Written by Chris Humphris 13Nov13

CSS G95 Observations

I was not aware of the G95 issue during CSS (G96) until Hood pointed this out. I had simply assumed that CSS worked correctly because my focus was on the spindle speed changing correctly with diameter change.

Since I have been made aware that the G95 (Feed per Revolution) had issues, I performed some tests. This is what I observed:

I calibrated the X Axis with the Master Tool and moved the tool position to X50.

From the MDI, I issued G95 F0.1 (the DRO “F” changed to 0.1)

Then I issued G96 S50 M3 (The spindle started at 348RPM)

I then issued the command G1 X25 and watched the “F” DRO. As the spindle increased in speed, the DRO reading dropped gradually to 0.05 just before reaching X25. When X25 was reached, the DRO instantly reverted to 0.1.

I performed more movements, summarized as follows:

Movement (mm)	“F” DRO Observation (mm per rev)	Remarks
X50 to X25	0.1 to 0.05	“F” reverted to 0.1 when movement stopped
X25 to X12.5	0.1 to 0.05	“F” reverted to 0.1 when movement stopped
X12.5 to X0 (Centre)	Remained at 0.1	This is because the spindle had reached its maximum (no RPM change)
X0 to X12.5	Remained at 0.1	This is because the spindle had reached its maximum (no RPM change)
X50 to X12.5	0.1 to 0.03 (0.025?)	“F” reverted to 0.1 when movement stopped
X12.5 to X50	0.1 to 0.3	“F” reverted to 0.1 when movement stopped
X12.5 to X25	0.1 to 0.14 (0.15?)	“F” reverted to 0.1 when movement stopped
X25 to X50	0.1 to 0.18 (0.2?)	“F” reverted to 0.1 when movement stopped
Any Z movement without X movement	0.1	No change in Feedrate observed

Note that accurate ”F” figures could not be absolutely determined, because of:

a. time lag in the display update, and
b. resolution of the display.

Assuming that the “F” DRO readout is reasonably accurate, and the actual Feed per Revolution is really changing, it is obvious that the Δ Feed per Revolution occurs only when there is X motion and a change in RPM. This is also supported by the observation that Feed per Revolution appears not to change when a Z movement is occurring at a fixed diameter (where the RPM is constant).

In summary:

As the diameter decreases by ½, the Feedrate appears to decrease to ½ F.

As the diameter increases by 2, F appears to increase by approx. 1.5.

This was also observed by jastein in Post 17 of

http://www.machsupport.com/forum/index.php?topic=20643.10

What does all this mean?

What all this means is that Constant Surface Speed (Tangential Velocity past the tool tip) is functioning correctly, but chip load is not constant. Rather, a varying chip load appears to be taking place.

In other words, G96 functions correctly, but G95 does not when X Axis is in motion.

Does it really matter?

My lathe, which I thought was cutting at 0.05mm per rev during parting operations; actually appears to, in theory, start the cut at X50 at 0.05mm per rev and linearly decreases to 0.0125mm per rev at X12.5, when the feedrate reverts back to 0.05mm per rev until the centre is reached.

I've OD machined and cut off hundreds of parts using G96 and G95 F0.05 over a few years now, totally unaware of this phenomenon. Obviously, for me, it didn't matter. The parts always come out looking great.

Where to now?

Now that I am aware of the apparent varying chip load, I am turning my thoughts as to how to compensate in G Code. I am thinking:

1. Peck Parting, where the pecks are relatively small (say 5mm). In this example, for parting a 50mm rod, ΔF would be approx. F/10. So for F=0.05, ΔF becomes 0.005 for each peck. This would mean that the chip load variation during the parting sequence would be insignificant.

2. Some form of mathematical calculation using a variables and formulae to keep F constant, as RPM and X vary. (Maybe a macro?)

I'm not a mathematician so looks like option 1 for me, if required. Perhaps other more learned folks could tackle option 2 for us all.