Screw Auger Measuring Device

For measuring and mixing base filament material

Team Jones

Christopher Anderson

Kyle Kruger

Tosh Jones

Travis Holm

ENGR 480

Manufacturing Systems Engineering

12/18/15

Table of Contents

Table of Figures

Introduction

This manual provides a list of instructional material, figures, and wiring diagrams, which are crucial to the operation of this machine.

Loading the Machine

Before turning on the machine, make sure that the hopper is filled with the material you wish the machine will sort. The machine will run without any material in the hopper, but loading it before will increase efficiency. Also ensure that the collector is in the upright position.

Starting the Machine

To start this machine, follow these steps:

1) Check that machine is loaded
2) Connect air pressure to pneumatics
3) Connect the 24V and 15V power packs to the mains
4) Open LinuxCNC
5) Load and run the start program

Clearing Jams

Clearing jams on this machine generally requires running the screw auger in reverse for about 2 seconds, and then continuing as normal.

Description of Machine Operation

Section 1: Hopper

The hopper is a temporary funneling system used to guide the tiny pieces of material into the screw auger. The hopper size of the hopper depends on how much time you'd like to spend away from this machine, the larger it gets, the more material you can process before you need to refill. In the most advanced design, the hopper should be attachable to the screw auger. The hopper is shown in figure 1.

Section 2: Screw Auger

The screw auger is used to move material at specific rates, quickly at first, but then slowly when fine tuning the mass of material held in the collector. This step allows for a marked improvement in batch accuracy. The screw auger is a screw made from Delrin and is loosely encased in a clear PVC tube. The auger is controlled by a stepper motor. The screw auger is shown in figure 1.

Section 3: The Collector

The collector collects and measures the material being run through the auger. It consists of a half cup Tupperware container, glued to a 3D printed, ABS plastic coupler which screws into a 100 gram load cell. This load cell is then screwed into a mounting bracket which attaches to a pneumatic rotary actuator. The collector is shown in figure 1 and in detail in figure 3.

