coincide.)

For this example, we are investigating the influence of Crown and Camber of the flying characteristics of the NSIC slider. The base value for Crown is 25.4nm and the base value for Camber is 2.5nm for the NSIC slider. We will arbitrarily choose to vary the Crown by +/- 5nm, or about 20%, and we will vary the Camber by +/- 2.5nm, or 100%.

The resulting Crown values will range between 20.4nm and 30.4nm while the Camber ranges between 0nm and 5nm. Our first Multimodule set has the following options:

Table 1. Crown/Camber Example Options, Set 1

Parameter	Range	Sample				Search		
		Type	Freq.	Total Cases	Notes	Type	Freq.	Total Cases
Crown	+/- 5nm	Cross	10	101	uniform, boundary	Cross	10	101
Camber	+/- 2.5nm	Cross	10	101	uniform, boundary	Cross	10	101

These options represent a Sample Space that coincides exactly with the Search Space. Thus, the results will be accurate to within the accuracy of the static air bearing solver. After running the first set, we export the results table and plot the Nominal FH versus Crown and Camber in Figure 23.

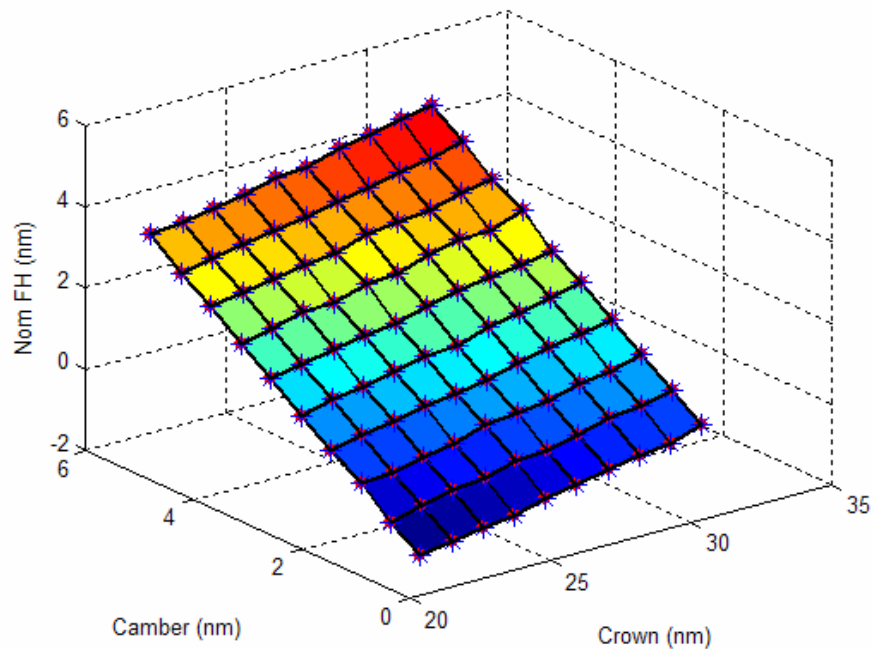


Figure 23. Results—Nom FH for First Set (True Results)

In Figure 23, one of the results columns, Min FH, has been exported and plotted in a surface plot to show the true relationship between Crown, Camber, and Nom FH. The blue asterisks represent the Sample Cases and the red circles represent the Search Cases, but they coincide in this set (making them hard to see). In addition, black lines connect



the points of the Sample cases to aid in visualization. This convention will also be used for subsequent examples. These values for Nom FH will be used as the true values for comparison between the other sets.

The first test set will be of Sample type Solo, with a frequency of 3. The Search options will remain unchanged. Table 2 gives the options for this set.

Table 2. Crown/Camber Example Options, Set 2

Parameter	Range	Sample				Search		
		Type	Freq.	Total Cases	Notes	Type	Freq.	Total Cases
Crown	+/- 5nm	Solo	3	8	uniform, boundary	Cross	10	101
Camber	+/- 2.5nm	Solo	3	8	uniform, boundary	Cross	10	101

Notice this time that there are only 8 Sample cases instead of the 101 in the previous set. This will speed up the simulation dramatically. Before we show the Nom FH results, Figure 24 shows the Sample Space and Search Space.

