# **Weekly Report – W10 Spring 2023**

## **Task & Problem**

1. Verify the relationship between torque/force and pressure input in the chambers;
2. Mount IMUs onto the end effector of the arms, and test if they can work properly;

## **Solution**

1. Verify the relationship between torque/force and pressure input in chambers

Since our safety pressure is 65 psi temporarily, which is about 4.5 bar in the simulation package, we have the deformation for two axial directions, x and y, any pressure input value beyond 4.5 bar is not so meaningful, and the effect caused by external force exerted on the SRA tip is pretty straight forward, with the same amount of force or torque in opposite direction, the deformation should be opposite as well. So we can first compute the maximum deformation (displacement) and find a proper amount of force or torque (if we can achieve the same maximum deformation with 50 N, we don’t have to test for 100 N). And the maximum displacement is simulated to be 0.0558 m in x or y direction.

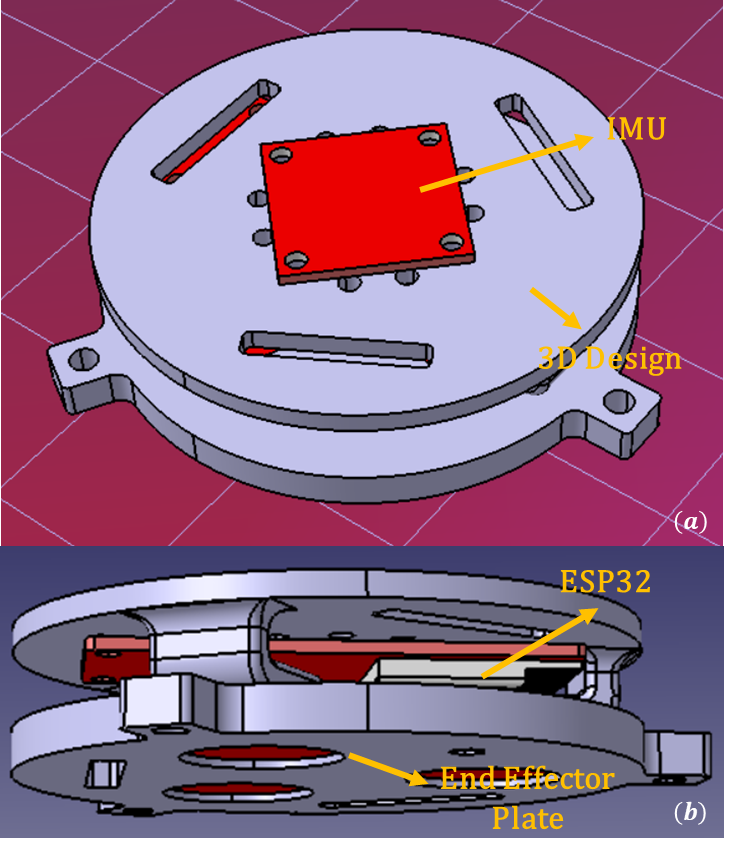
However, it turns out that only about 3 N exerted in x or y direction could cause 0.0558 m displacement, and the displacement in z direction does not coincide with the pressure supplied case, which means that we have to supply additional force in the z direction. But it was not such easy as I thought, by simply adding a constant or a random linear changing force in z direction, it will cause unexpected displacement in the rest two axial directions. So we have also to discover the proportion of force in z and y (or x) direction.

Also, considering the situation that in the real experiment, 65 psi could make the arm bend but in the simulation it could only make the arm bend slightly, which gradually raises my doubt. I will look for some more scientific methods in research papers in the following weeks.