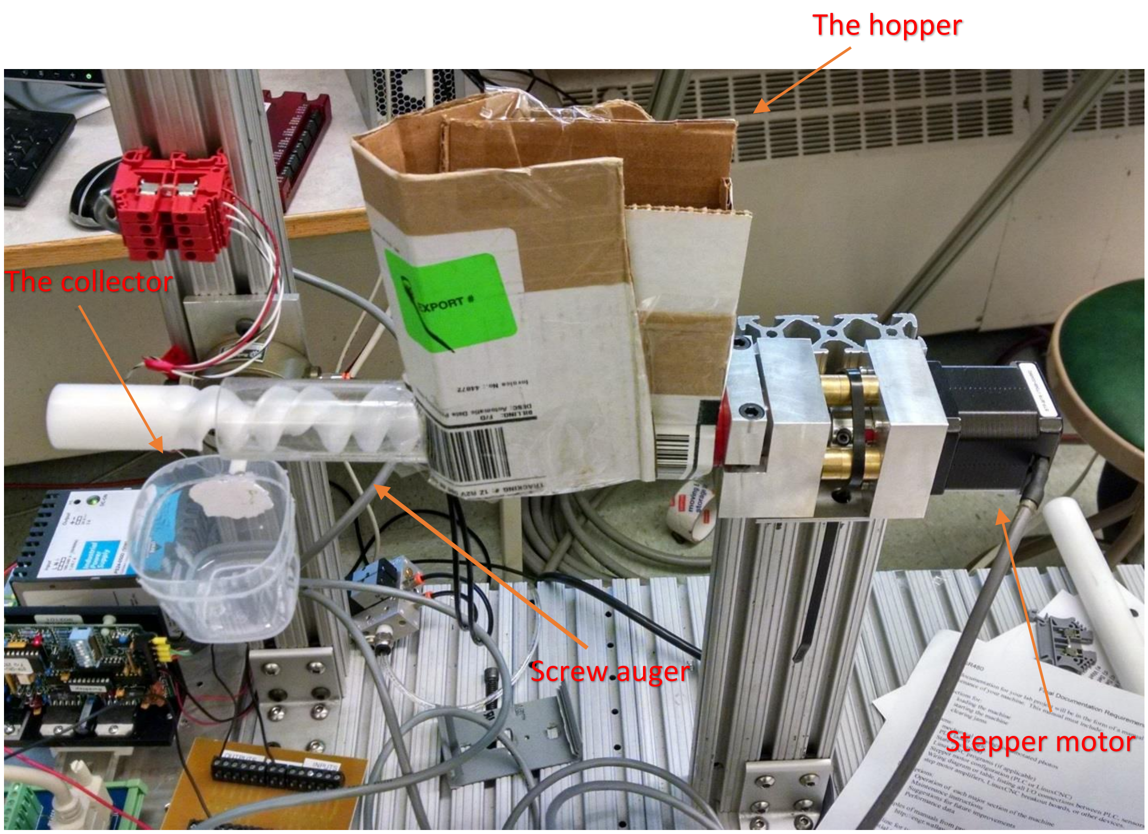

Figure 1: The full Screw Auger Measuring Device.

Maintenance Required

The collector is the component of the machine that will need the most maintenance. The frequent movement of the wires can cause them to rip and tear, and the adhesive used to bond the Tupperware and the ABS plastic coupler was not a high quality adhesive and should not be considered for long term operation. The wires and adhesive will most likely need to be replaced monthly.

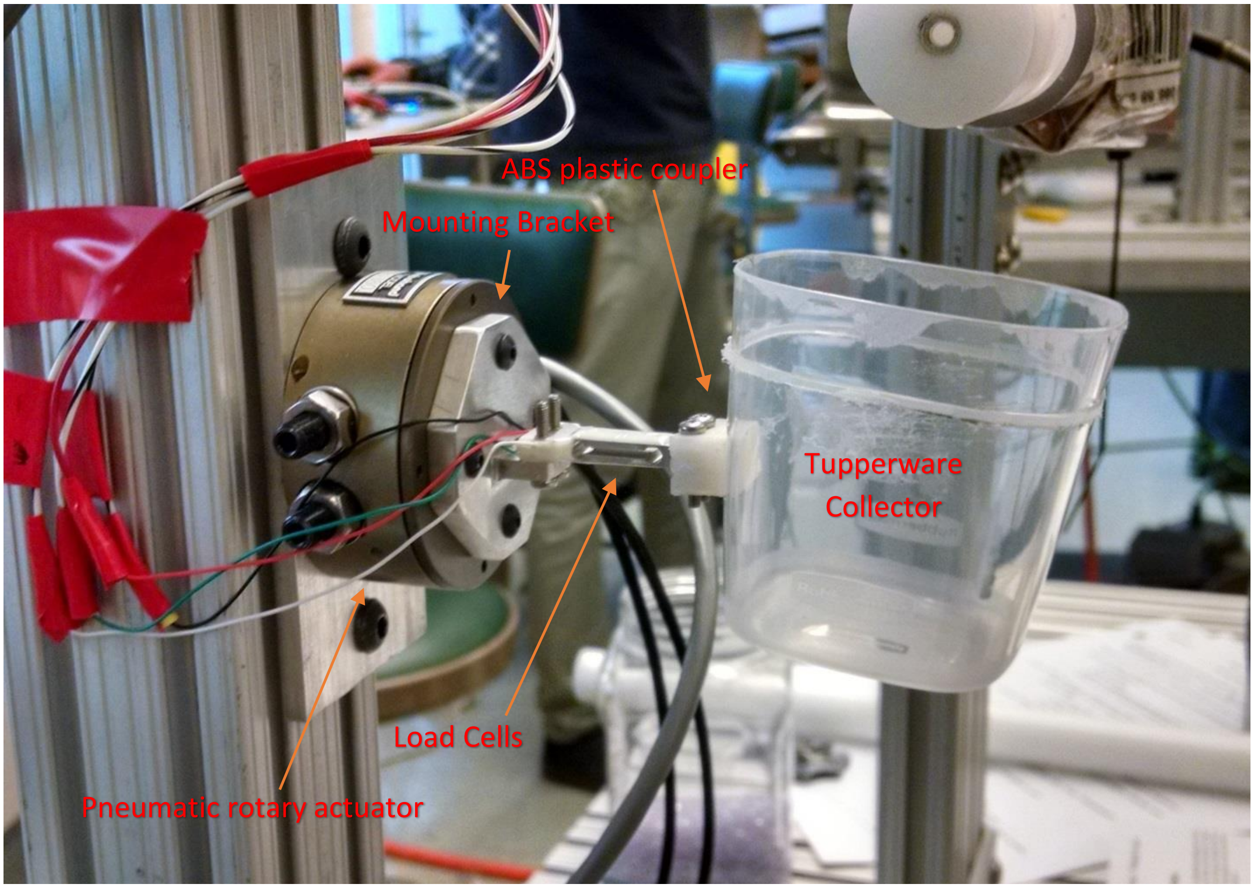

Figure 2: A detailed view of the collector.

