Motor Calibration for an Autonomous Gardening System

Brian Carlsen  
Department of Mathematics  
University of Colorado  
carlsen.bri@gmail.com

Ben Leduc-Mills  
Department of Computer Science  
University of Colorado  
benjamin.leducmills@colorado.edu

Abstract—This paper discusses the motor calibration experiments conducted to calibrate the X-Y movement gantry for an autonomous robotic gardening system, called Autoponics[1], being constructed at Solid State Depot[2]. Discussed is the system setup of the hardware and software components, the calibration experiment methodology, the results of the calibration testing, and a discussion of these results.

I. INTRODUCTION

For any autonomous system a crucial element to incorporate is localization—the ability of the system to know where it is. In the creation of Autoponics, an autonomous robotic gardening system, localization is involved in almost every task. Constructing accurate point clouds from sensor data, identifying plants and their particular growth stages, and harvesting all become very difficult if not impossible without accurate localization in the system. For this reason, spending time to precisely and accurately calibrate the gantry system movement is essential. Accurate calibration will yield more reliable data and more accurate movement while the system is running.

In this project’s setup an X-Y gantry provides movement for a vision system and end-effector. The carriage on which the vision system and end-effector is mounted is attached to the X-axis of the gantry, which runs parallel to the ground. Each end of the X-axis is mounted to a Y-axis, which allows the X-axis to move vertically. The carriage is driven along the X-axis using a stepper motor mounted to the carriage. The motor controls a belt drive which the carriage crawls along. Each end of the X-axis is also driven by a stepper motor. The motors are mounted at the top of each respective Y-axis acting like a winch to control the X-axis vertical movement. The entire X-axis is counterbalanced on either side to ease the strain on the Y-axis motors. The experiment and results presented here consist only of calibration tests performed with the carriage movement along the X-axis. However, because control over the Y-axis motors is nearly identical to that of the X-axis motor, the methods applied here can be easily adapted to calibrate the Y-axis. The remainder of the paper is as follows: we present a more detailed description of the gantry system, followed by a description of the methodology, results, and discussion of our motor calibration tests.

II. MATERIALS

For our calibration tests we used the X-axis of the gantry system. The X-axis consists of a 250 cm length of MakerSlide[3] to which our carriage attaches. The carriage itself consists of a stepper motor (and associated hardware to move the carriage), a Hokuyo laser scanner, and an IPEVO document scanner, pictured in Figure 1. The MakerSlide is set between two other lengths of MakerSlide which create the Y-axis gantry support. Attached to the top of the MakerSlide is a belt drive running the length of the axis. The carriage consists of a plate of steel with four wheels attached, two that rest on top of the MakerSlide, and the other two on the bottom. The top two wheels carry most of the carriage weight, while the bottom two wheels provide tracking support, keeping the carriage vertically aligned. Attached to the carriage is a stepper motor along with two pulley wheels on either side. The drive shaft of the motor has a pulley wheel attached to it as well.

The belt drive is thread over the pulley wheel of the motor and under the additional pulley wheels. The two additional pulley wheels act to both keep tension in the belt drive, allowing the motor’s pulley wheel to grip the belt, and keep the belt drive aligned with the motor’s pulley wheel. In addition to the movement system, the vision system is attached to the carriage using a metal L-plate. The vision system consists of an IPEVO RGB document camera[4] and a Hokuyo URG-04LX[5] laser scanner. A picture of the carriage with these components visible can be seen in Figure 2.

The movement of the carriage along the X-axis is controlled using an Arduino Uno[6] microcontroller. The Arduino is connected to the X-axis stepper motor, via an EasyDriver v4.3[7] breakout board. The program on the Arduino can give commands to the EasyDriver, which the EasyDriver board then translates into step and direction commands to the stepper motor. We configured the EasyDriver board to use full steps rather than microsteps; as full steps are still accurate enough for our calibration, and allow us to deal with simpler data-types for keeping track of step counts - microstepping would produce a step count number eight times as high without noticeable performance or accuracy gain (our stepper motor normally operates at 200 steps per rotation of the motor shaft, microstepping would put it at 1600 steps per rotation. Also
connected to the Arduino board are two infrared photo interrupters. The photo interrupters work via an infrared emitter and detector pair; providing the IR beam is uninterrupted the sensor will read HIGH to the microcontroller, but as soon as the beam is interrupted (i.e., an object passes between the emitter and detector) the sensor reads LOW. This way, the photo interrupters act as end-stops within the gantry system, ensuring that the X-axis carriage stays within the specified range. One interrupter is mounted on each side of the carriage, aligned with one of the slots in the side of the MakerSlide rail. In this same slot two screws are placed, one on either side of the carriage, which trigger the respective interrupter, thus acting as limit switches, as seen in Figure 3. The software system running the hardware consists a single ROS (Robot Operating System) node. To communicate with the Arduino board (and thus control the motor) the node uses the rosserial Arduino ROS package.

