

Progress Review 9

Nikhil Baheti

Team F: ADD_IN

Teammates: Ihsane Debbache, Dan Berman and Astha Prasad

ILR08

February 24th, 2016

1 Individual Progress

This fortnight, I had to work on completing the firmware for the 3D printer. This includes the control of the r-axis stepper motor similar to other axes by modifying the firmware. I modified the other G-codes where the r-axis variables are used to ensure that the system is consistent across all axes. All this was done to make a single firmware. Also, I extended my work to do the inverse kinematics map of the nozzle tip to consider the offset in the position of the nozzle while considering the R-axis rotation. Also, in the software we prefer that the R-axis is controlled by a relative mode so that the values don't cumulate and thus the firmware was also changed to be in accordance with this.

1.1 Firmware working of E1 port

After changing the use of the winmerge tool it was evident that the variables changed during this process were responsible for the bug and thus the following steppers were used to debug the problem:

- Check where each of the above variables changed. This did not lead to any progress.
- Considering that the bug may be due to an unused variable, I removed all the unwanted variables from the firmware. This also did not give any clues.
- To print values of the variables in the ISR I initialized a static variable which prints after 500 iterations.
- Using the above variable I monitored the 5 pins that control the E1 port by using the DigitalRead function of the arduino library and compared them with other port pins. Both the ports produced the same sequences, however, only the y-axis port was running the motor and not the E1 port. This was very unusual because there are only 5 variables that control the driver that drives each stepper motor and if the stepping sequence is the same it means that the motor should be stepping.
- The only reason the motor would not step is if the sequence being generated is quicker than the operation frequency of the stepper motor. But since we have added lines of code it should slow down the process and not increase the frequency. Also, this property must be common to all ports.
- Finally, with the help of teammate Dan, we used an oscilloscope to monitor the pins of the stepper motor drivers. All the pins gave the same result. Finally, we checked the digipot pin which controls the supply to the motor (This was not indicated on the circuit diagram). We noticed that the supply to the motor was down for E1 port. Thus, we checked the digipot pieces of code and in one part we were accessing a 6th channel that was not initialized in the firmware. This resulted in an uninitialized E1 port. After correcting this error the E1 port worked.

1.2 Modifying different G-codes

Various G-codes use the x, y and z axis variables to update and use them. Like G92 is used to set reference points. I have modified these codes to work for r-axis.

1.3 R-axis relative control mode

In the software we prefer that the R-axis is controlled by a relative mode so that the values don't cumulate and thus the firmware was also changed to behave in accordance with this. To do this the relative mode variable has been set to true.

1.4 Inverse Kinematics of the Nozzle Tip

Currently, all the axes are being moved synchronously with constant velocities with the x-axis and y-axis being moved in a straight line. This results in the extruder tip moving along a curve as shown in Figure 1. So, I worked out the inverse kinematics for the x and y location such that the extruder tip moves along the required straight line. The following are the equations for the inverse kinematics:

```
function q = IK(x,y,r)
offset = 5;
q = [x - offset*cos(r) y - offset*sin(r) r];
end
```

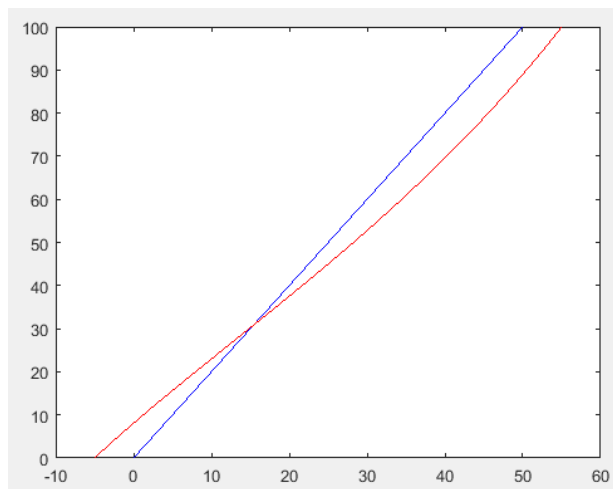


Figure 1: Red line shows the path the X, Y axis must follow to ensure that the extruder tip moves along the blue line which is the profile that the G-code suggests

2 Challenges

The challenges faced during the term of this progress review which are quite consistent with the previous problems and are explained as follows:

1. **Arduino IDE:** The Arduino IDE for tracing variable declaration, references and initiations is not user-friendly. Thus, I had a hard time tracing variables across files.
2. **Debugging:** The firmware could not include break points to monitor the variable which resulted in a lot of time waiting to write the code even for a small change in the code. Also, I had to use print statements echoed to the serial terminal to debug sections of the code.
3. **Scoping the RAMBo board:** The RAMBo board PCB lines are covered and thus it is only possible to read at the IC pins. It isn't possible for one person to execute the commands and read the pins simultaneously. Dan helped me overcome this challenge by reading the pins using an oscilloscope while I executed the commands.

3 Teamwork

- **Astha:** She worked on making a collision check function which will be used for files with multiple COTs item to check if the path planning is causing the nozzle to collide with other COTs item. She also resolved the error in the G-code files which was producing NaN values for the R-axis.
- **Dan:** He worked on making a collision check function with Astha which will be used for files with multiple COTs item to check if the path planning is causing the nozzle to collide with other COTs item. He also tested his nozzle which we are looking to incorporate in our design.
- **Ihsane:** He was responsible for testing the machined nozzles which is still facing clogging issues. He also had designed a mount for the encoder and an encoder wheel for the R-axis stepper motor to keep track of the R-axis home position.
- **Team collaboration:** There was no major component of team collaboration as most of the work was divided, but we did meet regularly to check progress. Also, Dan helped me with solving the firmware bug by narrowing the error to the digipot using an oscilloscope.

4 Plan

For the next progress review, I plan to work on the inverse kinematics map. This includes making modifications to the velocity profiles to ensure that extrusion of the filament is consistent along the print line. Also, I will modify the new printer's firmware to ensure that testing may be started on that printer too. After these modifications we expect to be able to print around COTs item and thus I will be a part of it and start testing by printing around the COTs item.