Future Improvement

Hardware

There is a great potential for future improvement on this machine. Firstly, the Tupperware collector and the ABS plastic coupler could be replaced with a single part; thus, eliminating the need for an adhesive and improving the durability of the collector. Secondly there could be a housing system for the wires that allows them to move freely, but also keeps them stable while the pneumatic rotary actuator is moving. The hopper needs to be improved, currently it is made of cardboard and tape, there could be a permanent hopper made that is connected to the casing of the screw auger. This should reduce the number of jams. A better load cell should be used. The current cells cost $7 a piece and they are not very durable. A more durable, higher precision, higher weight load cell could be used, that way the load cell could be placed on a part that is not rotating, as the rotation greatly decreases the longevity of the machine. The electronics could be reduced by using a higher quality breakout board, such as the Mace Electronics breakout board. This board has its own analog import, which eliminates the need for the LabJack, which turned out to be difficult to use and program.

Software

Currently the system does not keep track of weight cell readings over time, but to make the system more accurate, the program could be edited to calculate a running average. This average would be used by the program to compensate for any discrepancies with the desired weight.

Performance Data

When taking measurements, we ran our system with a spindle speed of four that reduced to one when the 80% load point was reached. It ran at these conditions for 5 cycles of 30 gram batches collecting a total of 150 grams of material in 60 seconds.

$$Mass\ Flow\ Rate = \frac{150\ grams}{60\ seconds} = 2.5\ \frac{g}{s} = 9\ \frac{kg}{hr}$$

This flow rate can be improved by increasing the spindle speed or the percentage load at which the system slows for the sake of precision. Alternatively, the batch size can be increased which reduces the no flow time when the batch is dumped. The target flow rate is 10-30 kilograms per hour for the entire system. As this would comprise only one part of the system and that the flow rate can be increased further, this system falls well within the required mass flow.

The following is a table with 5 cycles of data to show calculation of short term accuracy:

Initial Voltage (V)	Final Voltage (V)	Voltage Difference (V)	Mass (g)
-1.0738	-2.7166	-1.6428	30.08791
-1.0752	-2.7614	-1.6862	30.88278
-1.0758	-2.7211	-1.6453	30.13370
-1.0758	-2.7254	-1.6496	30.21245
-1.0779	-2.7250	-1.6471	30.16667
Averages			
-1.0757	-2.7299	-1.6542	30.2967

The equation used to convert voltage difference to mass is shown below. The scaling parameter was acquired and verified by two tests using standard mass pieces.

$$W = \frac{Voltage\ Difference}{-0.0546}$$

With this data, the short term accuracy is 0.99%, which is below the target short term accuracy of 1%.