1. Mount IMUs onto the end effector and test their performance

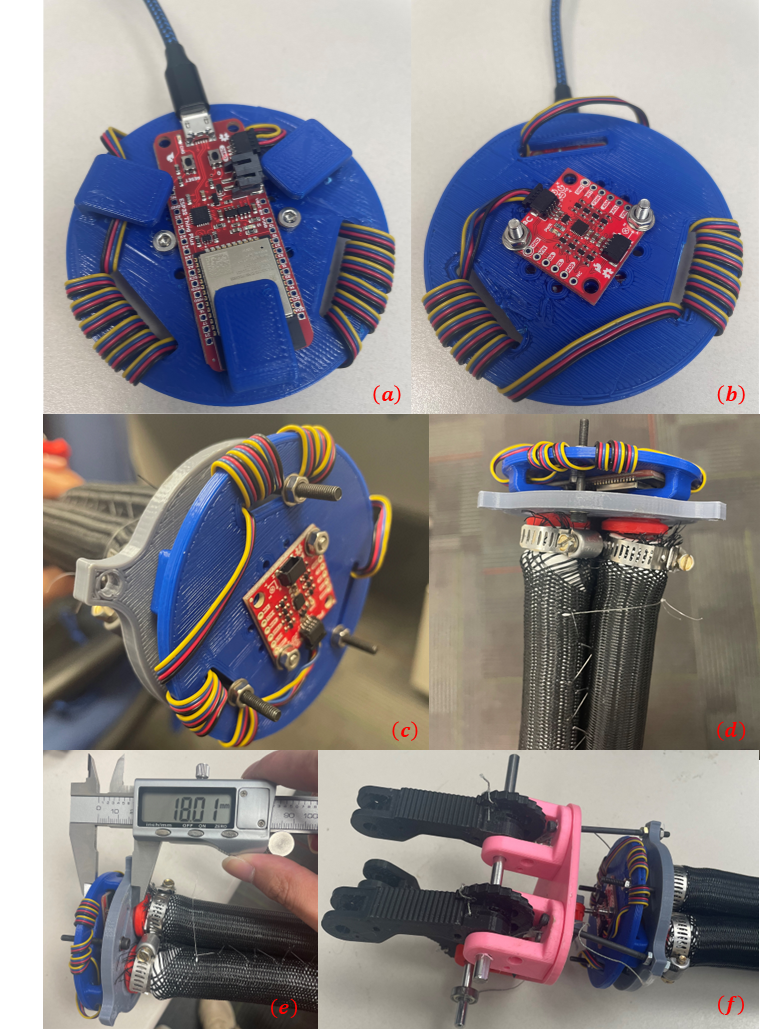
#### (1). 3D design

This week considering the properties of the end effector, we need additional auxiliary 3D printings to mount IMU and ESP together onto the end effector. To fully protect the IMUs and ESP32s from the possible damage caused by collision during the real experiments, the first priority is not to let the IMUs and ESP32s be exposed to the space that could contact with any objects for the gripper to grasp (like the handrail in our experiment). So a rough 3D design is shown as follows.



**Fig. W10-1** The conceptual assembly diagram for the IMU, ESP32 and end effector of the arm. (a) top view of the design, (b) side view of the design.

After the 3D printing parts were ready, the final effect is shown in a series of diagrams shown below, the designed part worked perfectly, in (a) and (b) the ESP32 and IMU are mounted on the different sides of the plate, and meanwhile the redundant wire is twined through the groove to decrease the chance of wire contacting with the environment as much as possible to avoid interconnection issues during the experiment.



**Fig. W10-2** The read effect when the IMUs and ESP32 mounted on the end effectors through 3D printing parts. (a) bottom view before mounted, (b) top view before mounted, (c) front view after mounted, (d) side view after mounted, (e) dimension of the additional 3D printing part occupies, (f) effect after gripper being mounted.

When the plate is mounted onto the end effector, the space it occupies is so small that it will not affect the assembly of the gripper as shown in the figure above (e) and (f), whether we need to add some other protection to the electronics depends on the final state of the gripper. I will consider about it in the future.

#### (2). Data collection and verification

The hardware issue can be a unstable factor during the experiment, after the plate mounted on the end effector for the first time, I cannot read any data from one of the IMUs, below are the different indications in the serial monitor of Arduino IDE.