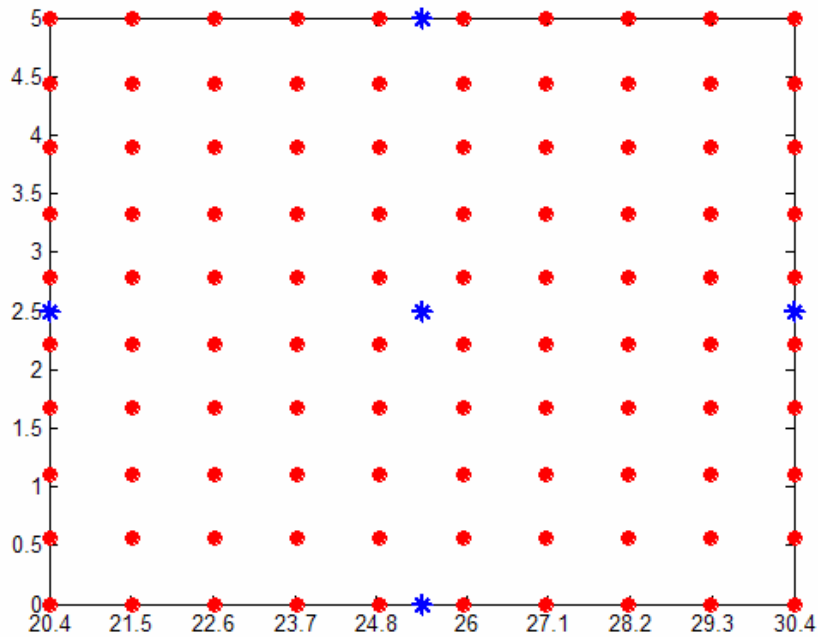


Figure 24. Sample Space and Search Space for Set 2

Since the Search Case results (red circles) are interpolated from the Sample Case results (blue asterisks), we will expect the largest error to occur in the corners, where the Search Cases are farthest away from the Sample Cases. Similarly, we expect the smallest errors to occur near the axes of the Sample Cases.

Figure 25 plots the Nom FH versus Crown and Camber for the second set.

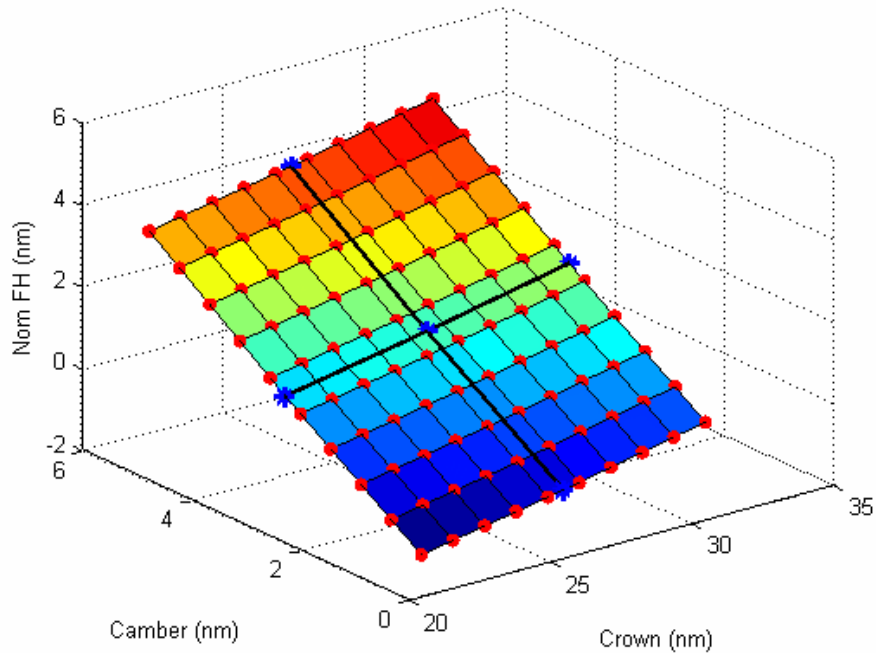


Figure 25. Results—Nom FH for Second Set (Solo)

In comparing Figure 25 to the true results in Figure 23, it is immediately apparent that the general shape of the Nom FH surface has been captured. To see just how well the relationship has been captured, we calculate the absolute error by subtracting the true Nom FH values from the interpolated values at each of the Search Cases. Figure 26 shows the error for the second set.