<u>CAD Database</u>

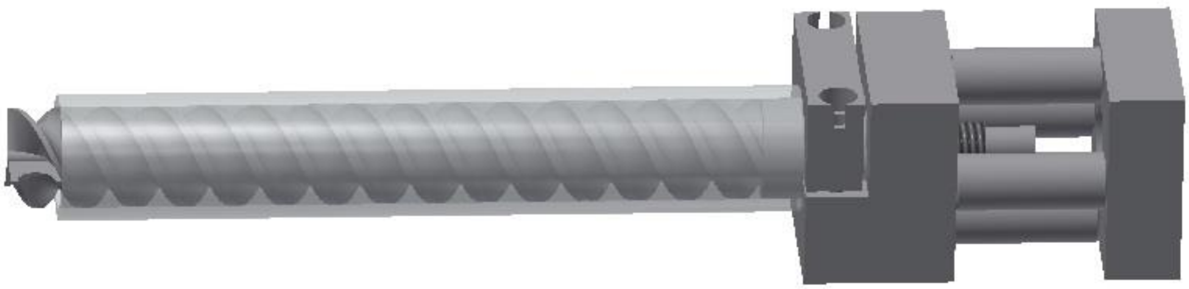

Figure 3: Screw Auger in CAD

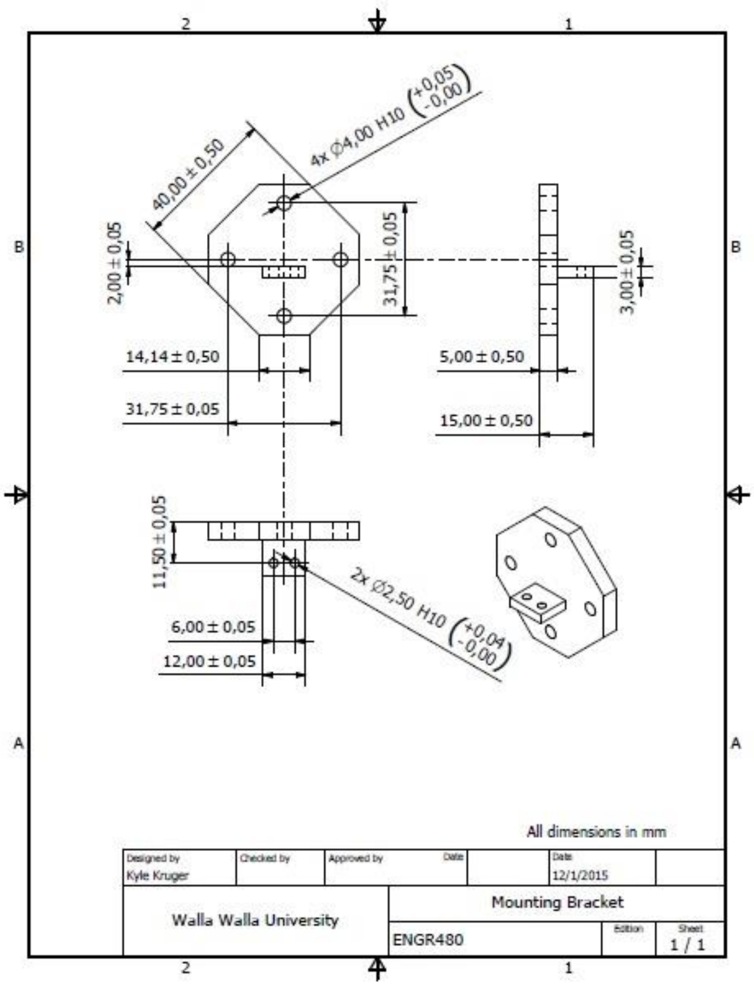

Figure 4: Annotated Drawing of Mounting Bracket

Wiring Diagrams

The wiring diagrams have been configured to show how additional systems would be added.

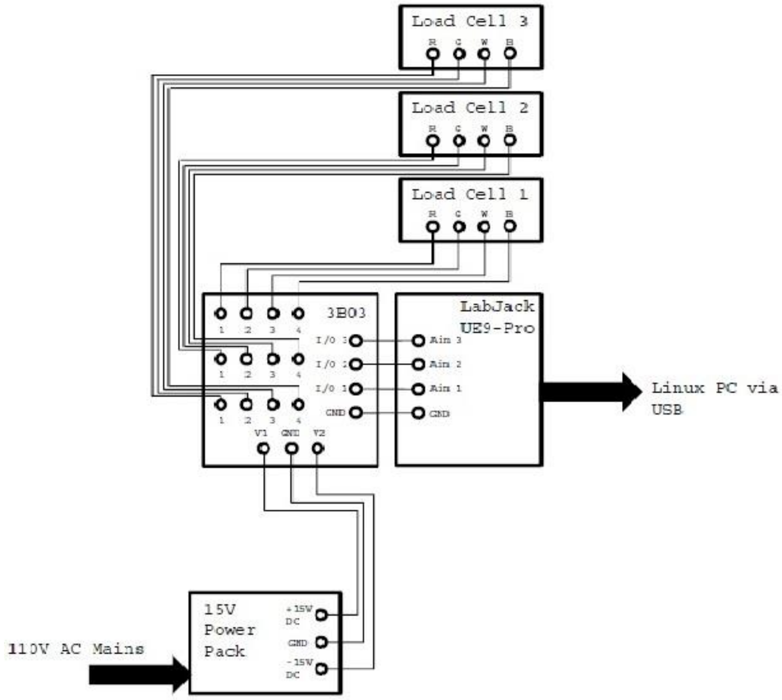

Figure 5: LabJack Wiring Diagram.

