HADRON MONITOR CALIBRATION SYSTEM FOR NuMI[*]

N. L. Muldrow[†], P. Snopok, Illinois Institute of Technology, Chicago, IL 60616, USA
K. Yonehara, Fermi National Accelerator Laboratory, Batavia, IL 60510, USA

Abstract

NuMI (Neutrinos at Main Injector) beamline at Fermi National Accelerator Laboratory provides neutrinos to various neutrino experiments. The hadron monitor consisting of a 5×5 array of ionization chambers is part of the diagnostics for the beamline. A gamma source is needed to calibrate the hadron monitor. We present the status and progress of the development of the calibration system for the hadron monitor. The system based on a Raspberry-Pi-controlled computer numerical control (CNC) system and position sensors would allow us to place the gamma source precisely to calibrate the signal gain of individual pixels. The ultimate outcome of the study is a prototype of the calibration system.

INTRODUCTION

We discuss an experimental setup to update the hadron monitor calibration system for the NuMI beamline at Fermi National Accelerator Laboratory (Fermilab). NuMI is a neutrino beam facility that began operation in 2005. NuMI is a conventional horn-focused neutrino beam designed to accept a 120-GeV proton beam from the Fermilab Main Injector accelerator. A simplified design schematic is shown in Fig. 1. The hadron monitor measures the spatial distribution of the uninteracted and undecayed pions produced by the main injector beam interacting with the carbon target. The hadron monitor is comprised of a grid of 1-mm ionization chambers orientated perpendicular to the beam direction and contained in an aluminum box. The calibration of the hadron monitor involves a radioactive source moved in front of the monitor by a motor-driven motion table to find the highest sensitivity position and observe the spatial sensitivity of the individual pixels [1]. The focus of this project is to create a system to control and display to the user the position of a radioactive source.

CALIBRATION SYSTEM SETUP

The prototype is based on a four-sided frame composed of 40 × 40-mm aluminum T-slotted beams 610-mm long. The beams are attached at the four corners using L-brackets. As shown in Fig. 2, the cross beam is attached to three set of rollers moving along the T-slotted framing beam. The two sets of rollers on the left side are held together by a custom square bracket providing extra rigidity. This configuration allows the cross beam to move back and forth along the frame with minimal friction and minimized unwanted lateral motion. This way, the motion of the cross beam can be controlled by a single motor.

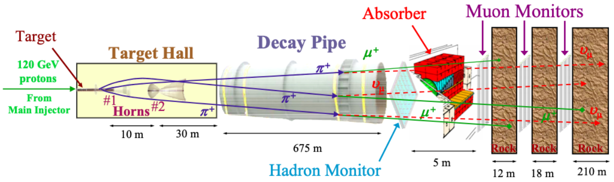


Figure 1: The major components of NuMI. The Fermilab Main Injector delivers protons to a carbon target producing pions which are then focused by pulsed horn magnets into the decay pipe. The hadron monitor measures the spatial distribution of any protons and pions left after decay. These are then absorbed by the hadron absorber [1].

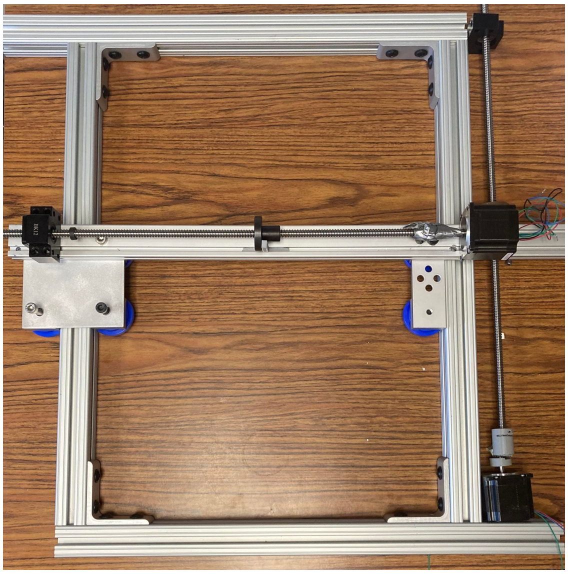


Figure 2: The experimental setup showing the frame, cross-beam, location of the two motors, and the set of the track wheels.