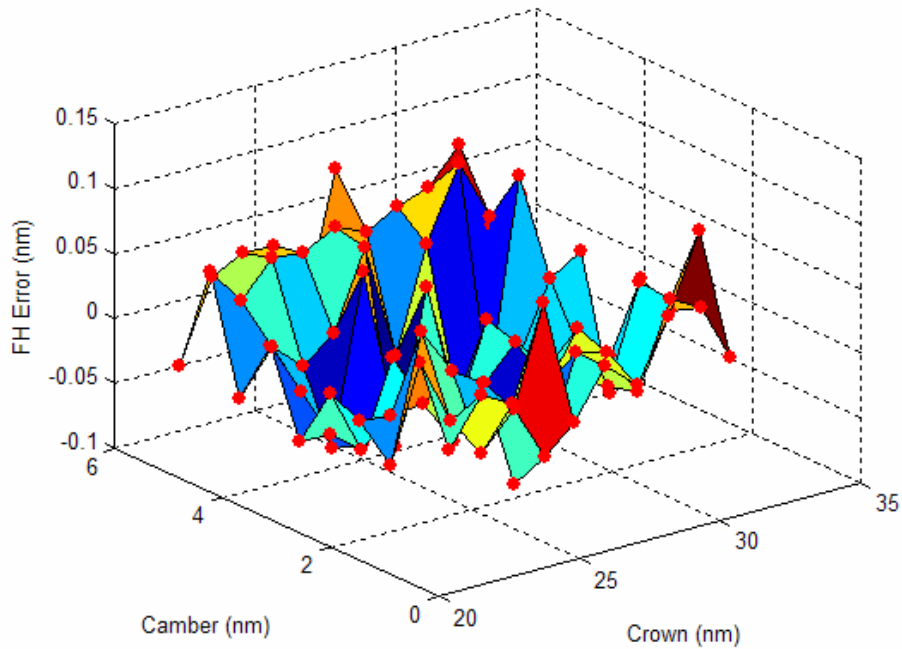


Figure 26. Absolute Error—Second Set (Solo)

It is immediately clear that the absolute errors in Nom FH for the second set are on the order of 0.1nm. Since the NSIC slider tends to fly at heights of a few nm, we can say that the errors in the Search Case interpolations are low. We can quantify the “average” absolute error by taking the RMS of the errors:

$$RMS = \sqrt{\frac{x_1^2 + x_2^2 + x_3^2 + x_4^2 + \dots + x_n^2}{n}}$$

Here, the  $x$  values are the errors at each of the Search Cases and  $n$ , the total number of points, is 100. For the second set, the RMS error is 0.049nm. If the Nom FH is 2nm, a Search Space specified by the options in Table 2 introduces around 2.5% error in Nom FH.

We can compare these results to those that come from a Sample type of Cross. If we do so, we will arrive at a RMS error of 0.045nm for the Cross simulation. Since the Solo simulation had a RMS error of only 0.004nm larger, we can say with confidence that a Sample type of Cross is not really necessary.

Based on the previous analysis, we see that crown and camber are independent parameters and should always have Sample types of Solo, simply because the errors are very small, even with a low frequency (we used a frequency of three here) and a Sample type of Solo. However, we came to this conclusion using a full Sample Space to find the true values. If performing a full analysis is too computationally expensive or time-consuming we need to be able to reach the same conclusion without referring to the true Nom FH values at each of the Search Cases. We can do this by measuring how much the Search Case results change when we change the Sample type from Solo to Cross. If the results change drastically that suggests that the parameters are dependent on each other and should have a Search type of Cross. However, if the results do not change drastically we can infer that the parameters may be independent. Making conclusions of this type should be influenced by more than one comparison at a few different frequencies. If a parameter set is truly dependent, a Sample type of Solo will never sufficiently capture the relationship, no matter how high the sample frequency.

To illustrate how to determine dependence without performing a full “true value” analysis, we ran a Multimodule set with the same frequency as before (3), but with a Sample type of Cross. The parameter choices are given in Table 3.

Table 3. Crown/Camber Example Options, Set 3

Parameter	Range	Sample				Search		
		Type	Freq.	Total Cases	Notes	Type	Freq.	Total Cases
Crown	+/- 5nm	Cross	3	10	uniform, boundary	Cross	10	101
Camber	+/- 2.5nm	Cross	3	10	uniform, boundary	Cross	10	101

With these options, the Nom FH is plotted in Figure 27.