Notes: The LabJack is our Analog-to-digital converter to convert and analog voltage from the load cell to a digital value in the Linux CNC program.

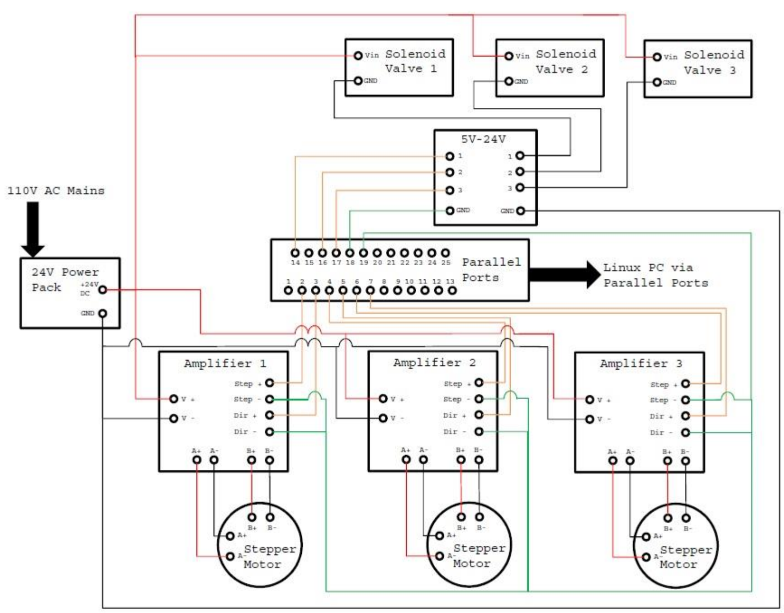

Figure 6: Stepper Motor control wiring diagram.

Linux CNC

```
(axis,STOP)

M3 S4                               (Set speed)
#1 = [1]                            (Counter for Number of Cycles)
#9 = [10]                           (Total Desired Cycles)
#10 = [30]                          (User defined goal weight in grams)
#11 = [-0.0546*#10]                 (Voltage difference required)

(Fill the load cell and dump it iteratively for the total number of cycles)
o101 while [#1 LT #9]
    M66 E0 L0 Q10                   (Zero Voltage)
    (debug, #5399)                  (Display Initial Voltage)
    #14 = [#5399]
    M64 P0                          (Turn motor on)
    #2 = [1]
    #3 = [1]
    #12 = [#11 + #14]               (Load cell voltage at goal weight)
    (Read the load cell ten times every second. When it reaches the critical
value, break from the loop)
    o102 while [#2 LT 2]
        N5 G4 P0.1                  (Tenth of a second delay)
        M66 E0 L0 Q10               (Read the load cell)
        (If the load cell reads the appropriate load, break from the loop)
        o110 if [#5399 LT #12]
            #2 = 2                  (Successful load operation)
        o110 elseif [#5399 LT 0.9*#12]
            M3 S1                   (Slow spindle down at 90% of load)
        o110 elseif [#3 EQ 600]     (If the loop has run for 60 seconds)
            #2 = 3                  (Unsuccessful load operation)
        o110 endif
        #3 = [#3+1]
    o102 endwhile

    o120 if [#2 EQ 2]               (Condition for successful load operation)
        (debug, #5399)              (Display final voltage)
        M65 P0                      (Turn motor off)
        N10 G4 P0.5                 (Half second delay)
        M64 P3                      (Rotate canister dumping load)
        N20 G4 P1.5                 (one and one half second delay)
        M65 P3                      (Rotate canister to receive load)
        N30 G4 P1                   (Second delay)
        #1= [#1+1]
        M3 S4                       (Reset Speed)
    o120 elseif [#2 EQ 3]           (Condition for unsuccessful load
operation)
        #1=[#9]                     (End all cycles)
    o120 endif

o101 endwhile
M30                                 (End program)
```