AFM Lab Log

Session 1 (20/06/2025)

Objectives:
The instrumentation/acquisition AFM system has been built over the last semester; This session consists in a thorough testing of the build, and will provide a protocol for the calculation of the calibration factors and an estimate of the resolution/error.

<u>Procedures:</u>

- **Calibration Factor Calculation (A):**
    1. Set up your AFM for acquisition (laser alignment, autotuning, etc…);
    2. Acquire an AFM image;
    3. Use manual control in the acquisition/instrumentation system to displace X and Y by S steps (for example, 32 steps). Note: the AFM image size must be chosen in agreement with S: if S is large, your image will have to be larger. Experiment for some image sizes to see the optimal image size in order to allow for overlap of distinguishing features.
    4. Return to point 2, and repeat until this step. Once you have acquired a fair amount of images (6 images should suffice), proceed to 5.
    5. Using all the acquired images, choose a pivot point which is present on all images (or at least between each pair of images). Measure the distance (horizontal and vertical !!) from one of the image corners (always the same corner for the rest of the procedure) to that object (the object must be visible and well defined, within reason); Compute the difference - dX and dY. Incorporate an error estimate if need be (the object not being well defined). Write down the differences.
    6. Now you may compute the calibration factor from Steps to Nanometres (DeltaNm/S) (FOR EACH DIRECTION), with an error estimate (max(mean error, {individual errors})). To calibrate the acquisition system, simply introduce the two (S,DeltaNm) pairs (both x and y!) per measurement (up to 6 measurements, as originally coded (can be changed by editing the source code!)). The mean between the factors is used by the system to acquire tiled images.

- **Calibration Factors - Correlation between X and Y (B):**
    - For a value of S, repeat the previous experiment but only in one direction, two times (1 for X and 1 for Y). Check if the results are different (does the movement in the X axis affect the Y axis?).

- **Motor kickback (C):**
    - For an estimate of motor kickback effects/mechanical imperfections in the system, simply choose a step S, and go back and forth (+S, -S, +S, …), to see the movement of a pivot point throughout. Proceed with usual error calculations to obtain a dKB. Do this experiment for X and Y independently.

- **Image Step Bounds for tiled acquisition (D):**
    - Simply repeat the ‘Calibration Factors’ procedure multiple times with different values of S. Compute the errors, and plot them.
    - This may be used along with **(C)** to set a limit on the imgStep for tiled acquisitions:

Further Experiment Proposals: Local portion of cog vs calibration factor (axis wobble error estimate).

<u>ACQUIRED DATA:</u>

Note: 001-009 are old acquisitions which were not documented (not useful).
(A/D) Documents->PIC_JC->pic.005-010. S=32. Imgs on AFM computer, Transferred to PC w/ USB pen drive.
(A/D) Documents->PIC_JC->pic.011-014. S=128. Imgs on AFM computer, Transferred to PC w/ USB pen drive. Note: original plan was to acquire up to 016, but along the path located a large impurity which risked breaking the tip; I decided to abort this sequence and just work with 3 data points for the calibration estimate.
<u>Estimated time for this acquisition</u>: 2 hours.

NEXT ACQUISITION PLAN [20/06/2025]
(A/D): S=8, S=16, S=64; (C) for one of the S values; (B) for one of the S values; Additional experiment: engage multiple times on the same spot to measure error intrinsic to the AFM system -> incorporate it for the calculation of the calibration factors. <u>Estimated time for all the experiments</u>: 6 hours.

Session 2 (21/06/2025)

Objectives:
Continuation of session 1's experiments, as stated in its end.

<u>Procedures:</u>
Same as before. Included experiment to test AFM intrinsic error on engage **(E)**: simply engage multiple times and see if objects move. If they move substantially, include this error in the measurement of distances for the previous experiments.