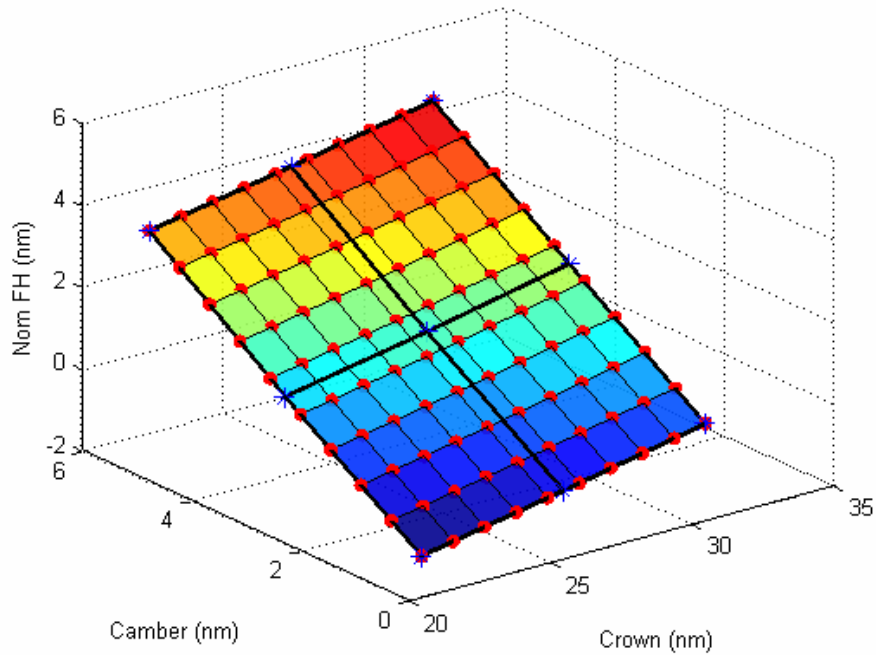


Figure 27. Results—Nom FH for Third Set (Cross)

Next we plot the error, but instead of comparing the Nom FH from the third set with the true values provided by the first set, we compare the third set (sample type Cross) to the second set (sample type Solo). This will tell us how much the Nom FH values change when moving from a Sample type of Solo to a Sample type of Cross. If this change is small it lends support to the conclusion that Crown and Camber are independent parameters, as the change to Cross does not significantly change the results. The error plot with respect to the second set is given in Figure 28.

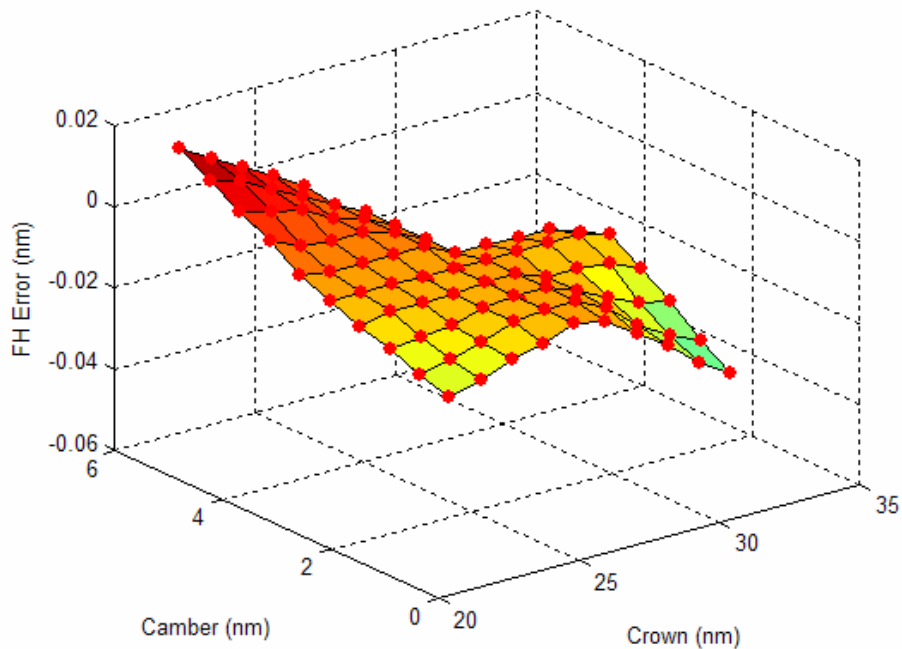


Figure 28. Absolute Error—Third Set to Second Set

The RMS change in Nom FH for the third set with respect to the second set, calculated according to the formula given earlier, is 0.013nm, which is a very small difference in FH between the two sets.

If we repeat the process of measuring the improvement that results from changing the type from Solo to Cross, but with a frequency of 5, we arrive at a RMS change of 0.030nm. Similarly, for a frequency of 10, we arrive at a RMS change of 0.049nm. While the RMS change is getting larger for higher frequencies, it still remains very low. Taking all three of these Solo to Cross comparisons together, we can say with confidence that Crown and Camber are independent of each other with respect to the Nominal FH. It is expected that the other results will obey a similar relationship.

