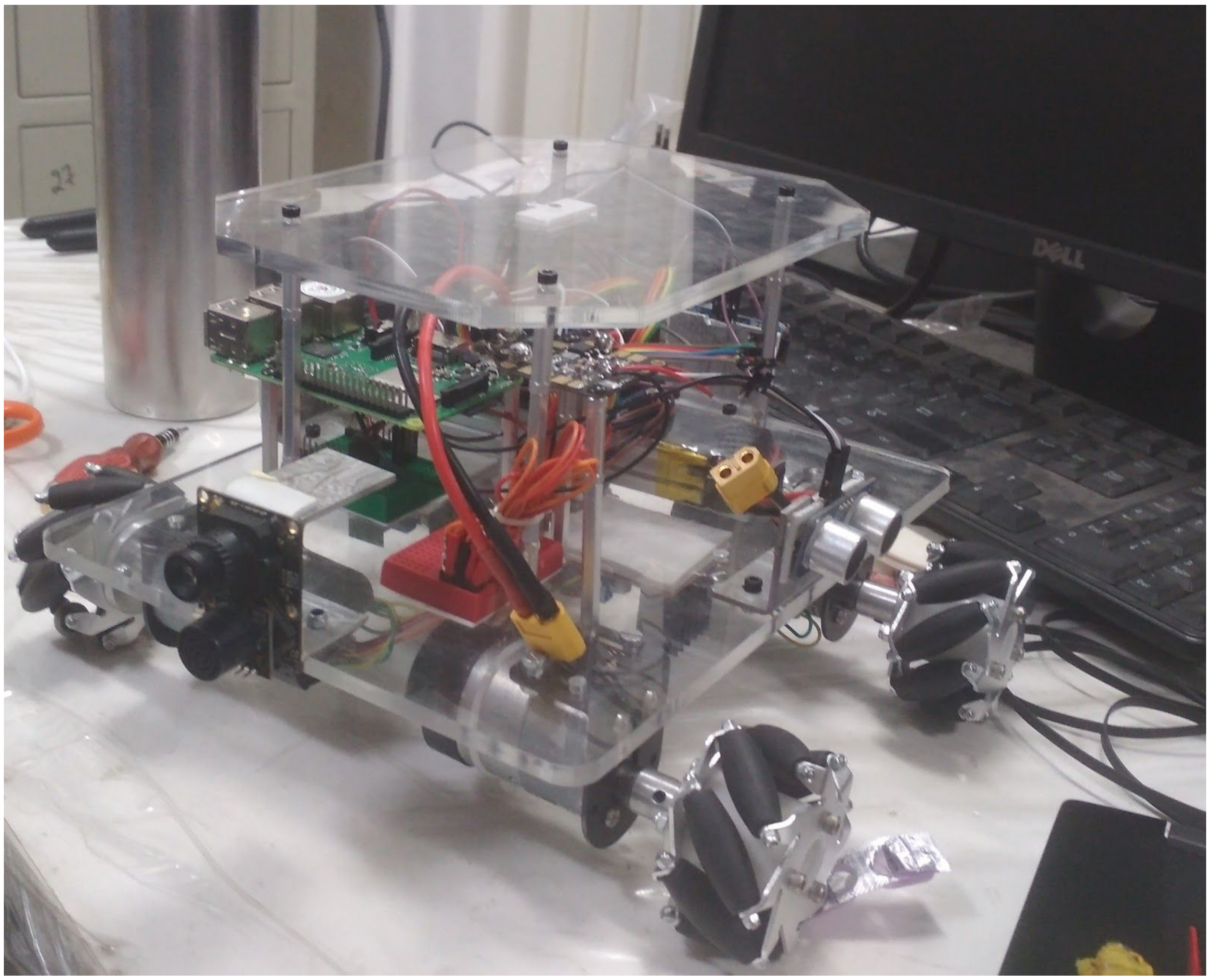
Summer Project

SWARM Robotics

Vivek Agrawal

Advisor: Dr. Soumya Ranjan Sahoo



Introduction -

Design -

- **Mechanical -**

    - Link for CAD design
- **Electrical -**

Processor and Controller Setup

We have used RPi as the master processing unit for our swarm-bots. Along with it, Arduino is used as a microprocessor for processing sensor data and send it to ROS nodes

Arduino -

- We have used Arduino mega 2560 in our setup to process data of sensors and send it to RPi via serial communication.

