

# Prototyping of Superconducting Cage for nEDM Measurement

Rohan Rauch

July 9, 2022

## 1 Introduction

The electric dipole moment (EDM) is a property of particles characterized by electrically positive and negative poles, that defines their torque in electric fields. One of the questions currently being investigated in particle physics is whether the neutron has a non-zero electric dipole moment. Such a discovery would have important ramifications due to the violation of charge and parity symmetries by the neutron's EDM (or nEDM). One illustrative example of this violation is that a time-reversed version of a neutron would have its magnetic moment change direction, but not its EDM, leading to a non-symmetric situation with the initial system in relation to T (time) symmetry, which implies a CP violation. This CP symmetry violation is especially important in regards to baryogenesis, the theorized process by which the early universe had more matter than antimatter and thus, after the two had interacted and nullified each other, was left with the matter that would become our material universe.[2] According to the Sakharov conditions, a set of conditions that allow for this baryon asymmetry, a violation of CP symmetry is required of sufficient magnitude that the estimated difference in matter over antimatter was produced. While sources of CP symmetry violation are already known of, they are much too small to explain baryogenesis, and observing a non-zero nEDM may provide the required CP symmetry violation to satisfy it.[3] Experiments to measure the nEDM at Oak Ridge Laboratory's (ORNL) Spallation Neutron Source (SNS) involves inducing Larmor precession of the neutron spin in a strong electric field. While the outside magnetic field will be small, the neutron's large magnetic moment will create a precession in the field from this spin, and the resulting modulated signal can be detected to measure the nEDM.[2]

The requirement for constant magnetic fields within the small size of the cryostat in which the experiment is being performed could be fulfilled by designing a superconducting cage. Due to the ability of superconductors to maintain a current for an almost indefinite amount of time, exposing such a material to an outside B field would induce a current by Lenz's Law that opposes any change in the B field; thus, the B field would be maintained even when the external source is removed. Forming the superconductor into a cage allows for multiple current

loops to form around each hole in the cage, “pinning” magnetic field lines such that they stay constant over space and time within the cage[1], thus creating the necessary field for the nEDM experiment that remains unaffected by external B-fields. Using a prototype superconducting cage design composed of lead, this project will accurately obtain temperature and magnetic field strength data on the cage during testing in liquid helium, and thus determine whether the design is able to produce the desired fields.

## 2 Procedure

### 2.1 Cage Design

The first step of prototyping was designing the superconducting cage, which occurred before I joined the project. The cage was constructed of leaded PCBs with a grid of small holes, after which multiple pieces were soldered together to create the cage itself. This apparatus was placed on the end of a long metal rod with a hollow interior, so that it could be safely lowered into a dewar holding the liquid helium during testing. Two thermometers and a magnetometer were attached to the cage and had their cables run through the rod. Each thermometer was placed on the farthest and closest face of the cage, to be able to measure when the front and back of the cage reached the critical temperature and thus confirm the cage is superconducting. The thermometers and magnetometer are attached to separate, external control devices that convert the data from the sensors into voltages, which are then sent to a National Instruments DAQ device (an NI 9239) that converts the two data streams into a single data stream that can be read by a computer through a single USB port. Note that the two thermometers input into the same converter, but only one of their signals is outputted at a time; during testing, we will manually swap between the two to check both thermometers.

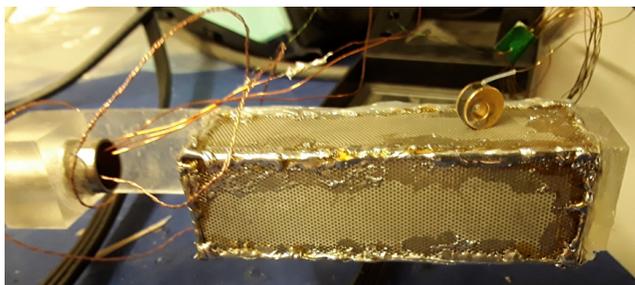


Figure 1: Built version of superconducting cage. One of the thermometers is on top of the cage in this image; this is not where it will be attached during testing. Magnetometer is not shown as it is inside cage, and other thermometer is outside the image bounds.

## 2.2 Data Acquisition Software

While the NI 9239 is capable of storing data from the sensors into a buffer, this buffer cannot be read by a computer without the use of a program. Originally, this was performed using a combination of DAQExpress, a program which contained the NI-DAQmx driver necessary to access the NI 9239, and a previous student's executable version of a C program, called from a separate MatLab program. The executable generated a text file list of 2000 data points, every four data points being a list of the readings from the NI 9239's four channels at some unknown time, and the MatLab program reorganized this list into a clearer format, with 500 lines that each contained four data points from all of the channels. This set-up was unusable, as we could not know when each data point had been read, nor be sure of how consistently data was being taken. While we attempted to investigate whether we could average the data from the output, and thus loop the calls to the executable, it was decided to instead rebuild the code anew ourselves.