**Fig. W10-3** The normal output of IMU and ESP32



**Fig. W10-4** The abnormal start-up of ESP32 and IMU

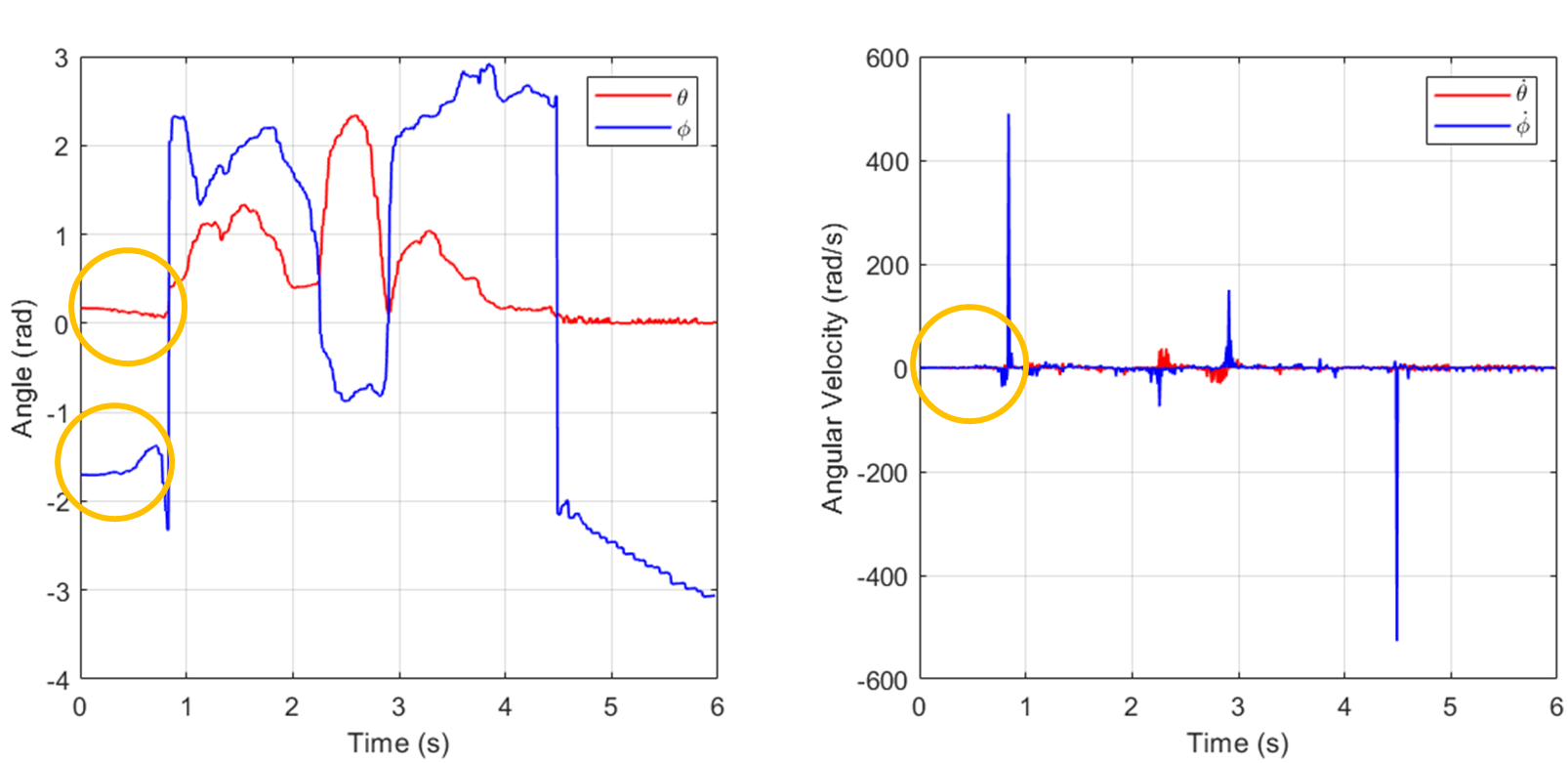
Because in MATLAB temporarily there is no way to read the setup information before showing up the stream data, for the ESP32 and IMU from which we cannot read data, the start up code is shown in the figure above, I have consulted some trouble shooting documentations online, if it is indeed an error, it will directly tell us that something failed, but it seemed that there are no errors for our case, and since no one could answer my “question”, the possible reasons are summarized as follows,

* I have test the data collection on two PCs on the first day the IMUs mounted onto the plate, I recalled all the operations I did, I think the only thing could be that I pulled out the wire of the mouse and connected it to another one, after that I connected the one of the IMUs to the port used for mouse (for both PCs), though there wasn’t any port conflict information, there is quite a large possibility that the port for mouse previously was still being occupied; however, it still didn’t work after rebooting the PC;
* The hardware issue can be excluded firstly actually, since the power LED was on, if the ESP32 was destroyed during assembly, the IMU light will not be on;
* Another possible reason might be that the buffer (cache) has not been cleaned up since I received the devices from Manu, so I add a line of code to clean it after the data collection process is terminated.