The calculations and comparisons given above show the relationship between two independent parameters, Crown and Camber. We have shown how to determine this relationship both with respect to the true values (full simulation) and by measuring the improvement between Solo and Cross.

#### IV.B. Dependent Parameters: Trailing Pad Position

In contrast to the independent relationship between Crown and Camber seen above, we expect rail position changes to be dependent on each other because of the very complex fluid dynamics interaction related to the geometry of the slider. In this exercise we vary the vertical and horizontal position of the central trailing pad on the NSIC slider. Figure 29 shows in which directions the trailing pad will be moved.

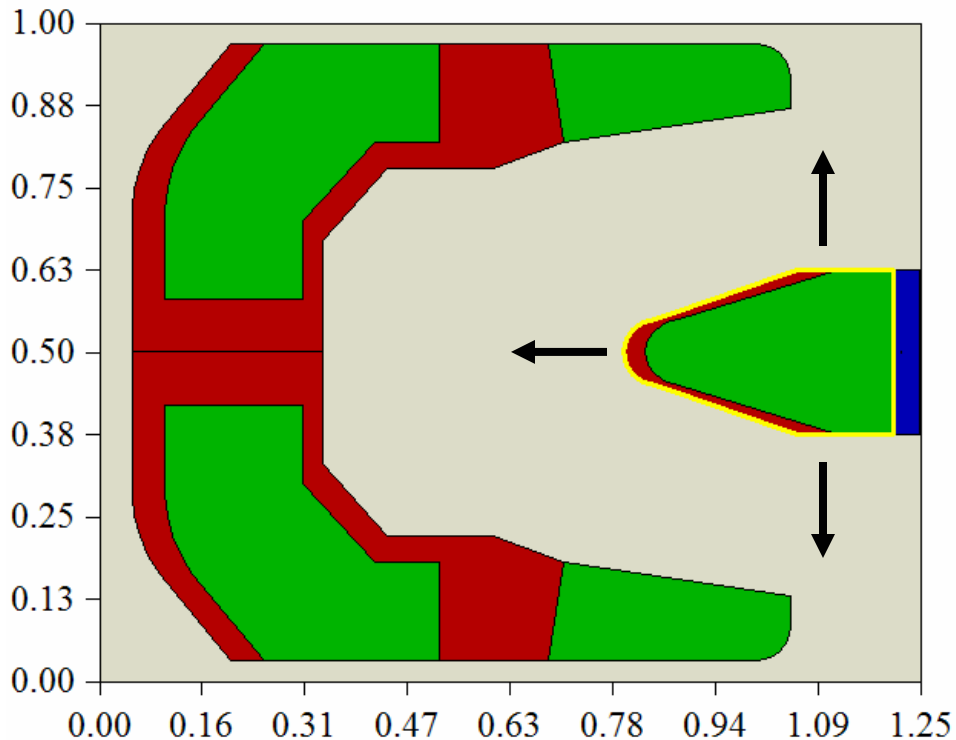


Figure 29. Movement of Trailing Pad

The actual variations are  $\pm 0.2\text{nm}$  in the y-direction and  $-0.2\text{nm}$  in the x-direction. The tips of the arrows in Figure 29 show the approximate position of the corresponding edges at the extremes of the displacement.

The trailing pad on the NSIC slider is composed of three different rails. Since we want them all to move as a single unit we have to use the “relative” option under Rail/Point Motion. In this way, we specify the motion of rail number 1 and force the other rails (numbers 2 and 6) to follow rail number 1 wherever it goes (see section III.F.) When we do this, we end up with six Active Parameters (3 rails x 2 directions = 6). However, only

two of these Parameters actually control the motion, as the other Parameters are “relative”.

Following the exact same procedure as we did for the Crown and Camber example, we calculate the true results by making the Sample and Search cases coincide. The settings are given in Table 4.

Table 4. Trailing Pad Motion Settings, Set 1

Parameter	Range	Sample				Search		
		Type	Freq.	Total Cases	Notes	Type	Freq.	Total Cases
Rail X	- 0.2nm	Cross	10	101	uniform, boundary	Cross	10	101
Rail Y	+/- 0.2nm	Cross	10	101	uniform, boundary	Cross	10	101

The Nominal FH for this Multimodule set is plotted in Figure 30.