The new data acquisition program was written in Python, utilizing the `nidaqmx` library to communicate with the NI 9239. This library was specifically used to create an object representing the device and its channels, setting its sample rate of the sensors, and requesting to read the data in the buffer. The program operates by first creating an interpolated curve function translating voltage signals to temperature, based on a document describing curve points for the outputted voltage signals from the converter equipment when receiving the thermometer's temperature readings. Then, the program asks the user to provide a sample rate in samples per second, as well as the number of samples that will be gathered. Next, the program starts a for loop that will count up to the requested number of samples, assuring that this many samples are gathered. Within the for loop, a while loop is used to wait until a specific time period has passed since the last loop iteration, equal to the reciprocal of the sample rate. When this amount of time has passed, a thread is started that contains the actual read operation, and the next iteration of the loop begins. The thread contains the function call to read all data from the buffer but not saving this data (which clears any accumulated data in the buffer), read the data from the buffer (as a new set of data will be in it already), and write the gathered data to a text file in a readable format along with a timestamp of when the computer requested the data. Additionally, the thread has function for translating the voltage signal data from the thermometers into a temperature using the previous interpolated curve function, and notifying the user when this temperature has gone below the critical temperature (set in-code) or returned to above it. After all iterations of the loop have been processed, the program ends, and a text file is left containing all the data collected.

During the process of developing the program, certain decisions were made to alleviate issues and errors. One of the more significant was the original usage of MatLab integration, through a MatLab engine, within the Python code to call a MatLab program when converting voltage signals from the thermometers to temperatures. This was later removed to reduce complexity and because it

was believed to be causing a reduction in performance. The addition of clearing the buffer through reading all available data without storing it was implemented to prevent an error caused by the mismatched rate of how fast data was being added to the buffer and how fast data was being read (and thus deleted) from the buffer; clearing the accumulated data help prevent the buildup of entries that caused the error. The last changes worth discussing are related to performance. It was noticed that the program would take longer than expected to read an amount of samples, especially at much higher sample rates; 10000 samples at 1000 samples/sec, expected to take 10 seconds, took 140, while 100 samples at 10 samples/sec took 10 seconds as expected. By changing the read operations to be executed in a thread instead of within the for loop itself, performance was massively increased, reducing 10000 samples at 1000 samples/sec to 40 seconds. The remaining performance issues were believed to be caused by the computer's CPU itself, as during the while loop the CPU was likely getting overwhelmed and would hang temporarily before continuing the program, leading to the timing discrepancy we were seeing. As this issue was occurring with the CPU itself and would happen regardless, there was no change made to fix it. Because the performance issues only became problematic at high sampling frequencies, and we would be testing at low frequencies, the performance issues would not affect us.

### 3 Future Work

Two sets of tests are planned for the superconducting cage prototype – one using liquid nitrogen, and one using helium. Both tests will consist of dipping the cage into a container of cryogenic liquid, using the rod to support the cage and keep it stable within the liquid. Sensor readings will be taken from the magnetometer and thermometer, using the NI 9239 and the Python code written for it. We will also expose the cage to an external field magnetic field, though the method by which this will done has not yet been determined.

The liquid nitrogen test consists of a small container of liquid nitrogen being used as the cryogenic liquid. Due to the small size of the container and the length of the rod that the cage is attached to, a wooden stand was built to hold up the rod during testing. Liquid nitrogen is only 77 K, so the lead of the cage will not reach the critical temperature of 7.2 K, and thus won't be superconducting. Using liquid nitrogen instead allows for the identification of any issues with the design, more specifically any materials that do not stand up under cryogenic temperatures.

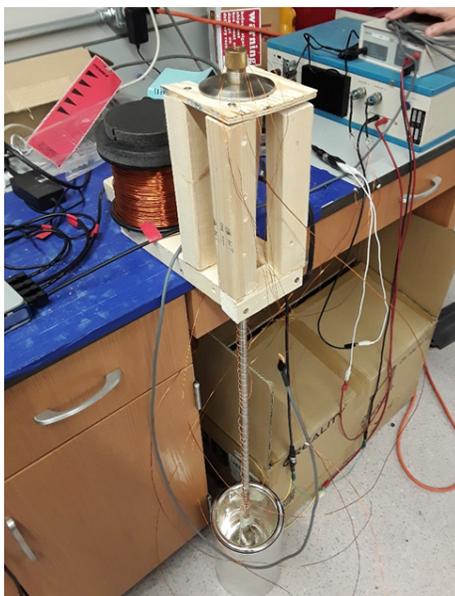


Figure 2: Set-up for liquid nitrogen test. Wooden platform in center holds the rod attached to the cage. Container of liquid nitrogen is at bottom, with cage inside of it. Copper wiring is a counterweight and otherwise unused. External converter equipment is shown at top right, and the edge of the NI 9239 is shown at middle left.

The liquid helium test consists of a large dewar of liquid helium being used as the cryogenic liquid. The length of the rod is such that it can be threaded through the hole in the dewar and let the cage reach the liquid inside, while keeping the whole container sealed. Liquid helium is 4.2 K, which allows the lead to reach critical temperature and for the whole cage to become superconducting. This test will thus consist of waiting for the cage to reach the critical temperature, then use an external magnet to see if the cage can create the induced, opposing B-field that it was designed for. The creation of said B-field will confirm the viability of the design in the SNS nEDM experiment.

## References

- [1] F Gömöry. *Superconductor Dynamics*. 2014.
- [2] J. S. Nico and W. M. Snow. “Fundamental neutron physics”. In: *Annual Review of Nuclear and Particle Science* 55 (2005).
- [3] E. Sather. “The mystery of the matter asymmetry”. In: *Beam Line* 26 (1996).