**Running Manual Control (ManualControlV6.4.21.py)**

Program written by: Matt Bisbee at NARS Lab Ohio State University under Dr. Raymond Cao

Instructions Date: December 16, 2021

Prerequisites to Running the code:

* If this is the first run on a computer, please see installing NARS Auto nCT instructions
* Ensure Micro-Manager (MM) and pycromanager is up to date on the nightly build.
  + Go to <https://micro-manager.org/wiki/Micro-Manager_Nightly_Builds> to download most recent build of MM and within a command line, type “py -m pip install --upgrade pycromanger”
* Find what communication port the stages are connected to by going to device manager and finding the related “Port” information, the default values for communication ports for camera, rotation stage, and XYZ stage are listed in the entry boxes already. This would most likely change if running on a different computer.
* If running from .py, ensure folder has the following: FewViewPart2.py, LeftoverTomography.py, NewCameraControl.py, SMC100Final.py, XYZMotion\_Station.py, OSU\_NRL.ico, OSU\_NRL.png
* This code is written on a Windows based machine and only tested for Windows OS, adjustments would have to be made for other forms of OS

Running the Code:

1. Open Micro-Manager 2.0 Gamma and connect to the appropriate camera. Once this is open correctly, you may run the ManualControlV6.4.21.py file through command line or code editor
2. This code is utilized to control the three main peripherals separately best served in alignment and exposure time testing. This part of the package is not necessarily used for tomography purposes but any experimentation to get radiographs can use this code.
3. The top section of the code has the camera control area:
   1. First line has the camera port and a button to initialize the camera, if MM is running and the camera is connected properly, once this button is pressed it will be replaced saying it is connected
   2. The user can specify File name, file location, suffix, exposure time in seconds and EM gain if using EMCCD camera (if not using EMCCD or using gain feature of EMCCD, leave value as 3 and it will ignore this).
   3. The suffix is present to help ensure that images have a unique name as there are no fail safes for overwriting other images saved at that location. One can change the suffix and after taking an image it will start counting up by 1 from that value.
4. The second is used to control the rotation stage:
   1. The user must press the initialize button which will test that the rotation stage is connected properly. It will show up green if connected or an error will populate in the general frame at the bottom.
   2. The user can reset the stage to 0 deg with “home stage” button or stop communication with disconnect button. The current position and speed will populate when those values are changed through the buttons on the right.
   3. To move a set number of degrees from the current position, the user should use the move relative command. Positive for CW and negative for CCW.\*
   4. To move to a specific degree, the user should use the absolute move command.\*
      1. \*The stage is set up as homed at the zero position with movements -180 to 180 degrees. There is not continuous rotation so if the user wants to move in increments such as 150 160 170 180 190 the equivalent degree for “190” is -170 which can take some time getting used to.
5. The third section is used to control the XYZ linear stage:
   1. This section is quite dense with buttons and positions as it includes movements for all 3 axes as well as position in mm and inches. Current movement ranges are listed in the control xyz section and due to the constraints of the imaging station the Z direction has a hard coded limit of 3.543 inches as opposed to the over 5 inches of travel it could do if unrestricted. This limit could be changed by modifying the check limits section of XYZMotion\_Station.py.
   2. The connect button must be pressed and subsequently turn green to control the stages. One can disconnect the XYZ stage with the stages being at any position but **Note: The zero position of the stages is currently defined as fully back away from beam exit for X, fully to left of beam for Y and fully in lowest position for Z. If the controllers are unplugged/turned off not in these positions, it will reset the zero as wherever the stage is at. The easiest way to reset the zeros at the desired positions is to turn on the controllers, manually move the stages with the job buttons on the controller to the correct positions, then turn off the controllers.**
   3. The home button will take all three stages to 0 position, currently so will the “Go to Setup Positions” button as they are one in the same. This could be altered for a different system setup so the button was left in and the coded value could be adjusted.
   4. The experimental positions for our system are listed in the Status of XYZ section, if the user presses the “Go to Experimental Positions” it will send the stages to these spots”
   5. If the user finds that an experiment has different experimental positions, they can select the “Set Positions as Experimentals” to update the values at the bottom of the Status of XYZ section. This will not change those values for later times the code is started up, to change them for startup, the entry values will have to be altered in the ManualControl code
   6. In the Control XYZ Stage section, one can zero, move relatively from current position, and move to a specific position for each axis. The user can shift between mm and inches with the drop down button and then press move. The current position of the stage will populate on connection and update after each movement.
   7. The current motor speed is set as 2000 steps/s which has been a reasonable movement speed that does not jostle the sample. If the user would like to decrease the movement speed on any axis, they can select the axis from the drop down, enter the speed in the entry box, then press set.
6. The bottom frame has a button to home the stages and close out and some messages
   1. The general messages section will update if there are any errors the come up. This could be an error in connection of a stage or an error in a movement attempted on a stage.
      1. The most common error for the XYZ stage is that there are many hard coded limits for these positions due to the presence of the light tight box and sample holder that would hit each other if the wrong movement is sent. Currently, the Y direction movements are very limited until the X stage has moved passed 80mm. Then, the Y stage can be moved to its furthest position. Once the Y stage is passed a specific value, the X direction cannot move further back. The Z position cannot move too far up as the rotation stage would hit the bottom of the LTB area so it is hard coded to stop at 3.5 inches.
   2. Movements will print in the second line for reference as well as save file information for the previous image

Notes:

* A log file for the movements and saves will be created upon starting of the ManualControl file so the user can go back and see what movements relate to what saves.
* The contrast of most photos is quite low if the exposure time does not fill the bit depth. To see the images as they are taken, the contrast is increased by the value below the image and can be adjusted to best see the phantoms.