Dimensions

The sweep above is the simplest form of *dimension*, a dimension is a collection of one or more *sub-sweeps* which vary parameters in parallel. For instance, if we were being diligent then we should vary centre of gravity height in parallel with car mass if we wish to properly represent the addition of fuel. To do this we click on the + *Sub-interpolation* button below the sweep we've already created, this time targeting `car.chassis.zCoG`. You'll notice that although we can add start and end values etc., we cannot vary the number of points independently for sub-interpolations of a sweep - this is because all sub-interpolations of a sweep must have the same number of points. The exploration will create the prescribed number of experiments where each one has both `mCar` and `zCoG` perturbed by corresponding amounts. The ability to combine multiple parameter perturbations into a single dimension enables some very powerful explorations to be defined: varying aero-balance at fixed total CL is a case of varying front and rear lift coefficients in parallel, varying total lift in parallel with drag will reveal the isochronal ratio for a track, and thus the optimal aero development direction.

Star

The sweep above is the simplest form of exploration - a single sweep in a single dimension (albeit one varying multiple parameters). The more general case of a single sweep is a *Star* design exploration; a set of independent sweeps around a common baseline. Each sweep in a star can have a different number of points, and comprise a number of different sub-sweeps. A common example of a star exploration is the Fundamentals Study, whereby various fundamental parameters of the car are varied around a baseline to extract the sensitivities of the car to those parameters. Typically the parameters in a fundamentals study will be gross design parameters such as CoG height, track width, wheelbase, chassis stiffness etc. etc.

Factorial

Whereas a star exploration varies the car along only one dimension at a time, a factorial exploration runs an experiment at every combination of every value in all dimensions. A star design with 3 dimensions, each with 10 points, specifies 30 experiments; a factorial design with the same 3 dimensions of 10 points each will contain 1000 experiments. When a small number of dimensions need to be fully explored, a factorial design can be ideal. For instance exploring the front/rear ride height space might be ideally done with 20 points along each of those two dimensions, leading to a complete set of 400 experiments covering every combination of front and read ride height. As soon as you go much beyond 2 dimensions, factorial explorations become very inefficient. A typical 8-dimensional setup exploration with 10 points on each dimension would require 100,000,000 experiments if it were done factorially, whereas we can show that a Monte-Carlo exploration can capture all of the information in that 8-dimensional space with only 1000 experiments.

Monte-Carlo

So called because it relies on randomness, much like the games of chance in Monte-Carlo's casinos, the Monte-Carlo method chooses random points inside the parameter space. The space is defined much as for the star and factorial designs, by specifying a number of dimensions, each comprising one or more sub-ranges. The difference between the dimensions of a Monte-Carlo design and those of a star or factorial is that in the Monte-Carlo case we do not specify a number of points for each dimension, rather we specify a number of points for the exploration as a whole. The dimensions then do not define where exactly the experiments will be carried out in parameter space, rather they define the bounds of a parameter hypercube within which points will be chosen at random as the locations of experiments.

Metamodels

Because randomly placed points don't lend themselves to traditional linear interpolation routines (table lookups) like factorial points do, we fit a response surface to the results of Monte-Carlo explorations. It's this response surface that you see in the slice charts of the study viewer and which we evaluate at the point defined by the dimension sliders to give the red line on the parallel coordinates plot.

Parameter Paths

When entering the path for an array element, or for a subfield thereof, you must index that element using square brackets and a zero-based integer index. If you would like to, for instance, explore the effect of the z-position of your pushrod outboard point, you can enter the parameter path for just the z-coordinate of the pushrod outboard point like so: `car.suspension.front.external.pickUpPoints.rPRO[2]`. If you would like to perturb one of the parameters of an electric motor, you must remember that the electric motors element is itself an array (to allow multiple motors to be specified), and that this array must be indexed-into before you can refer to a parameter of one of the motors. For instance if you have only one electric motor, and you'd like to change the harvest power allowed, you should enter the following into the *Parameter Path* field of your exploration: `car.powertrain.electric.electricMotors[0].deployment.PHarvestRegulatoryLimit`. The same syntax is used to refer to polynomial terms within one of the aero polynomial definitions, for instance to refer to the coefficient of the second term in the front downforce polynomial you would use the path: `car.aero.PolynomialCLiftBodyFDefinition[1].coefficient`.

Saved Configs

A saved car is intended to be a snapshot representing only the components that could be fitted to an individual car at any one time. This keeps the car file size small and helps with simulation speed.

If there is a choice between multiple components where only one can be fitted to a car at any time, it may be desirable to produce a list of saved configs, in effect building up a database of components to select between. For example, if there are a number of different powertrains, then one way to save these options into a list of saved configs is to use the "Save As" button next to the Powertrain section.