

Computing Junction Deviation for Marlin Firmware

As part of developing my own 3D printer firmware, I also keep an eye on what is happening in other firmware. One feature that is causing confusion in the Marlin community is the junction deviation setting. Up until recently, Marlin used the jerk method (hence forth referred to as "archaic jerk") it inherited from Grbl for computing corner speed (junction velocity). With the option now in Marlin to use junction deviation instead of jerk, there are many people who want to know what are good settings for junction deviation to insure they get reasonable movement while printing. In this post I will give an equation for converting the jerk

values into junction deviation and my derivation of this equation.

I will not rehash what junction deviation is in this post, so anyone interested in learning more about it should refer to these two links:

https://onehossshay.wordpress.com/2011/09/24/improving_grbl_cornering_algorithm/

<https://reprap.org/forum/read.php?1,739819>

For those who are only here for the end results, here is the equation to compute junction deviation:

$$d = 0.4 \frac{Jerk^2}{Accel_{printing}}$$

A typical Cartesian machine might have a jerk setting of 9mm/s and a printing acceleration of 1500mm/s². Plugging these numbers into the above equation goes like this:

$$0.4 * 9 * 9 / 1500 = 0.022$$

So for this example the junction deviation would be set to 0.022 and the configuration_adv.h file would look like this:

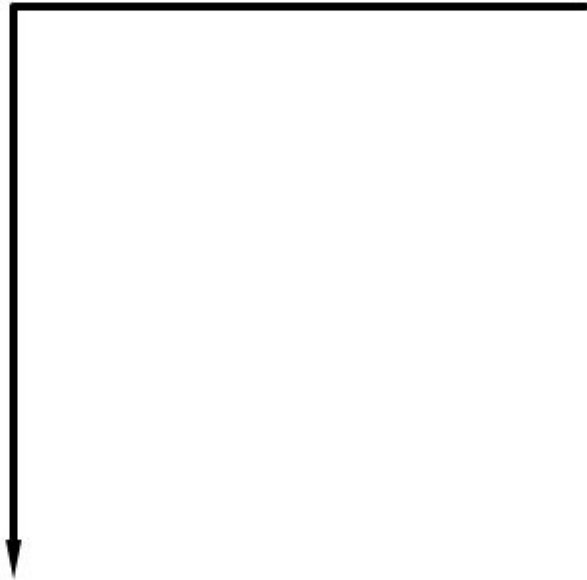
```
//  
// Use Junction Deviation instead of traditional Jerk Limiting  
//  
#define JUNCTION_DEVIATION  
#if ENABLED(JUNCTION_DEVIATION)  
    #define JUNCTION_DEVIATION_MM 0.022 // (mm) distance from real junction  
edge  
#endif
```

The smaller the junction deviation number, the more the machine will slow down when encountering corners. Making this number too small may slow down the print speed and cause over extrusion in the corners. Making the number larger will increase the speed at which the machine traverses corners. Setting this number too high may cause vibrations in the mechanism that produce ringing in the print surface or in extreme cases a stepper motor to loose position.

If you only came here for the basic equation, you do not need to read any further. For those who would like to learn where the above

equation came from, keep on reading.

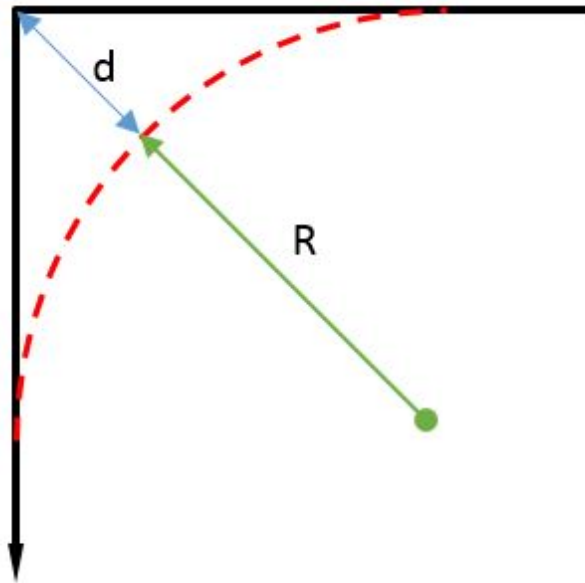
To setup this problem, we are going to imagine a situation where the printer traverses two moves that are at right angles to each other:



This is a convenient situation as it is the case where one axis suddenly stops moving, and the other axis suddenly starts moving. When using the archaic jerk functionality the first axis will decelerate down to the the jerk velocity in the corner and then instantly stop, at the same

moment the second axis will suddenly jump up to the jerk velocity in its direction and accelerate back up to printing speed. To simplify this, we can say that with a 90 degree corner, the junction velocity is the same as the jerk setting. (note: this is not the case for any junction that is not 90 deg).

Next we can take that same corner move and add the virtual arc that junction deviation uses to compute the junction velocity:



In the above picture **d** represents the junction deviation and **R** is the radius of the virtual arc.

Using the Pythagorean theorem and a little algebra and we get this relationship between **R** and **d**:

$$(R + d)^2 = R^2 + R^2$$

$$d = \sqrt{2 * R^2} - R$$

$$d = (\sqrt{2} - 1) * R$$

$$d \approx 0.4 * R \quad (1)$$

Next we take the equation for radial acceleration and solve for the radius:

$$A_{radial} = \frac{V^2}{R}$$

$$R = \frac{V^2}{A_{radial}} \quad (2)$$

If we combine equations 1 and 2 above we find the following:

$$d \approx 0.4 \frac{V^2}{A_{radial}}$$

In this situation, the radial acceleration is the same as the printing acceleration and the junction velocity is the same as the archaic jerk setting. Substituting these in, we get the final equation:

$$d = 0.4 \frac{Jerk^2}{Accel_{printing}}$$

Using this equation will allow someone who is migrating from the archaic jerk to junction deviation to start with settings that are very close to each other without excessive experimentation or testing.

Please let me know if you have any questions.

Phillip