Structure of our Robot

Cortex-M4

Our microcontroller functions as the brain of the robot and provides 18 analog ports, and controls the 4 motors. This newly designed microcontroller board has extra ports in order to help with debugging and extensions of code.

MPU-6050

Instead of using the compass sensor for measuring which angle the robot is facing, which had proven to be inaccurate and unreliable in certain spots on the field, we decided to use MPU this year, which detects Earth's magnetic field to find the north and is more reliable in a game.

Ultrasonic

We implemented 4 ultrasonic sensors - left, right, front and back to detect the distance to the wall and calculate where the robot is on the field.

Motors

The motor controller board sends high current directly from the battery to our joinmax motors and supply all the power of the motor.

IR Sensor

Our IR sensor is attached to a 3D printed mold to detect 18 ports of ball position.

Grayscale Sensor

On our attack robot, we implemented 4 grayscale box light sensors that have been tested to be the most reliable light sensor type.

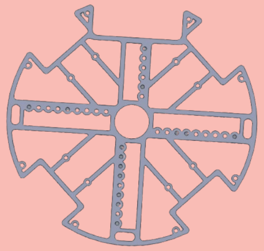
On our defence robot, we implemented 3 T-shaped grayscale sensor that returns the average value of 6 light sensors and covers a wide range of the field, providing the defence robot a better understanding of its position on the field.

Takaha Kicker / Capacitor

The Capacitor is connected to the battery and stores up potential energy. It then converts the 12 volts of input power into 24 volts of output power.

Chassis / 3D parts

All chassis boards and hardware designing are done via solid works and 3D printed with the aid of Bambu Labs



LOVBOT DYNAMICS

Henry Zhou

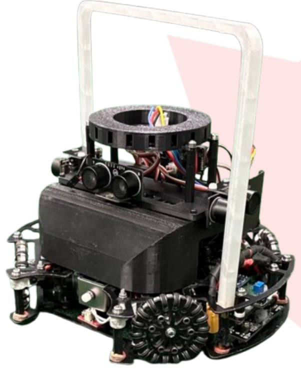
Hardware Manager
Defence Programmer

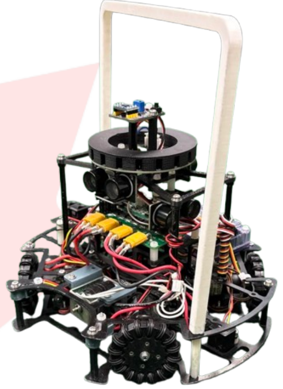
Ethan Xie

Documentation
Offence Programmer

Abstract

Based in Vancouver, Canada, Lovbot Dynamics has been dedicated to building competitive robots since 2023. Our team, composed of Henry Zhou and Ethan Xie devotes around 6 hours per week to design, test and making improvements. Our strategy features 1 defence and 1 offence robots, both built at a cost of approximately $1500 CAD. We decided to use the C programming language.





Past Rewards

2024 Western Canada LWL
3rd place

2025 Western Canada LWL
3rd place

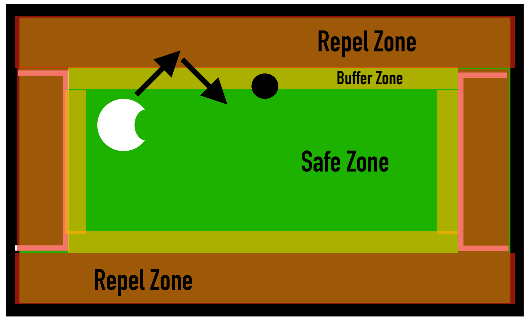
2025 Super-Regional Americas
2nd place

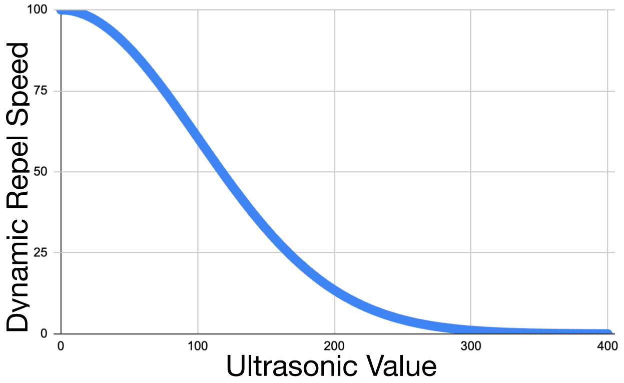
Data & Discussion - Dynamic Repel Feature

We divided the field into three zones: safe zone, buffer zone and repel zone. When the robot is in the safe zone, the robot can move freely. When the robot is in the buffer zone, the robot slows down. But when the robot steps into the repel zone, this feature activates and repels the robot in the opposite direction to avoid it going out its designated area.

```
double adjustment = 0.5 + (0.5 - repelForce);
repelX += adjustment;
```

We also have the feature so that the farther the robot passed the threshold line for repel zone, the faster it repels. The following graph shows the speed change of the robot based on the ultrasonic value. (Threshold value is 300)

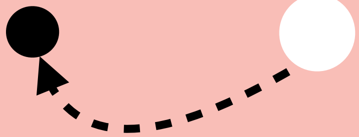




Software - Attack

aroundBall() {

This function manages how the robot moves when it sees the ball, but when the ball is not yet secured. When the ball is directly in front of the bot (ballAngle == 0), then the direction is simply set to 0 (dir = 0) and drives straight forward.
In all other cases, the bot calculates an offset based on how far the ball is, creating a "curve" effect instead of driving towards it directly.



}

aimGoal() {

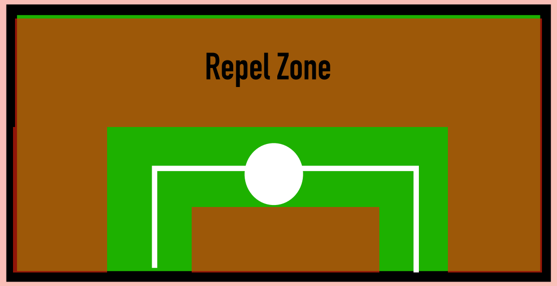
The aimGoal() function is called within the execution of aroundBall() and only executes when conditions are met. It takes the sum of the left and right ultrasonic sensors' values, and if the sum is not equal to zero, it normalizes the sum, returning a float between -1 and 1. -1 indicates that the bot is very close to the right wall, and needs to maneuver sharply to the left to score, whereas 1 indicates that the bot is extremely close to the left wall and needs to set its direction to the right to score. This value is multiplied by a const int MAX_OFFSET, converting the normalized float into a bot-friendly value.

}

Software - Defence

trackingTheLine() {

This year, we have came up with a new innovation regarding tracking the line by not depending on grayscale sensors. Instead, we use the ultrasonic sensors to create repel zones, which are defined by ultrasonic thresholds on the field. Whenever the robot steps into a repel zone, the robot repels back, preventing the robot from exiting its designated area as a goalie, while tracking the ball.



}