RPi Setup -

- Install Ubuntu Mate 18 on RPI from the following link
- Do some initial settings on RPI using following instructions
    - *sudo raspi-config*
    - navigate to '3 Interfacing Options'
    - since I plan to use ssh connection and the GPIO pins, I enable these two options for future projects
- Steps to install ROS Melodic on RPI -
    - Before starting to install ROS Melodic, we have to be sure that the system allows us to install different repositories:
        - Go to System -> Administration -> Software & Updates
        - Check the checkboxes to repositories to allow "restricted," "universe," and "multiverse."
    - Now to install ROS Melodic on RPI run the script given in the link
    - To Install and Setup Arduino on RPI run the script given in the link

Communication -

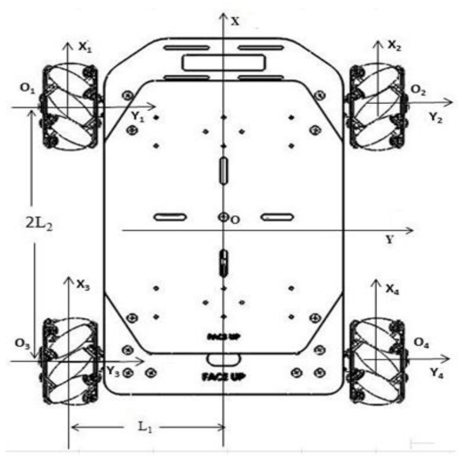
- The ESP8266 wifi module is low cost standalone wireless transceiver that can be used for end-point IoT developments.
- ESP8266 wifi module enables internet connectivity to embedded applications. It uses TCP/UDP communication protocol to connect with server/client.
- The WiFi AT Commands are useful in controlling the WiFi features of the ESP8266 Module like setting up the WiFi Mode of operation, get the list of WiFi networks, connect to a WiFi Network, setup the Access Point (AP), control DHCP, WPS, MAC Address, IP Address, etc.

Obstacle Avoidance -

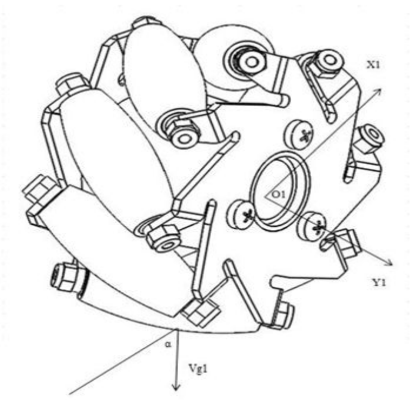
- Since robot needs to navigate in the environment with avoiding obstacles we use an ultrasonic sensor to detect the distance of robot from obstacles around it and if the robot encounters an obstacle in a defined radius it moves in the opposite direction to avoid it.

Omni Directional Motion -

- As our project is aimed at making swarm-bots which can travel in all direction with minimum change in orientation we decided to use mecanum wheels(ϕ 60mm) in our robots.
- The **Mecanum wheel** is a design for a wheel that can move a vehicle in any direction. It is a conventional wheel with a series of rollers attached to its circumference. These rollers typically each have an axis of rotation at 45° to the plane of the wheel and at 45° to a line through the center of the roller parallel to the axis of rotation of the wheel.



- Global Coordinate System



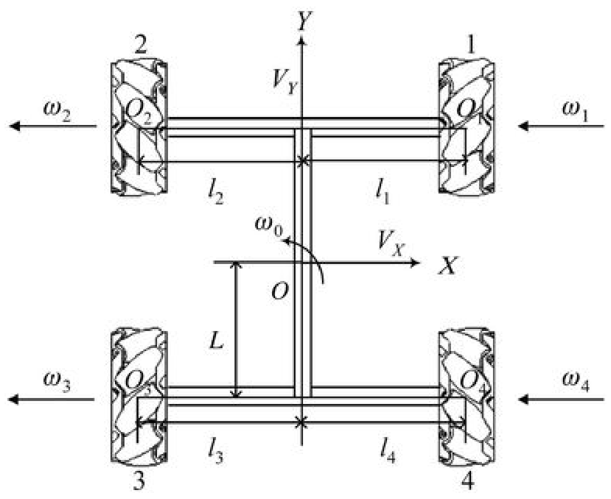
Local Coordinate System

- Mecanum wheels were oriented in bots as shown above in the figure.
- For driving a mecanum wheel robot following equations were used -

$$
\begin{aligned}
v_{\omega 1} &= v_{t_y} - v_{t_x} + \omega(a+b) \\
v_{\omega 2} &= v_{t_y} + v_{t_x} - \omega(a+b) \\
v_{\omega 3} &= v_{t_y} - v_{t_x} - \omega(a+b) \\
v_{\omega 4} &= v_{t_y} + v_{t_x} + \omega(a+b)
\end{aligned}
$$

v_{ω}: speed of Mecanum wheel
v_{t_x}: speed of vehicle in x-axis
v_{t_y}: speed of vehicle in y-axis
ω: angular velocity of vehicle
a, b: the length in x-axis and y-axis between the center point of vehicle and the center point of Mecanum wheel