The setup uses two NEMA 23 stepper motors. The first motor is used to push and pull the cross beam along the frame. This motor is connected to a ball screw via a shaft coupler shown in Fig. 3.

As this specific coupler was not readily available, a CAD drawing was created to the project's specifications, and the part was printed using a 3D printer. 3D printing was chosen as the method of construction rather than a custom metal fabrication to reduce cost and time between prototypes. This coupler attaches the motor shaft to the end of the ball screw. A circular flange is screwed into place on the ball screw, and a custom-fitted bracket is screwed into place, as shown in Fig. 4. This piece slots into the bottom of the cross beam. Finally, a mount is placed at the other end of the ball screw, allowing it to rotate in place. When the motor is activated,

---

[*] Work supported by the U.S. Department of Energy under award number DE-SC-0020379.

[†] nmuldrow@hawk.iit.edu

ISBN: 978-3-95450-232-5 ISSN: 2673-7000 doi:10.18429/JACoW-NAPAC2022-MOPA66

Figure 3: 3D printed coupler fitted to motor and ball screw. Slits on both sides of the coupler provide flexibility.

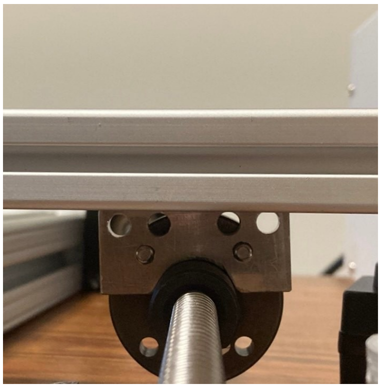


Figure 4: Machined bracket fitted to flange and cross beam.

the ball screw rotates, moving the flange and the cross beam. The second motor is placed on the cross beam. This motor is attached to a second ball screw via an identical 3D printed coupler. A flange is screwed into place on the ball screw, this time attached to a stabilizer as seen in Fig. 5. As the ball screw rotates, the flange travels along the length of the threads. To prevent flange rotation, a custom stabilizing piece was designed for the application and was 3D printed to reduce the time between design iterations and utilize a material with less friction. The stabilizer fits into the slot of the T-slotted cross beam and attaches to the flange via two holes. As the flange begins to rotate, the stabilizer catches it, stopping the rotation while allowing the flange to travel along the length of the ball screw and the cross beam. Finally, a mount is placed at the end of the cross beam to hold the ball screw while allowing it to rotate, as well as counterbalancing the weight of the motor. The motors are connected to a CNC board on a Raspberry Pi. Using the CNC board, the motors can be moved in controlled step sizes, allowing precise motion in the two dimensions. A singular command initiates the motors to move the flanges to a specific location inside the frame. Two distance sensors are utilized to check the motors' controls. One sensor is placed along each motor axis. The sensors read out the voltage that is converted into distance using a Python script. The input from the sensors can be used to pinpoint the source location within the frame, and as a safety mechanism in case one of the motors fails to stop for any reason. If the readouts are different than the intended CNC input, the problem can be acknowledged and acted upon. The details of the calibration of the motors and distance sensors are discussed below.