The most weird thing is that for the second day, I didn’t do anything based on the experiment results on the first day, I only connected the IMU from which we cannot read data to the ESP32 which is good to read data, just to test if it is the problem of the IMU or the ESP32, it worked, and when I connected it back, it worked as well. So the conclusion is that it is mostly like for the ESP to have some bugs, it can successfully connected with the PC and IMU, but it seems like to be in sleep mode, maybe next time we can try to press the “reset” button to see if the problem can be solved. I will leave it as an open question.

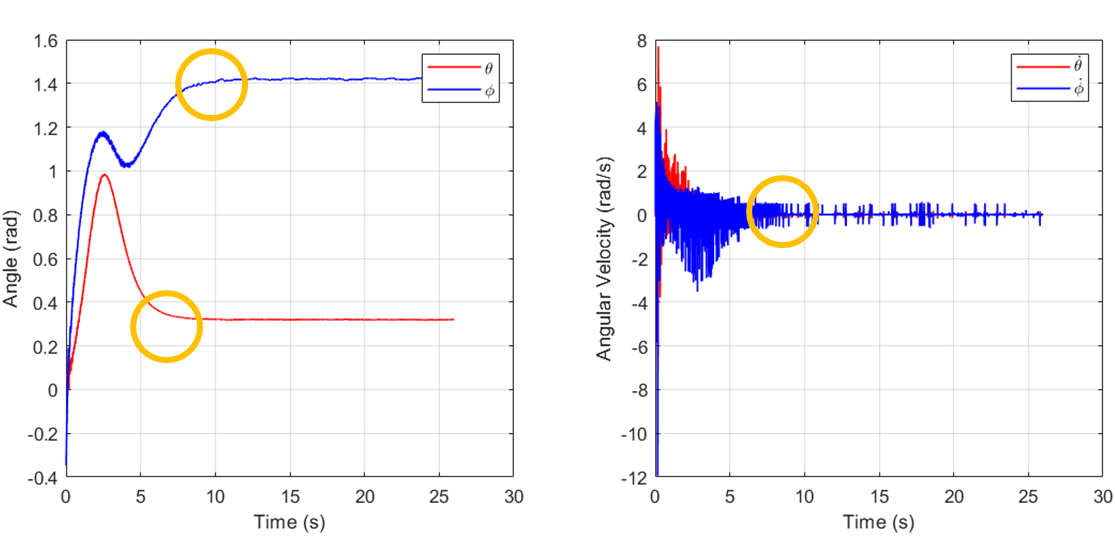
And back to the data collection, probably due to the wiring approach in Fig. W10-2 , when power is on, there will be several magnetic fields generated, and it will take a few seconds for the date reading to stabilized. To clarify what kind of difference after assembly, we shall make a comparison of the results before and after.

As we can see in the diagram below, for a random test with IMU and ESP32 not being mounted on the end effector, no matter for the angle or the velocities, all of them were very stable at the beginning of the test.

