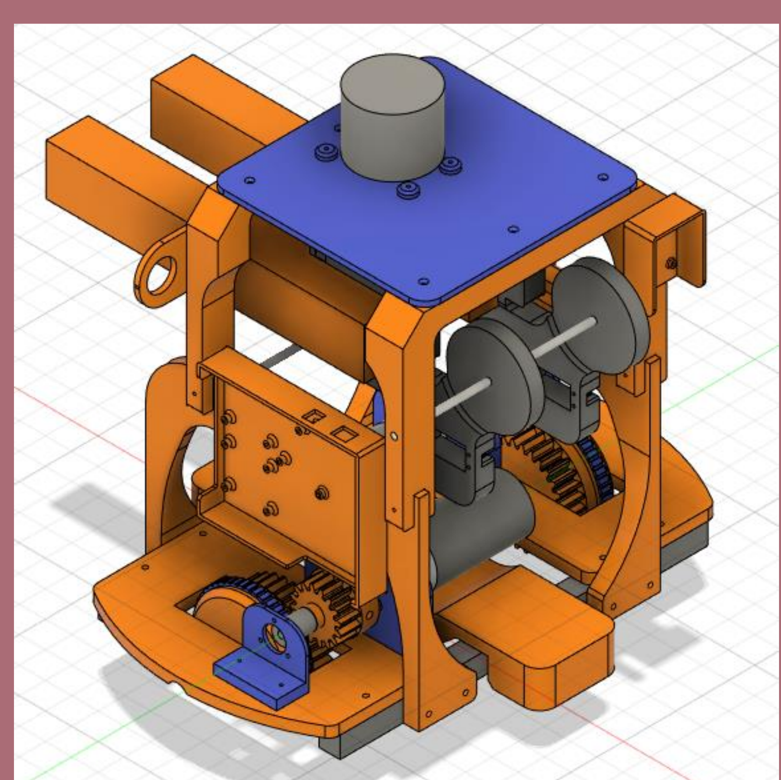
# Education Autonomous Robotic Platform

Bill Cai, Han Cao, Bruce Jiang, Hao Song, Manjun Li

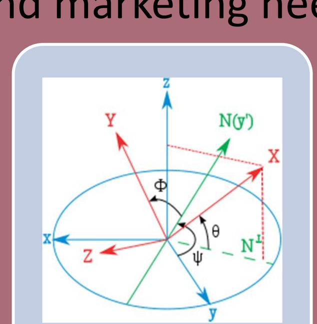

## Abstract

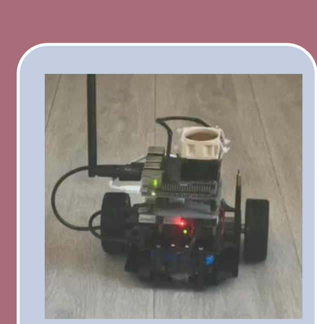
We are designing an open-sourced indoor autonomous robotic platform that can benefit education and industry. It consist of both the hardware and the software development of the robot. Our goal is to make getting into autonomous robot easier. By making the robot and solve problem along the way we provide a clear path to those who may be interested.

## Motivation

As we are engineering student, we clearly learned lots of knowledge during the college education. However, when try to put those knowledge in use, we usually found we have no experience at all. Thus, as Robotic is such an intricate subject that can easily involve machinal, martial, electrical and electronic engineering. It is the perfect testing ground to put our knowledge in used. But getting into robotic can be intimidating. That is why we want to make it easier. With our open-source platform, people can go straight into what they are interested and explorer the depth that can benefit both their research and marketing needs.