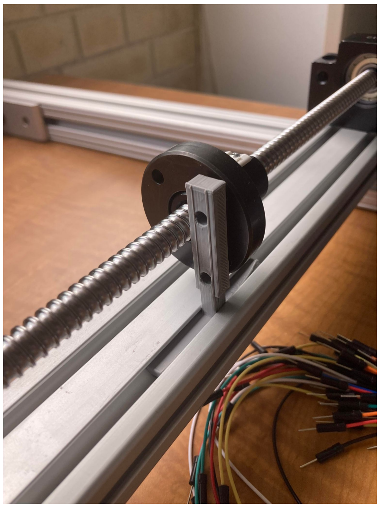


Figure 5: The stabilizer designed to slide along the cross beam. Holes in the element align with holes in the flange allowing the two to be connected. This prevents the flange from spinning with the ball screw while allowing longitudinal mobility.

bCNC SOFTWARE

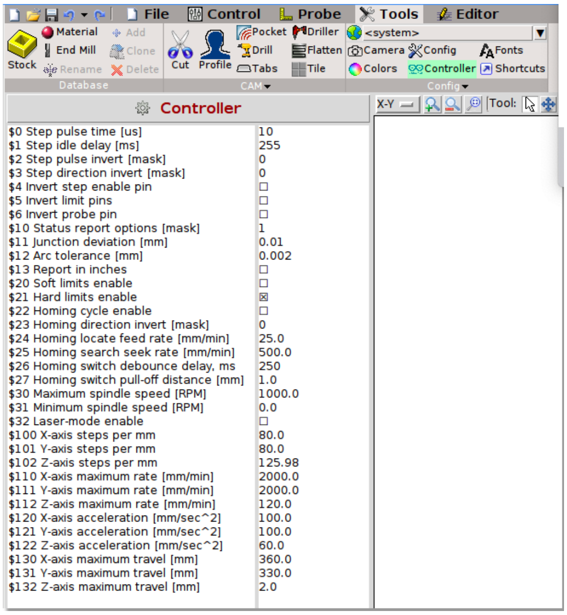
The motors can be controlled using the software bCNC [2], a cross-platform program written in Python to ensure communication between the hardware and an open-source GRBL CNC software. To operate the software, first, the controllers must be connected to the physical channels and connected to the software. A Protoneer RPI-CNC board [3] is attached to the Raspberry Pi using the 12 pins on the Raspberry Pi and is screwed into place. The motors can then be connected in the software using the menu shown in Fig. 6. Once connected and communicating via bCNC, the control panel can be used to adjust the step size of the motors for motion as shown in Fig. 7. The motor distance is measured in steps. The bCNC software must translate these step sizes into physical distances. To calibrate the physical distance moved by the motor, the size of the step must be determined. To do so, the motor is attached to a ball screw with a flange. The flange is marked at its starting distance and then moved a specified number of steps. The ending location is marked to determine the total distance traveled. This information can be used to determine the step size in mm, and then can be input into the bCNC software as shown in Fig. 7. The calibration of the motion from the motors can be verified using the position sensors as well, which will be discussed in the following

Status: Idle

WPos:	0.0	0.0	0.0
MPos:	75.0	-120.9	0.0

Port: /dev/serial0
Baud: 115200
Controller: GRBL1
☐ Connect on startup

Figure 6: The motors can be connected to the software through the bCNC.



Controller

$0 Step pulse time [us]	10
$1 Step idle delay [ms]	255
$2 Step pulse invert [mask]	0
$3 Step direction invert [mask]	0
$4 Invert step enable pin	☐
$5 Invert limit pins	☐
$6 Invert probe pin	☐
$10 Status report options [mask]	1
$11 Junction deviation [mm]	0.01
$12 Arc tolerance [mm]	0.002
$13 Report in inches	☐
$20 Soft limits enable	☐
$21 Hard limits enable	☒
$22 Homing cycle enable	☐
$23 Homing direction invert [mask]	0
$24 Homing locate feed rate [mm/min]	25.0
$25 Homing search seek rate [mm/min]	500.0
$26 Homing switch debounce delay, ms	250
$27 Homing switch pull-off distance [mm]	1.0
$30 Maximum spindle speed [RPM]	1000.0
$31 Minimum spindle speed [RPM]	0.0
$32 Laser-mode enable	☐
$100 X-axis steps per mm	80.0
$101 Y-axis steps per mm	80.0
$102 Z-axis steps per mm	125.98
$110 X-axis maximum rate [mm/min]	2000.0
$111 Y-axis maximum rate [mm/min]	2000.0
$112 Z-axis maximum rate [mm/min]	120.0
$120 X-axis acceleration [mm/sec^2]	100.0
$121 Y-axis acceleration [mm/sec^2]	100.0
$122 Z-axis acceleration [mm/sec^2]	60.0
$130 X-axis maximum travel [mm]	360.0
$131 Y-axis maximum travel [mm]	330.0
$132 Z-axis maximum travel [mm]	2.0