Here $\omega 1$ is the angular velocity of the front right wheel, $\omega 2$ that of the front left, $\omega 3$ that of rear right and $\omega 4$ that of rear left.



Odometry -

- To estimate the position of the robot from starting position we use the odometry data of mecanum wheeled robot by finding the inverse of jacobian matrix used to calculate velocities of motor and using velocity measured from encoders as input to determine resulting velocities in x and y direction along with the angular velocity of robot.
- As odometry data is not very accurate we also used **imu** data to along with it to correct position estimate of the robots.

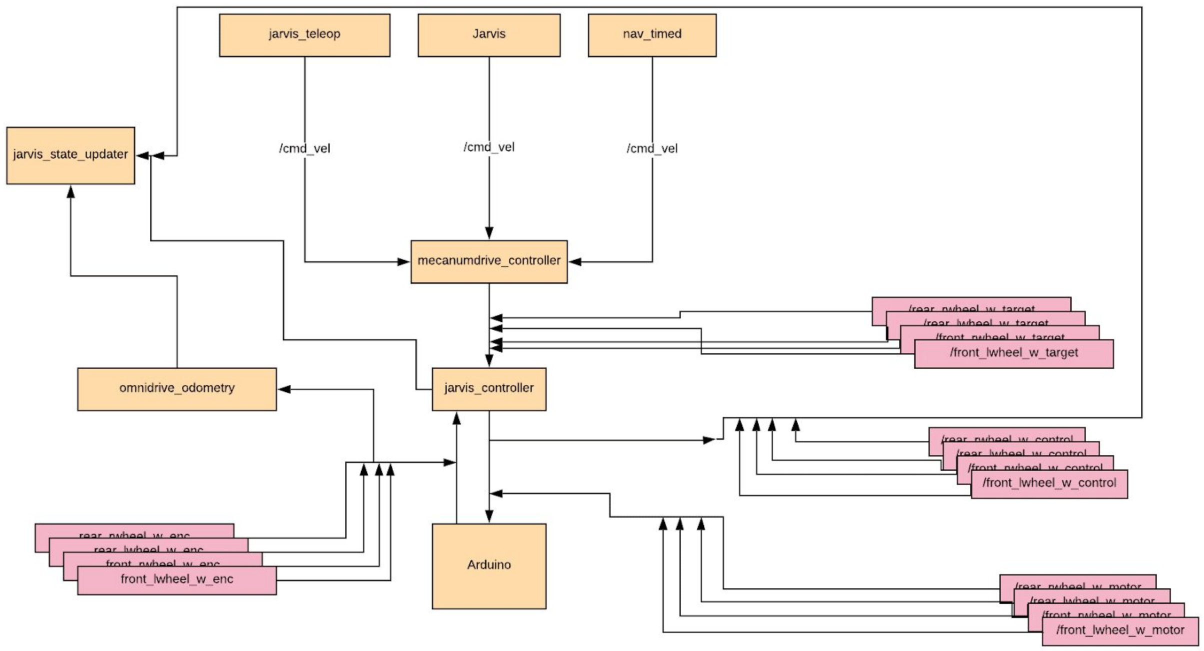
Controller -

- As there should be a feedback system in the robot to correct the commanded velocity of the robot we made a PID controller for each wheel so that it can correct the velocity of each robot.
- I also tried to plot the characteristics of the motor by drawing the curve between voltage and speed but encoder data does not attain a stable value so we find the parameters of PID controller manually.

ROS -

- Program for each of the operations is written in different packages so that code would be easy to read and can be modified easily. So we used the Robot Operating System(ROS) so that nodes of these packages can talk to each other by use of publishing and subscribing on topics.

- A flow chart of these packages along with publisher and subscriber on different packages is shown in the figure below.



Future work -

- Implement discrete slam on the robot by use of an ultrasonic and infrared sensor.
- Give an eye to the robot so that it can detect objects around it.
- Implement swarm Intelligence algorithm to make swarm for a specified purpose.