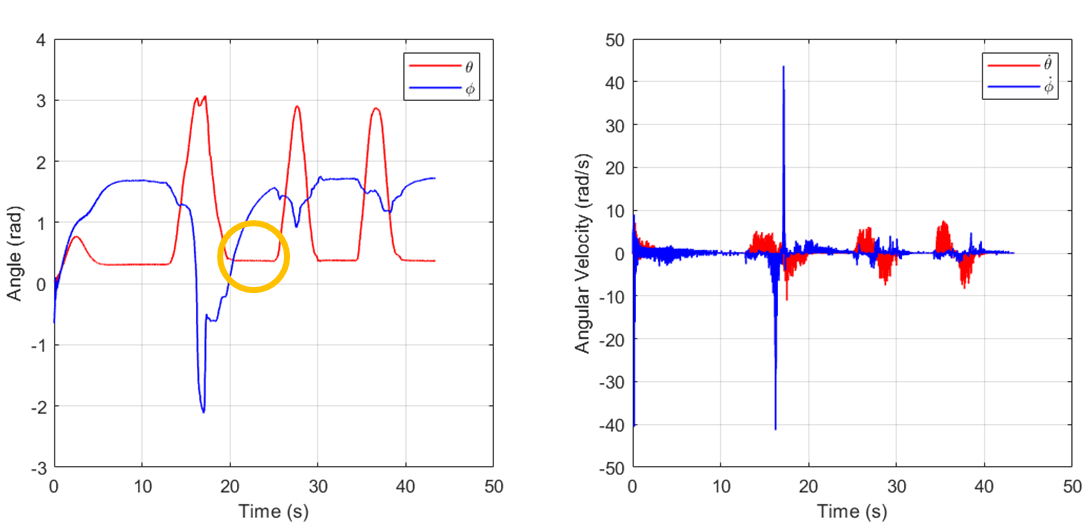


**Fig. W10-5** The angle and angular velocities from the IMU for random test

However, after all the electronics have been mounted onto the end effector, different from what showcased in the previous figure, the data outputs seemed pretty unstable at the very beginning, and it will take about 10 s to become stationary, it was deemed to be affected by the magnetic field formed by the twined wire; To discover whether this would influence the IMUs’ outputs during the real experiments, I made the arm move in a certain way, bend upwards first, then let it down to the equilibrium point for a while (to find out how long will the angle and angular velocities come to stable), it turns out that the change is continuous without any phase lag and it will not be affected by the magnetic field.



**Fig. W10-6** The angles and angular velocities when the arm is stationary



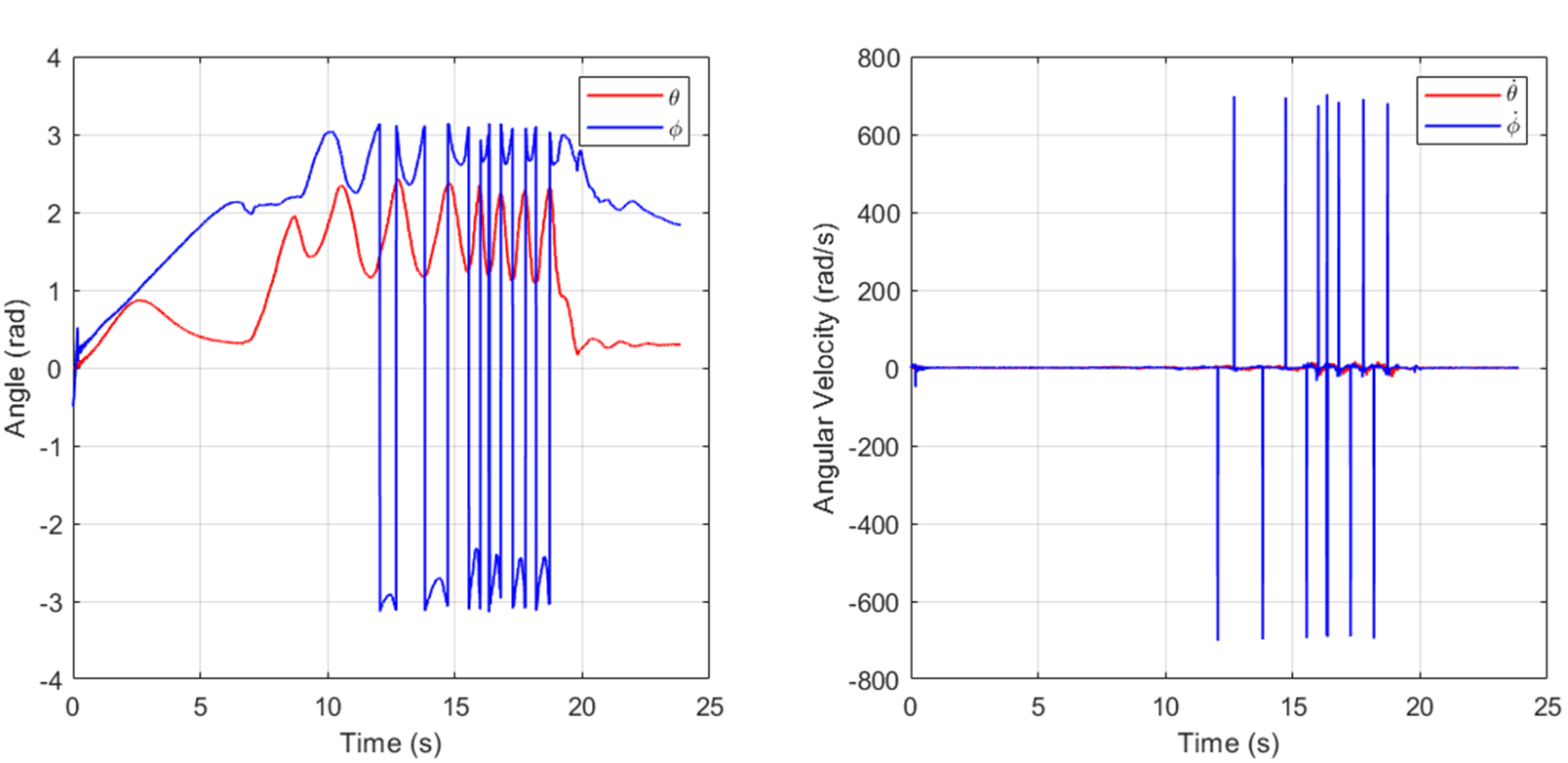
**Fig. W10-7** The angles and angular velocities when the arm is moving