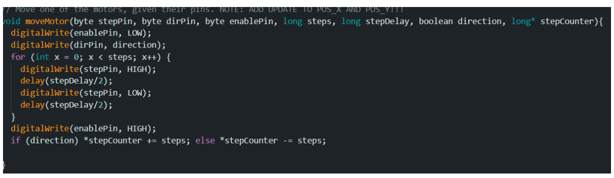
<u>ACQUIRED DATA:</u>
Last session's data seems to be inconsistent. Possible source: noise and bumping into the AFM table (unlikely, i am careful…). This time I will be even more careful. I will also do each direction independently and then do a test run for two directions at the same time (I suspect the correlation may also have resulted in the observed “noise”).

What will (hopefully) be gathered (adapted from last session due to time constraints):

- [x] ~~3 imgs of engage in the same region~~
- [ ] ~~S=16, dir=X, imgs=4~~
- [ ] ~~S=16, dir=Y, imgs=4~~
- [ ] ~~S=32, dir=X, imgs=4~~
- [ ] ~~S=32, dir=Y, imgs=4~~
- [x] ~~S=64, dir=X, imgs=4~~
- [ ] ~~S=64, dir=Y, imgs=4~~
- [ ] ~~S=128, dir=X, imgs=4~~
- [ ] ~~S=128, dir=Y, imgs=4~~
- [x] ~~S=+-..., dir=X, imgs=4, KICKBACK~~
- [ ] ~~S=+-..., dir=Y, imgs=4, KICKBACK~~
- [ ] ~~S=..., dir=X+Y, imgs=4 **- Don’t need to do it if Y is not viable…**~~

Imgs on AFM computer, Transferred to PC w/ USB pen drive.

Documents->PIC_JC->pic.015	200nm image to show AFM capabilities.
(E) Documents->PIC_JC->pic.016-018	Multiple engages w/ engage 200nm-(zoom)>1um-(zoom)>10 um. At first glance, there seem to be some differences, up to 1um … If it had problem keeping track of positions when zooming in/out, the expected error would have been greater, ~10um (img size).
Documents->PIC_JC->pic.019	Original 020, however gear wasn’t properly placed, so I use this image as an example.
(A) Documents->PIC_JC->pic.020-023	10um img size, S=64, dir=X.
(C) Documents->PIC_JC->pic.023-026	10um img size, S=+-64, dir=X, KICKBACK EXPERIMENT. -S(24)+S(25)-S(26)



```
// Move one of the motors, given their pins. NOTE: ADD UPDATE TO POS_X AND POS_Y!!!
void moveMotor(byte stepPin, byte dirPin, byte enablePin, long steps, long stepDelay, boolean direction, long* stepCounter){
  digitalWrite(enablePin, LOW);
  digitalWrite(dirPin, direction);
  for (int x = 0; x < steps; x++) {
    digitalWrite(stepPin, HIGH);
    delay(stepDelay/2);
    digitalWrite(stepPin, LOW);
    delay(stepDelay/2);
  }
  digitalWrite(enablePin, HIGH);
  if (direction) *stepCounter += steps; else *stepCounter -= steps;

}
```

Seems there is huge kickback (the motor moving in reverse seems to NOT return to the original movement)… Do we need a delay after setting the dirPin??? Very possible… This would explain the problems in the original paper…

UPDATED CODE TO TRY WITH DELAY BEFORE STARTING TO MOVE…
Visually, this fixed the kickback issue! It seems the original code of the paper (which I took a large inspiration from when writing the programme) was wrong! Let's repeat the experiment with the corrected code

```
digitalWrite(enablePin, LOW);
digitalWrite(dirPin, direction);
delay(2*stepDelay);
for (int x = 0; x < steps; x++) {
```

(C) Documents->PIC_JC->pic.027-030 10um img size, S=+-64, dir=X, KICKBACK EXPERIMENT 2.
(27) +S (28) -S (29) +S (30)

The Y Motor is not properly set up mechanically. We will proceed with the analysis ignoring the Y movement, as it is not consistent.

For 64 steps, we almost span the entire image size, so I won't test for S=128.

It seems that the X motor does not allow for microstepping. When I step 16, then another 16, I only see one movement. Since I am microstepping 1/32, it seems that this motor is not capable of microstepping. As such, I don't see much interest in trying out S=32. The resolution experiments are left as a recommendation for future builds with other motors (possibly NEMA-17).