Control system for the TAMUTRAP beamline and Penning trap

R. Burch, V.S. Kolhinen, D. Melconian, M. Nasser, A. Ozmetin, B. Schroeder, and P.D. Shidling

TAMUTRAP is currently preparing itself for the installation of its newly designed Penning trap [1]. As we wait for the RIBs to become available, we are commissioning TAMUTRAP by measuring the masses of stable ^{23}Na [2]. For this, we need to scan the excitation of the ions with respect to the frequency of the RF applied to the trap. The present report discusses the progress on automating these scans.

The present setup in terms of instrumentation control and data acquisition is one that gets the job done, but not as efficiently as it could be. Much of our equipment is initialized for experiments via a mixture of hands-on adjustments, LabVIEW, and 3rd party software. The computer-controlled components are done primarily on a Windows computer. This is only worth mentioning given that we acquire data through a separate Linux computer utilizing our ROOT driven data-acquisition software.

The control system for TAMUTRAP is a two-phase project. In the first phase, we will automate the scanning process so that it does not have to be continuously tended to (every 2-3 minutes.) This is done over a length of typically 30 to 40 minutes. We have begun this by programming a Python script that will handle all processes through TCP connections. This design choice is powerful in that it allows us to communicate not only with instrumentation controls such as LabVIEW, but also other computers involved in the scanning process. This removes complications from running the system across two operating systems. In addition to improving the efficiency of time, such a system will allow us to preform many short scans instead of one long one for each frequency, thereby reducing the effect of drifts in the beam current and RF power. Another bonus is that the TCP communication method means we can run the 'handler' on either the Windows, Linux, or any other computer on the network in our offices.

Phase two is to merge and rebuild our many means of instrumentation control into fewer entities. The 3rd party software's in use are convenient, but can with some fortitude be replaced by LabVIEW scripts. From here we will combine our numerous arrangements into one grand LabVIEW controller (with only a few other less complaint to upgrade mechanisms). At this point we will have two main components for experimentation: the LabVIEW master controller, and the Python scan handler.

As it stands, we have our 'handler' operational, but is still accepting improvements. Its current operation is updating scan frequencies and running a placeholder Python scan script which will be replaced by the data acquisition system as soon as it's done upgrading to become compatible with the 'handler.'

[1] V.S. Kolhinen *et al.*, *Progress in Research*, Cyclotron Institute, Texas A&M University (2017-2018) p. IV-42.

[2] V.S. Kolhinen *et al.*, *Progress in Research*, Cyclotron Institute, Texas A&M University (2017- 2018) p. IV-39.