III. METHODS

The system is based off of the publisher/subscriber model within ROS. When the Arduino is turned on it first homes to the left limit screw. This prevents inaccurate data from being reported. Once homed, the Arduino starts bouncing between the two limit screws. It accomplishes this by moving in five step increments and checking whether an interrupter has been triggered. If neither interrupter has been triggered then the command to continue moving in the same direction is given. However, if either limit switch has been triggered a command to stop the motor is issued, and the number of steps since the last interrupt is published to the ROS topic. The direction of travel is switched and the carriage is moved off of the limit screw. Normal operation then continues. The ROS node listening on the topic is given a user-specified file. The node opens the specified file and appends the number of steps and the ROS walltime to a new line. Figure 4 shows the ROS node live logger output as the ROS node is publishing stepper data.

In this way the system is completely autonomous once started. The Arduino will bounce the carriage between the two limit switches, publishing step data at either end, and the ROS node will record this data and write it out to a text file.
To conduct calibration testing a ruler is placed along the X-axis, and the axis is leveled. The limit screws are then set at desired locations and the carriage is set to run. We allowed several hundred iterations for each trial. Not only did we vary the distance the carriage travelled, we also varied the position of the screws along the X-axis. For example, we ran several tests setting the limit screws 10 inches apart. However, some of the tests took place on the left end of the X-axis, some on the right end, and some in between. This was done to test if there is a variation between moving on different sections of the X-axis. To test whether there was a difference between moving left and right, the Arduino also published its direction of movement which was recorded by the subscribing node.

IV. RESULTS

In all, we conducted 13 different trials of various distances along different parts of the X-axis. Table I shows the breakdown of runs, distances, and results for all trials.

The main purpose of running such a variety of trials was to find the average steps per centimeter for the X-axis stepper motor, along with its error, for the entire system. We decided to convert to centimeters after initially measuring steps per inch, as the rest of our system measurements were metric. To find this value we first averaged the number of steps for each trial of the experiment. We then divided this average step count by the distance the carriage travelled. It is of note that the distance recorded during the trial runs did not account for the length of the carriage; this was accounted for by subtracting the length of the carriage from the calculated distance. Comparing the average steps per inch of the left side of the axis and the right side of the axis yielded no significant difference. Compare Figure 6, which shows a 3 inch length on the left end of the X-axis, with Figure 7, which shows the same distance on the right end of the X-axis - we see that although the mean is higher in the 81-91 trial, the average steps per inch only differ by 23.4, which is only a 3% difference in values, while the standard error and standard deviation are almost identical. Also, the distance of the run showed no significant difference in the average steps per inch. To check if the number of steps per run drifted over time, we plotted the number of steps against the run number. As seen in Figure 8 there is no discernible trend in step recorded over time.

Several other interesting results presented themselves through the calibration process. As would be expected, the trials with the fewest runs (0-20 and 46-91) recorded the highest standard error. One unexpected result was the 0-91 trial. It was conducted 4692 times, the most runs of any trial, yet recorded a standard deviation of 58.2199 - roughly two times higher than the next closest trial (10-20, with a 30.0788 standard deviation). After noticing this peculiarity, we looked back at the data and remembered that an adjustment had been made to the system during the trial, and that a clear delineation in the values recorded by the data logger was evident. Thus, we present Figure 9 showing the split data histogram, with one cluster being the first 1204 runs, and the other being the remaining 3485 runs. When split, we see more uniform results, as shown in Table II.
These preliminary results, when adjustments are taken into account, show that the X-axis stepper movement is quite reliable and uniform in its movement, giving confidence when thinking ahead to the localization process. The final result for the steps per inch was found by averaging the averages, and taking the standard deviation. This resulted in $(104 \pm 6)$ steps per centimeter. Evaluating the data a second way, by first calculating the steps per centimeter for every run in each trial, then taking the average of this combined data yielded a result of $(103 \pm 6)$ steps per centimeter. This further uniformity further strengthens the reliability of the data.