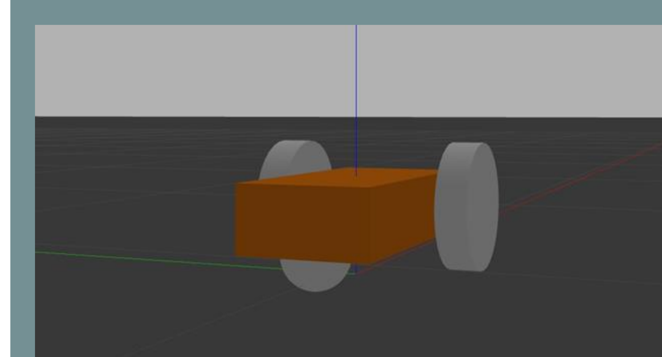


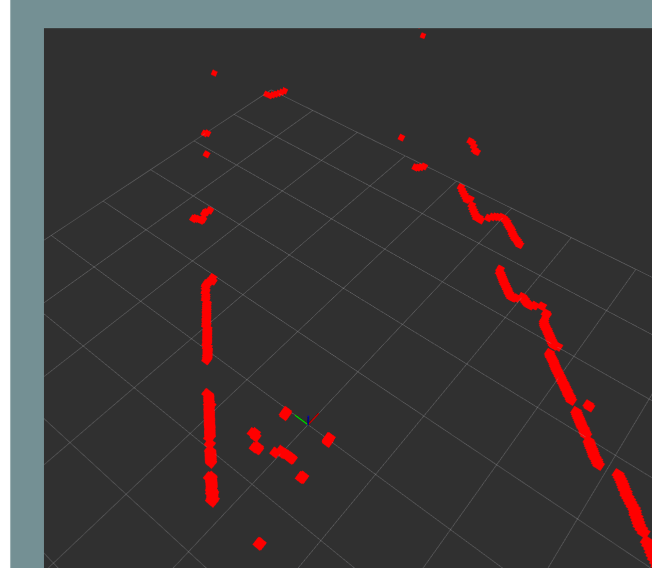


## Methodology

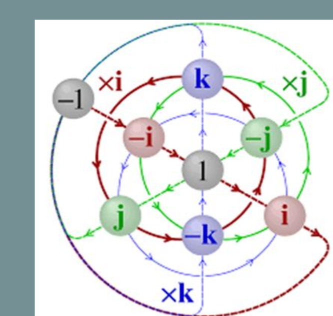
### Simulation

A simple simulation model was created in the gazebo shown below. This design featured a rectangular body with two round wheels attached to both sides and a universal ball joint on the back for passive movements.