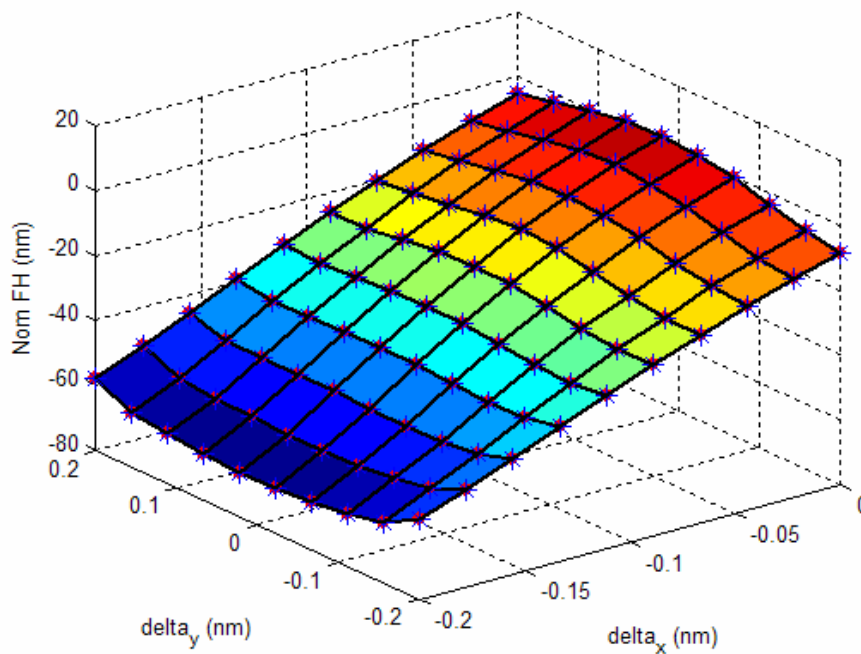


Figure 30. Results—Nominal FH for First Rail Set (True Results)

Figure 30 shows that the axes we are changing are labeled “delta<sub>x</sub>” and “delta<sub>y</sub>”. This is because the actual Rail X and Rail Y positions are meaningless, as there is no single



value that characterizes the position of a rail. Thus, we are dealing with changes in the x- and y-positions of the rails.

The first test set is a Sample type of Solo, with a frequency of 3, just as before. Table 5 gives the Multimodule options for this set.

Table 5. Rail Example Options, Set 2

Parameter	Range	Sample				Search		
		Type	Freq.	Total Cases	Notes	Type	Freq.	Total Cases
Rail X	-0.2nm	Solo	3	8	uniform, boundary	Cross	10	101
Rail Y	+/- 0.2nm	Solo	3	8	uniform, boundary	Cross	10	101

The Nom FH results are plotted in Figure 31.

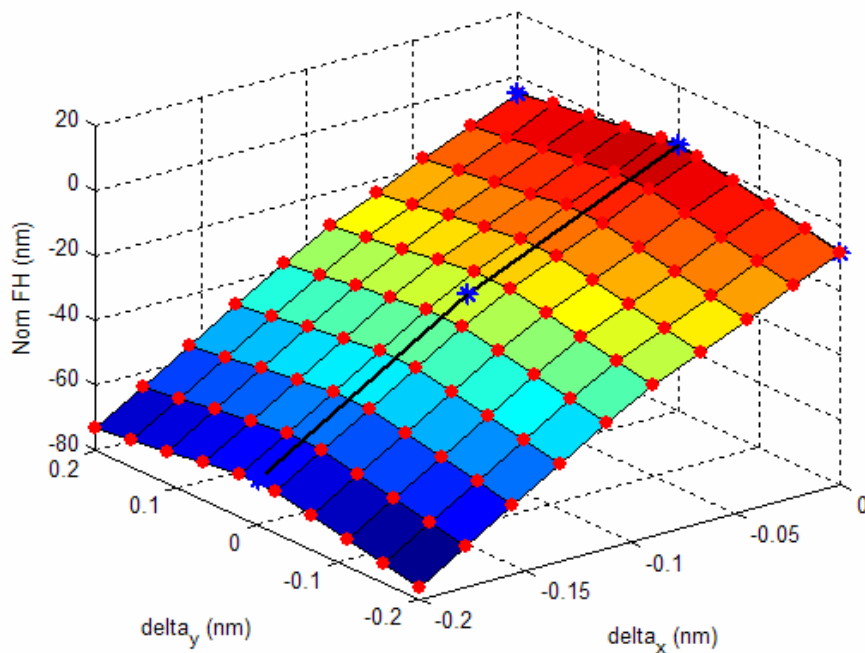


Figure 31. Results--Nom FH for the Second Rail Set (Solo)

The error for this set is plotted in Figure 32.