Figure 7: The step size of the motor can be calibrated and changed in the steps per mm line for each independent dimension.

section. With both motors connected, the $x - y$ position of the sample can be adjusted via the graphical user interface in which the user inputs the position in each direction in mm. The interface also shows the motion of the sample. This calibration was done multiple times for accuracy and showed repeatability and accuracy of positioning each time.

POSITION SENSOR CALIBRATION

Sharp analog infrared distance sensors are employed to control the axial motion of the motors. The sensors are accurate to ±2 mm for objects between 10 cm to 80 cm from the sensor [4]. These distance sensors produce a voltage output corresponding to some distance between the sensor and a target. Before use, however, these distance sensors must be calibrated. For calibration, the sensor is fixed in a location, and a simple rectangular target is moved fixed distances away from the sensor in one-inch increments. The uncertainty of the physical distance measurement was ±0.05 in, or approximately 1 mm. This procedure was repeated multiple times to ensure the accuracy of the measurements for both sensors used. The average output voltage from the sensor at each distance is then compared against the known measurements to develop a simple calibration curve. A sample curve for one of the distance sensors is shown in Fig. 8. This calibration curve is then implemented in a Python script to run in conjunction with the motor system. This ensures the accuracy of the distance for the CNC board and removes any additional uncertainty in the axial motion from the motors.

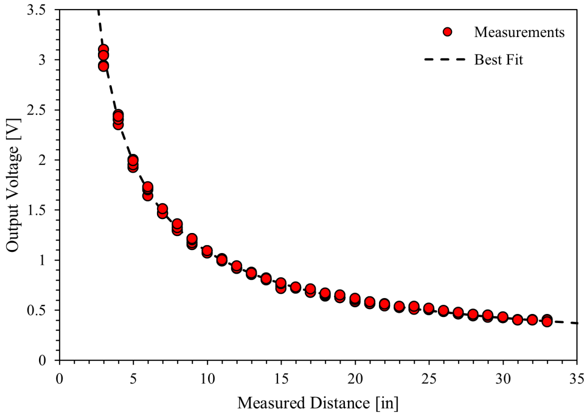


Figure 8: The average of the data acquired from calibration tests done to the sensor.

SUMMARY

In summary, this paper presents the design and fabrication of a prototype for the calibration system for the hadron monitor of the NuMI beamline. This project was designed to introduce a system of checking the location of the radioactive source, which is moved by a combination of motors used in the calibration of the monitor. This was accomplished by introducing and calibrating two infrared sensors to measure the distance of the stabilizers moved by the motors. In the future, further testing of the scaled system will be performed and a similar system can be implemented to calibrate the hadron monitor of the NuMI beamline.

REFERENCES

[1] R. Zwaska, "Accelerator Systems and Instrumentation for the NuMI Neutrino Beam", Ph.D. thesis, Phys. Dept., The University of Texas at Austin, 2005.

[2] V. Vlachoudis, "GRBL CNC command sender, autoleveler, and g-code editor". `https://openbase.com/python/bCNC`

[3] `https://wiki.protoneer.co.nz/Raspberry_Pi_CNC`

[4] Sharp GP2Y0A21YK0F Data Sheet, Sharp, Dec. 2006, `https: //www.pololu.com/file/0J85/ gp2y0a21yk0f.pdf`