**So the conclusion is that to ensure a more accurate data collection, when the power is on, it’s better to wait for another 10 s to let the system output come to stable.**

#### (3). Noise and drifting error

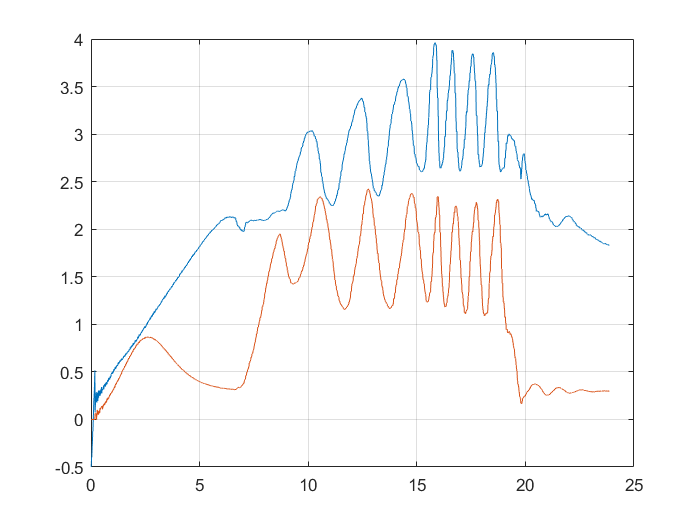
The noise caused by the original signal always exists, just like what have been showcased in the figures above, for continuous motion of the soft robot arm, there will be some noise especially for the velocities, as stated in last week’s report and several figures above, at some of the points, the angular velocities can reach 700 , it could not be the problem with velocities themselves, instead it could be the angle change or the raw data (quaternion). From the figure below it can be seen that the angle changes very violently when they achieve , and it seems that this phenomenon only happens for , and according to the definition provided by Manu, the range will be , apparently this is not the case. Since it could be so complex to fix the formula to convert quaternions into angles, we can make some modifications to the processed data to make the angle change continuous. The temporary solution is that to judge the negative and positive sign of angle value, the logic is shown as follows,

if and



**Fig, W10-8** The random test with data processing error

The reason for why we need to compare the value of the former step with 3 rather than is that at the mutational site, the value will be very close to , but not exceeding . And the estimation is not so accurate for some precise control, we need to identify and store the value for each peak. And the final effect is proved to be promising, which can be seen as follows.



**Fig. W10-9** The angle change after calibration

Next I need to transplant the test code into the real time data processing code to see if there is something else I need to fix.