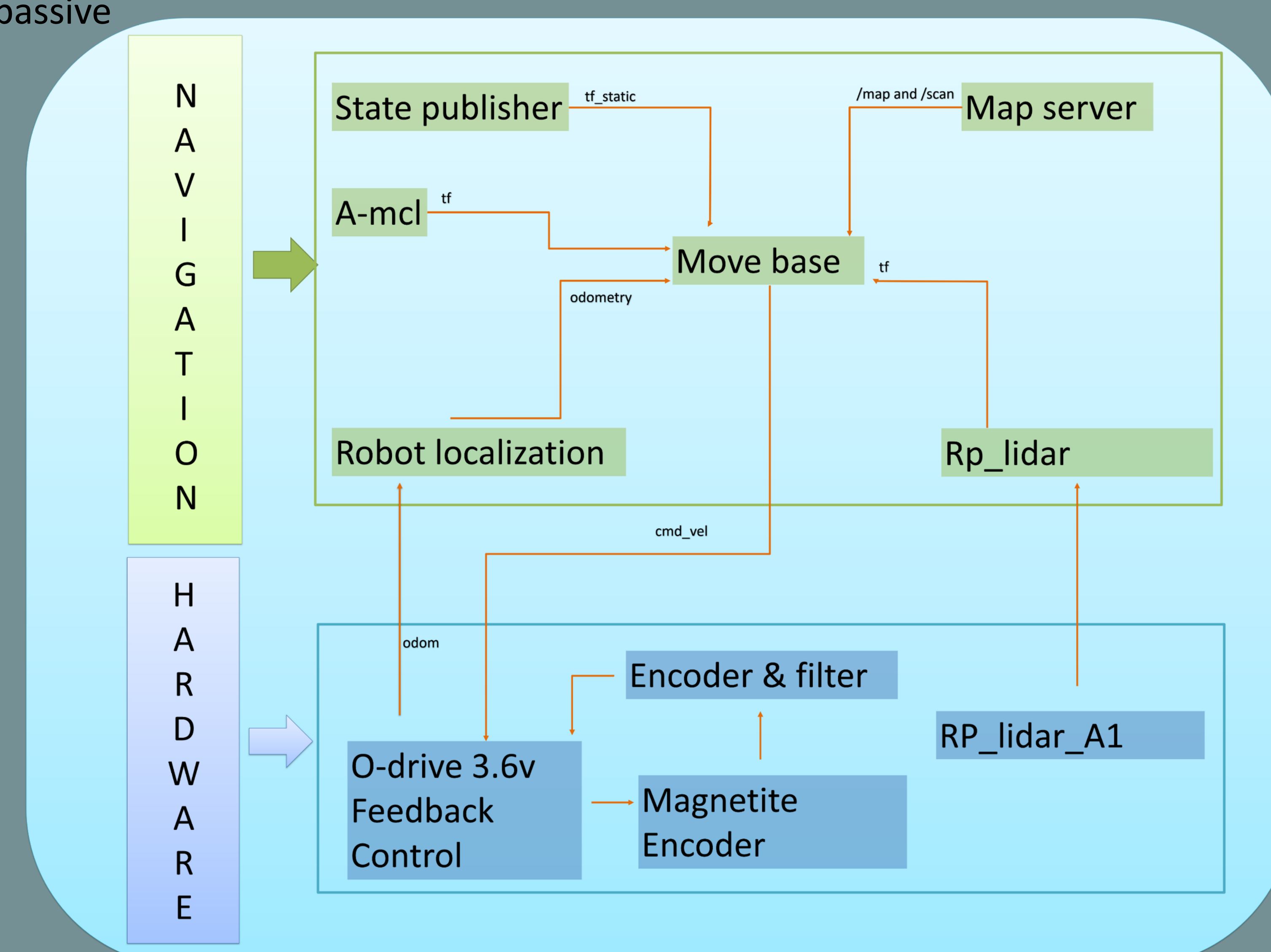


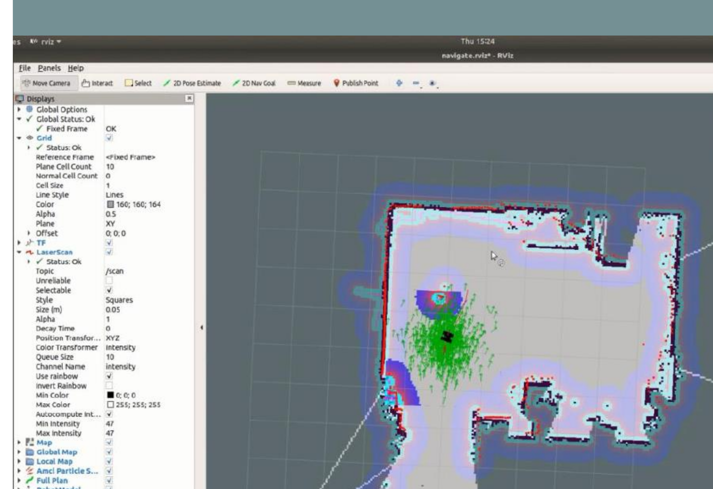


### Quaternion



the team decided to use quaternion curves to calculate the animation of the robot. With four real numbers and three imaginary units in one equation, this module was able to calculate the spherical movement of any angle from any coordinate.