### V. DISCUSSION

After conducting the calibration tests and calculating the mean steps per centimeter and the variance, we were able to use this information to write a program to control the absolute position of the carriage. When running this program we noticed that the carriage often travels past the desired location when moving to a given position, but upon return stops almost exactly where it began. This indicates that the steps per centimeter obtained is larger than the actual value. By running further tests and adjusting this parameter it is likely that the steps per centimeter will be more accurate and more precise (this hypothesis was supported in preliminary trials, with the most accurate distances being reported using about 95 steps per centimeter). This would theoretically yield a smaller error, which would improve localization. This inaccuracy is most likely a result of inaccuracies in the measuring methods used to obtain the carriage length and travel distance. To measure the length of a carriage a tape measure was held against the carriage, and the distance between the photo-interrupters was

<table>
<thead>
<tr>
<th>Trial</th>
<th>Dist.</th>
<th>Runs</th>
<th>Mean</th>
<th>StdErr</th>
<th>StdDev</th>
<th>SPI</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-91 S1</td>
<td>84</td>
<td>1204</td>
<td>21.177</td>
<td>0.4</td>
<td>15.1</td>
<td>252.1</td>
</tr>
<tr>
<td>0-91 S2</td>
<td>84</td>
<td>3485</td>
<td>21.309</td>
<td>0.1</td>
<td>5.7</td>
<td>253.6</td>
</tr>
</tbody>
</table>

TABLE II: Trial, distance traveled (in inches), number of runs, mean, standard error, standard deviation, and average steps per inch (SPI) for the 0-91 run, split to account for system adjustments.
measured. However, the carriage body is not very smooth, which made for a difficult time in accurately measuring the distance. The effect of this inaccuracy is difficult to predict, but is a systematic error. Because this length is subtracted from every distance measurement, it affects the shorter runs more greatly than the longer runs as the carriage length is a greater percentage of the recorded run distance. Another source of the observed inaccuracy stems from the inaccuracy of the ruler used to measure the total run distance. The ruler itself only had an accuracy of $\pm \frac{1}{8}$ inch, and was located on the front side of X-axis, while the limit screws were located on the back side of the X-axis. This gave rise to a visual parallax when trying to set the screws accurately. Because setting the screws used a bit of guesswork this is a random error, and probably has less of an effect than the first. A third source of error arises from the uncertainty of when the photo-interrupters are triggered. It was very difficult to tell how far the screw had to penetrate the photo-interrupter before being triggered. However, each photo interrupter body is approximately $\frac{1}{32}$ inch, so each photo interrupter contributes an additional $\frac{1}{16}$ inch error. This gives rise to a total additional error of $\frac{1}{8}$ inch. The largest improvement in our methods would certainly be improving the accuracy of the measurement techniques. By disassembling each component and measuring each piece individually a more accurate length profile of both the X-axis and carriage could be obtained. Also, by either using a different limit switch technique or placing the ruler directly under the limit screws, a more accurate run distance could be obtained.

VI. Conclusion

We have presented and discussed the system, methods, and results for calibrating a stepper-motor driven gantry system for the Autoponics project. We conducted 13 trials, and thousands of runs, recording distance, mean, standard error, standard deviation, and average steps per inch. From this data we were able to calculate an average steps per centimeter along with variance for the system, yielding two results: $(103 \pm 6)$ and $(104 \pm 6)$ steps per centimeter, which agree within error. We were then able to use this value to write a motor control program using absolute positioning. However, we are aware of certain areas of our experiments, particularly measuring accuracy, that could be improved to provide more accurate and reliable results.

Acknowledgments

The authors would like to thank Dr. Correll for his guidance and feedback throughout the course of the project, as well as our fellow group members and classmates for their valuable insights and suggestions.

References