# Screw Mapping in RRF

## What is Screw Mapping and what does it compensate?

*Screw Mapping* is a well-known correction mechanism for CNC mills that use leadscrews. Each motor that drives a leadscrew may use a lookup table based on measured results to improve precision.

The multi-axis screw mapping in RRF is a more generic method of correction for axis manufacturing defects since it corrects not just the moving axis but also any dependent axes. For example, take a look at this horribly exaggerated picture of a bowed rail with a head at 3 positions.



Here, the head rises (Z+) the farther from the center it goes. If you imagine the head is on two rails and one is the blue bar … as X changes **both** Z and Y change – a lot. The error is compounded by distance from the rail(s) to the end of the spindle/nozzle.

So, this RRF compensation is not just screw mapping. It is designed for per-axis physical defect correction. Like axis skew correction, one would rather repair the cause but it’s just not always possible.

### Algorithm

Each axis may have a compensation table with one row per adjusted destination axis. The table has start, increment and count parameters – so each row is the same length and they are applied to the range start, start + increment, …, start + increment\*(count-1). Any input below the range uses the start adjustment. Any input above the range uses the last adjustment. In range we use linear interpolation to find the adjustment value for a coordinate.

Axis adjustments are always applied in axis order X,Y,Z,…

**Note**: adjustments are applied to endpoints of line segments. If you want intersection points on lines to work independent of the map - add nodes to the lines at the intersection points. Curves becomes lots of small lines so they automatically follow the screw map. For normal correction amounts this should be irrelevant.

## MCode Programming

The Screw Map routines use 3 different M codes.

* Enable/disable screw mapping
* Create a new table for an axis
* Set some values within a table

### M800 Enable/Disable Screw Mapping

* Parameters

S – optional. 0 = disable, 1 = enable.

If S is omitted a report is generated showing the state.

* Example

M800 S1 ; enable screw mapping

M800 ; print state of mapping

### M801 Create a table

* Parameters

R – the table source axis

A – list of axes corrected

S – starting machine coord for table

I – distance between columns

N – number of columns

If no arguments, a report is printed

* Examples

M801 R”X” A”X” S50 I100 N10 ; create a table for X -> X starting at 50 going to 950 with 10 entries every 100mm.

M801 R”Y” A”XYZ” S0 I20 N30 ; create a table for Y -> XYZ starting at 0 going to 580 with 30 entries every 20mm.

M801 ; print a list of all of the axis table settings

### M802 Set some table entries

* Parameters

R – the table source axis

O – offset into the table(s) to place the entries

X or Y or Z or … - a list of data that goes into the destination axis

If there is no data provided, the table data is printed

* Examples

M802 R”X” X0:0.1:1.5 ; set the first three entries of the X->X map

M802 R”X” X-0.5:0.5:0 Y0:0:1 Z:0:1 ; set the first three entries of X->X and X->Y and the first two of X->Z

M802 R”Y” O10 Y0.13:0.14:0.15:1.13:1.14:1.15:10.13:0.1214 ; set the 10th through 18th entries in Y->Y map

M802 R”Y” ; print the table of Y axis adjustments

## A simple example of Axis Screw Mapping

Here’s a simple exaggerated example of screw mapping. We start with this VCarve profile to draw->



Then we run the following script->

M801 R"X" A"X" S400 I50 N6  ; start at 400, delta 50, 6 points

M802 R"X" X0:12:12:25:25:0  ; 0, expand, flat, expand, flat, shrink a lot

M800 S1

This sets the X transform table to 0,12,12,25,25,0. So it starts at no offset, then expands by 12mm then by 13mm then contracts by 25mm. Here’s the wood result->



Both line segments here show dips at the contraction area. That is because the segments are drawn piecewise linear in the original. It actually looks like this with the nodes shown->



## How to Compensate a Mill/Router using Screw Mapping

It is easy to compensate a Mill/Router using this algorithm. Assume we’re tuning X and the error terms are the usual small numbers.

* Get the best ruler you can find
* Pick an increment (like 20mm or an inch)
* Use an engraving bit to draw vertical lines at the increments selected in aluminum or other stiff material. Draw a cross (horizontal) line for reference.
* Set the ruler down on your drawn object and measure the error at each increment point.
* Type the error terms (minus) into your script

If you have a down-looking camera on the tool head, read the error off the camera views at each increment.