### System Overview

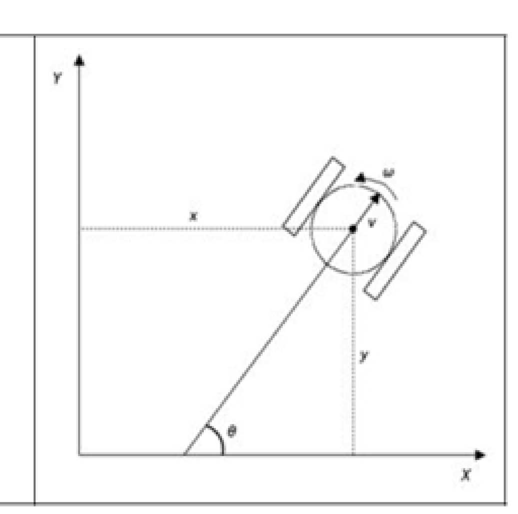
An Overall view of the control system design is show as followed, Have hardware level and navigation level. The hardware provide mobility to the robot as it also produce senser data to the navigation level. The navigation use ROS packages to dissect those data. After the decision is made by the navigation using those data, a move command is given to the hardware level.

$$k_{00} = v\cos(\theta_{n-1})$$
$$k_{01} = v\sin(\theta_{n-1})$$
$$k_{02} = \omega$$

$$k_{10} = v\cos(\theta_{n-1} + \tfrac{t}{2}k_{02})$$
$$k_{11} = v\sin(\theta_{n-1} + \tfrac{t}{2}k_{02})$$
$$k_{12} = \omega$$

$$k_{20} = v\cos(\theta_{n-1} + \tfrac{t}{2}k_{12})$$
$$k_{21} = v\sin(\theta_{n-1} + \tfrac{t}{2}k_{12})$$
$$k_{22} = \omega$$

$$k_{30} = v\cos(\theta_{n-1} + t\,k_{22})$$
$$k_{31} = v\sin(\theta_{n-1} + t\,k_{22})$$
$$k_{32} = \omega$$

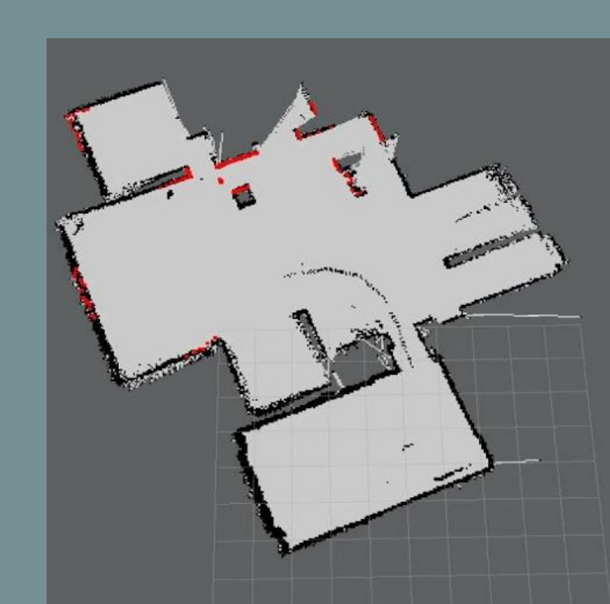


$$\begin{pmatrix} x_n \\ y_n \\ \theta_n \end{pmatrix} = \begin{pmatrix} x_{n-1} \\ y_{n-1} \\ \theta_{n-1} \end{pmatrix} + \frac{t}{6}\begin{pmatrix} k_{00} + 2(k_{10}+k_{20}) + k_{30} \\ k_{01} + 2(k_{11}+k_{21}) + k_{31} \\ k_{02} + 2(k_{12}+k_{22}) + k_{32} \end{pmatrix}$$