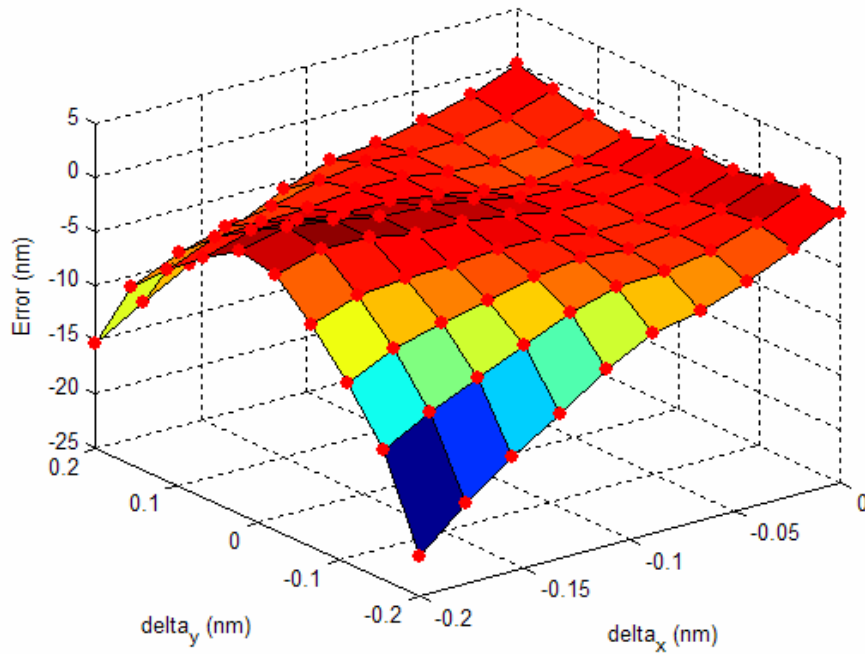


Figure 32. Absolute Error – Second Rail Set (Solo)

The large errors at the corners of Figure 32 are due to the curvature of the result surface. The true surface curves upwards at  $\text{delta}_x = -0.2\text{nm}$  (see Figure 30), while the surface for this case curves downwards at  $\text{delta}_x = -0.2\text{nm}$ . The RMS error associated with this set is 5.554nm. This is clearly a significant error.

We now define a third set with the same frequency as before (3) but a Sample type of Cross. Table 6 gives the Multimodule options for the third set.

Table 6. Rail Example Options, Set 3

Parameter	Range	Sample				Search		
		Type	Freq.	Total Cases	Notes	Type	Freq.	Total Cases
Rail X	-0.2nm	Cross	3	10	uniform, boundary	Cross	10	101
Rail Y	+/- 0.2nm	Cross	3	10	uniform, boundary	Cross	10	101

The results for this set are given in Figure 33.

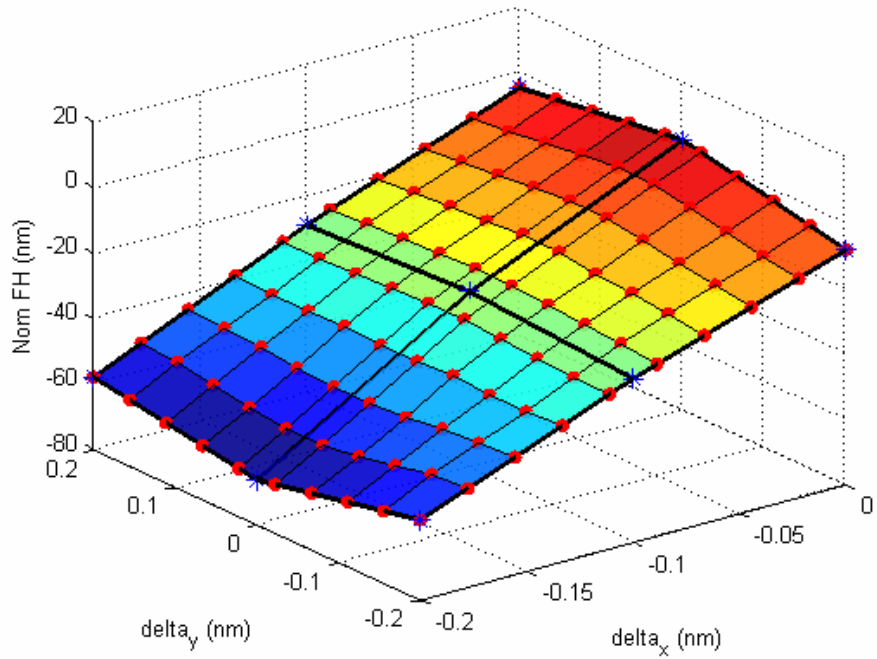


Figure 33. Results--Nom FH for the Third Rail Set (Cross)

The error for this set is plotted in Figure 34.

