In my master’s thesis, I developed a method to track the position and orientation of a permanent magnet using an array of magnetic sensors, and optimize the sensor configuration for better tracking performance. The magnet can be mounted on the tip of an instrument. By tracking the configuration of it, we can reconstruct the shape of the instrument to close the control loop. When we were discussing potential project idea, I found this particularly interesting because of my experience at Noah. I was thinking maybe closing the loop of the instrument by knowing its shape will further improve its control performance. By doing this project, I may get the skill set of shaping sensing that is beneficial for an industrial project.

(So, my project is interested in solving the inverse problem: determining the magnet’s configuration from the magnetic field recorded by the sensors. To do this, I derived the linearized magnetic dipole model, which is used in developing the efficient algorithm to estimate the magnet’s configuration. I also optimized the sensor arrangement to get better performance of the algorithm. I validated all of these with experiments, both in simulations and real-world testing.)

So, my project is interested in solving the inverse problem: determining the magnet’s configuration from the magnetic field recorded by the sensors. To do this, we first need to know how the magnetic field works, which is represented by the magnetic dipole model that computes the magnetic field based on the magnet’s position and orientation.

It’s a highly nonlinear model and difficult to work with. We are interested in linearizing it for two reasons: first, it allows us to use the knowledge of linear algebra to understand how the magnetic field behaves, and it is effectively used in the algorithm that solves the inverse problem.

To develop the linearized model, it requires parametrizing the orientation in the Euclidean space, which needs to be careful because the Euclidean space is not isomorphic to the space of orientation, which is the special orthogonal group SO(3). Naively using, for example, Euler angles, will have singularity issue and the linear model will not be physically correct. Instead, we used the exponential coordinates and a trick of local parametrization to solve the singularity issue and obtain the correct linear model.

With the linearized model, I developed a Gauss-Newton algorithm to calculate the magnet’s position and orientation from the measured magnetic field. This approach was 100 times faster than using other tricks to solve the singularity issue, for example, the quaternion, which requires additional constraint.

With the algorithm developed, I then try to optimize the sensor placement to improve the accuracy and speed of the algorithm. I proposed two criteria – one for accuracy and one for speed – optimizing them we obtained a circular configuration, where the sensors are put on a circle.

To compare the optimized sensor configuration and the non-optimized one, we did experiment, both in simulation and physically. From the results, we verified that the criterion for accuracy is true – the configuration with higher accuracy criterion has higher accuracy in estimating the magnet’s configuration. It gives us an indicator for choosing between different sensor configurations: if we want a configuration that has higher estimation accuracy, choose the one with the higher criterion for accuracy.

We also found that the criterion for convergence speed doesn’t work out as expected. It turns out that putting the sensors further away from the magnet actually makes the algorithm converge faster, no matter what the speed criterion. It was surprising and needs further investigation.

Besides the algorithmic performance, we also did experiments to verify the validity of the linear model that uses the trick of local parametrization for orientation. I found that the predications from the linear model matched up with real-world data as long as the magnet’s configuration didn’t deviate too much from the linearization point, which is expected for a linearly approximated model. This wasn’t the case if we use global parametrization, like Euler angles, because of the non-isomorphism.

So, to sum it up: my thesis properly derived the linearized magnetic dipole model, which is used in developing the efficient algorithm to estimate the magnet’s configuration. I also optimized the sensor arrangement to get better performance of the algorithm. I validated all of these with experiments, both in simulations and real-world testing.

First, I am thinking of the math, but as you both know, the math in research is much more involved than that in industry. I have never the seen the term special orthogonal group after the exact lecture that talks about rotation matrices. Not even in the exam!

So I would say the programming skills that I learned from programming in C++. Even though I am using MATLAB for my project, my code base has a lot of different functions, such as the nonlinear model, the linear model, the algorithm that solves the inverse problem, the optimization …, they all interact with each other. By compartmentalizing them like in C++, I have a very clean code base. For example, if I want to change one function, I only need to look for one part of the code to modify. Right now, I am also cleaning up the code to prepare for the future paper. Having a clean code base reduces a lot of pain.

Besides that, I would say the communication skills. After working in the diverse team like Noah, I found it much easier to communicate with my supervisors, for example when we are discussing the results, the objectives and the next steps. But I found that I still have to improve my skills of talking math, that’s a completely different language.