Progress Review 10

Nikhil Baheti

Team F: ADD_IN

Teammates: Ihsane Debbache, Dan Berman and Astha Prasad

ILR09

March 17th, 2016

1 Individual Progress

This fortnight, I had to work on completing the inverse kinematics map for the firmware for the 3D printer. This includes making modifications to the velocity profiles to ensure that extrusion of the filament is consistent along the print line. After these modifications we were able to print around COTs item with a few minor bugs. I also worked partially on the homing of the R-axis.

1.1 Inverse Kinematics Map

The inverse kinematics map for the nozzle tip was solved before the previous Progress Review. It is necessary to ensure that the nozzle tip moves along the G-code specified with a constant rate so that the extruded filament is uniform for which the inverse kinematic map was used. The following steps were performed to achieve the same:

- Difference equations for the nozzle tip were calculated based on the forward kinematics map given below:

$$dx = dxe - \cos(r + dr) + \cos(\mathrm{r});$$
$$dy = dye - \sin(r + dr) + \sin(r)\,;$$

$where,$
$dxe \rightarrow\ steps\ the\ nozzle\ tip\ move\ along\ X - axis\ per\ timer\ interrupt$
$dx \rightarrow steps\ the\ X - axis\ stepper\ motor\ must\ move\ per\ timer\ interrupt$
$dye \rightarrow\ steps\ the\ nozzle\ tip\ move\ along\ Y - axis\ per\ timer\ interrupt$
$dy \rightarrow steps\ the\ Y - axis\ stepper\ motor\ must\ move\ per\ timer\ interrupt$
$dr \rightarrow\ \ steps\ the\ R - axis\ stepper\ motor\ must\ move\ per\ timer\ interrupt$
$r \rightarrow current\ R - axis\ angle$

- Based on the above equations it is evident that the R-axis may be stepped at any desired rate and thus is stepped linearly with respect to time.
- Also, the x and y axes stepper motors stepping rate changes every time interval and is computed every ISR. Also, the fractional steps are carried to the next time interrupt ensuring that no steps are lost and the nozzle tip moves along the desired line.
- The planner function gives information about the number of steps the nozzle tip must move per timer interrupt along the respective axes. This information is used in the ISR to substitute for *dr*, *dxe* and *dye* in the above equations to compute the steps that the x and y axes stepper motor must traverse for the next timer interrupt.
- While making these modifications it was noticed that the variables which contain the information of the number of steps for each axis are always positive and thus care must be taken to ensure that the code works for all directions. This process took time

but finally an easy solution was found. The information of the direction of a stepper motor is stored in a “count_direction” variable which is multiplied in the above equations.

- The next corner case is a situation when an axis moves along both directions for a given Gcode. By default in the Marlin firmware the axes are expected to move only in one direction per Gcode. Thus when the axes are forced to move in both directions due to the nonlinearities in the difference equation, the stepping direction of the stepper motor is changed. This also solves the case wherein the Gcode requests only a motion in R-axis. In this situation the tip of the nozzle must be at the same location which can be achieved by moving the x, y and r axes in synchronization along a circular path with radius given by the offset in the nozzle tip and the axis of rotation.

The Figure 1 shows a straight line printed by the printer after the inverse kinematics map was solved.

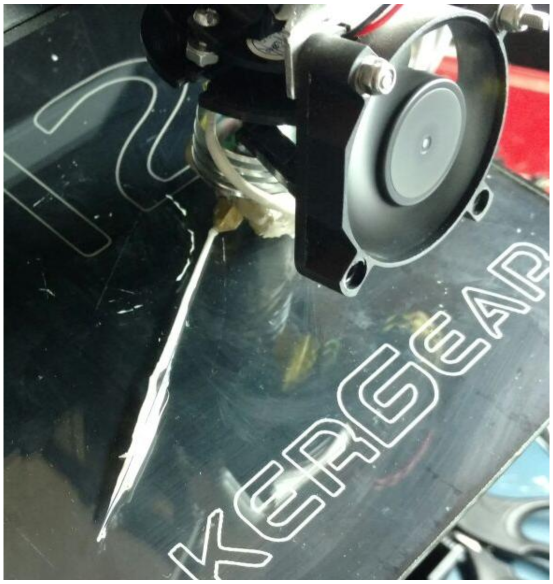

Figure 1: Straight line Gcode executed by the firmware

1.2 R-axis Homing

The R-axis homing is achieved by an optical encoder and slotted wheel mechanism connected to the hollow stepper motor. The encoder is connected to the limit switch connections on the RAMBo board. This pin is read to find the homing location. However, this section of homing is currently under development. Basic modules have been implemented but these must be tested.

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 is not user friendly
2. Changes in the firmware and testing takes a lot of time which slows the debugging process. Also standard JTag is not present on the printer which does not allow in program debugging.
3. I was stuck at one point in the inverse kinematic map trying to define the velocity of the x, y and r axes may be modified to ensure that the nozzle tip moves along the desired straight line defined by the gcode. Dan helped me come across this by explaining an alternate way to assign the velocities.
4. Solving the inverse kinematics map for the printer while being consistent with the coding method is time consuming. The current method uses a single timer to control the timing for stepping all the stepper motors. The rate of stepping for each axis is controlled by variables in the ISR. This not only takes more time to execute the code but also makes the axes coupled with each other. Thus a lot of care had to be taken to ensure all the variables and corner cases are taken care of while solving this inverse kinematic map.
5. Currently, the inverse kinematics map takes a lot of computation time in the ISR and thus we need to reduce this time. The excess time being used is resulting in communication errors with Pronterface.

3 Teamwork

- **Astha:** She worked on solving minor bugs in the software Gcode generation. She has also taken a brief look into the web development part.
- **Dan:** Dan also worked on solving minor bugs in the software Gcode generation. He also assisted everyone in different aspects of integration. He helped me with choosing the velocity choices for the axes.
- **Ihsane:** He was responsible for testing the machined nozzles and has solved the clogging issues. He also had designed an improved 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:** This progress review we integrated all subsystems and thus everyone was working in proximity with each other. Astha and I were working on the Gcode protocol bugs where the software was not consistent with the firmware. Dan created various Gcode file required for testing the firmware. Ihsane helped with the hardware issues, specifically with the nozzle jamming issues.

4 Plan

For the next progress review, I plan to work on the making the inverse kinematics map time efficient. Also, I will modify the new printer's firmware to ensure that testing may be started on that printer too. I will also complete the R-axis homing command. Since, we are in the integration phase I will take care of any issues that arise with the firmware during the testing.