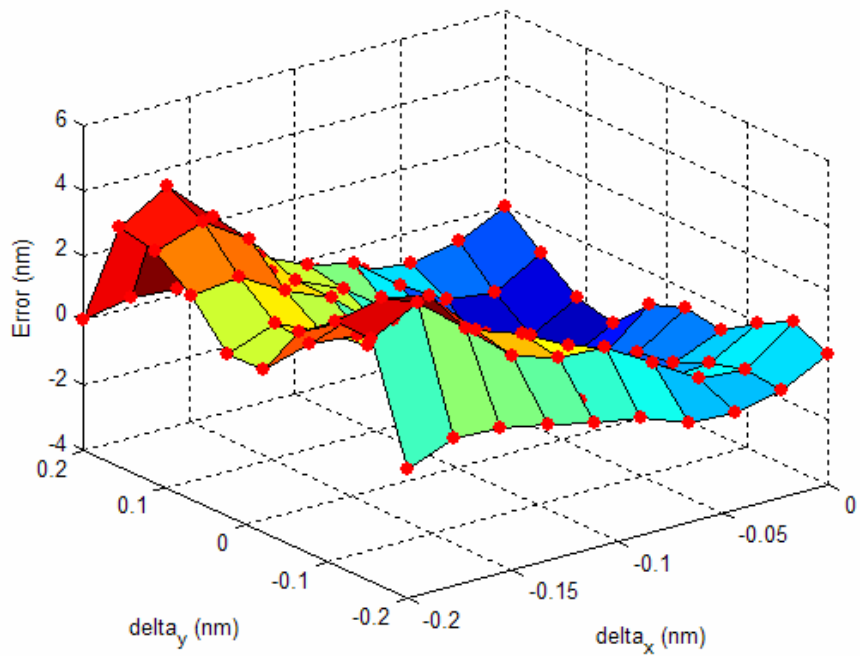


Figure 34. Absolute Error – Third Rail Set (Cross)

This error is less at the corners due to the Cross option placing Sample cases in the corners. The RMS error for the third set is 1.814nm.

If we repeat the above two sets with a frequency of 5, we get RMS errors of 5.258nm for the Solo set and 0.740nm for the Cross set. Table 7 summarizes the results.

Table 7. Rails Example RMS Errors

	Solo	Cross
freq = 3	5.554nm	1.814nm
freq = 5	5.258nm	0.740nm

Table 7 shows that the RMS errors are much better when the Sample type is Cross than when the Sample type is Solo. This leads us to conclude that the rail motions described in this example are dependent on each other with respect to the Nominal FH.

Again, we came to this conclusion by using the true results from a full simulation. If the full simulation is too computationally expensive, we resort to the comparison method described in the previous example.

Table 8 summarizes the results from such a method, where we see how much the results vary when we change from a Sample type of Solo to Cross, for frequencies of three, five, and ten.

Table 8. Summary of Comparison Method for Rail Example

	RMS change
freq = 3	6.280nm
freq = 5	5.553nm
freq = 10	5.349nm

Table 8 shows that changing the Sample type from Solo to Cross makes a big difference in the Nominal FH results, as 5nm is a significant variation in Nominal FH. With this information, we can conclude that the x- and y-positions of the trailing pad are dependent

on each other with respect to the Nominal FH. Again, we expect the other results to have the same relationship.

We have now given two examples of Multimodule studies, demonstrating the difference between dependent and independent parameters, as well as how to characterize two parameters of unknown relationship. Knowing how to characterize the relationship between parameters is very important when using Multimodule. While the examples here have all used only two parameters one can easily construct a Multimodule set with multiple complex interactions. Viewing the entire set as a combination of many two-parameter interactions will allow the user to choose the right options for each combination.

## V. Summary

This User's Guide began with a short discussion of the capabilities of Multimodule, focusing on its usefulness in slider design. Second, the relationship between Multimodule and the standard CMLAir static air bearing code was given in graphical form. Next, a quick walkthrough was given illustrating the steps to take when using Multimodule, along with a few examples of how Multimodule might be of use. The bulk of this User's Guide was dedicated to the detailed explanation of each option available to the user. Emphasis was placed on the reasons behind the options and why they might be useful. Last, two detailed examples were shown focusing on the determination of parameter dependence or independence. It is hoped that this User's Guide will give users of the CMLAir software a better understanding of Multimodule and how it may be useful in their design and research activities.