### SLAM

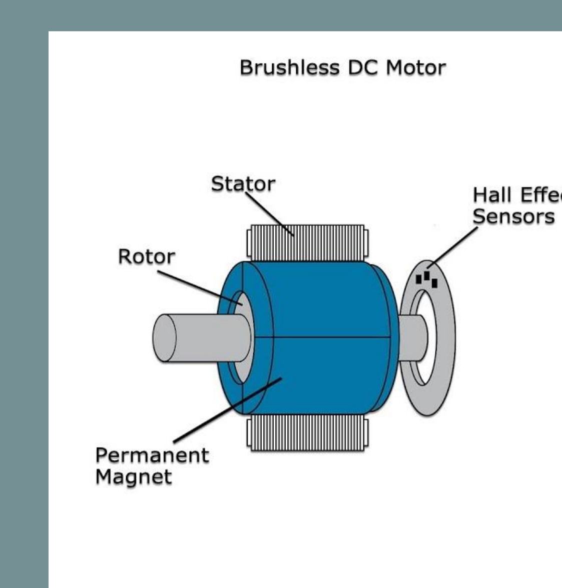
By comparison visual slam are expensive both in terms of the cost of the physical sensor and the processing power needed . they achieve a mediocre accuracy, but the upside is it is can be very compact and easy to integrate. Like to pre- existing solution of intel real sense and tacking sensor.

We can do stuff the old fashion way which is using encoder. We will convert the rotation of the wheel into odometry information. And as long as the wheel is not slipping, we should have a very accurate representation of the robot moving.



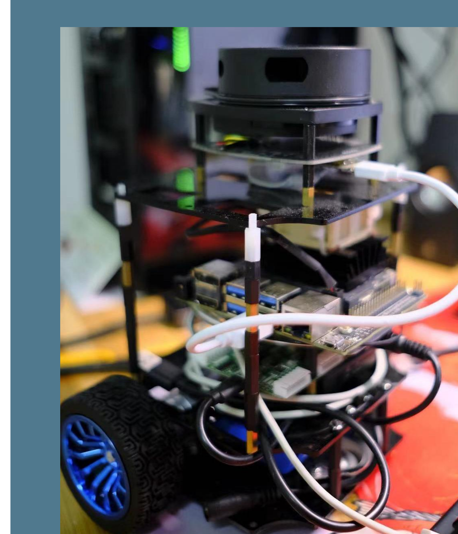
### Motors

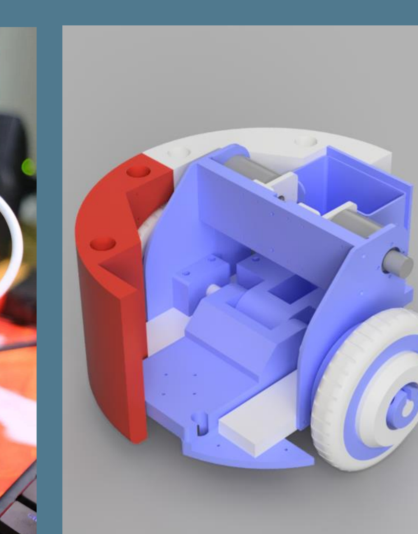
For brushed DC motors, it has a lower overall costs and it only needs simple controller, however, the brush may more out overtime, and it can't hold at certain angle. On the other hand, as its name suggests, the brushless motor doesn't have a brush inside the motor, thus require less maintenance. it also has relative higher efficiency. and most important part is its ability to rotate precisely to a required angle and to be able to hold it. This is critical in positioning and controlling of our robot since we might need to ask the robot to turn a few degree and that's what made us choose brushless motor for this project.




## Results

As a result, our team achieved a successful simulation run on gazebo, and calculated our robot's equations of movements. In the last part of discussion, our team argued the disadvantages and advantages from using this simulation model on our physical design.





## Conclusion

If physical model is not feasible at the end, a simulation environment will be built for testing.


## Acknowledgements

1. Fusion 360
2. Ubuntu
3. ROS
4. RViz
5. Gazebo
6. Github


Logos: Drexel, any funding agents, supporting companies