AARRAV ANIL

ROBOTICS & MECHATRONICS ENTHUSIAST

a.anil@student.tue.nl
linkedin.com/in/aarravanil
+31 623135624
aarravoltics
Eindhoven, Netherlands

ML BASED RESCUE ROBOT

Team leader | RoboCup France | Jan 2023 - Jul 2023

CONTEXT

Built an autonomous rescue robot to detect and help simulated disaster victims as part of the RoboCup Junior Rescue challenge.

GOALS

- Line follow a challenging path consisting of ramps, speed bumps and green patches
- Avoid obstacles autonomously and reach the evacuation zone
- Search and rescue victims
- Complete all tasks within 8 minutes

TECHNOLOGIES USED

- CAD
- 3D printing
- PCB design and assembly
- Integration of sensors and actuators
- Microcontroller programming
- Control Systems
- Machine Learning (TensorFlow Lite)

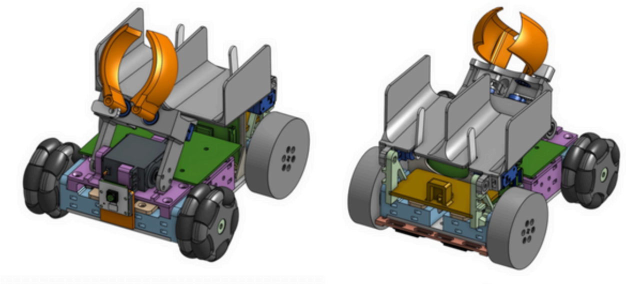
CONTRIBUTIONS

As the team lead, I contributed deeply to all major engineering subsystems from **mechatronic design and control systems to ML and system integration.**

Mechanical Design

- Designed the complete robot chassis and subassemblies using **Onshape**
- Kept in mind the principles of **Design for Assembly (DFA) and Design for Manufacturing (DFM)**
- Made 3D printed parts with **embedded nut slots, snap fit tolerances, and optimized internal clearances** to help in easy maintenance and assembly during the competition.
- Iterated through 6 prototypes, validating each for ramp clearance, center of gravity stability and overall performance.
- Each prototype was tested and iterated based on real world tests and performance.

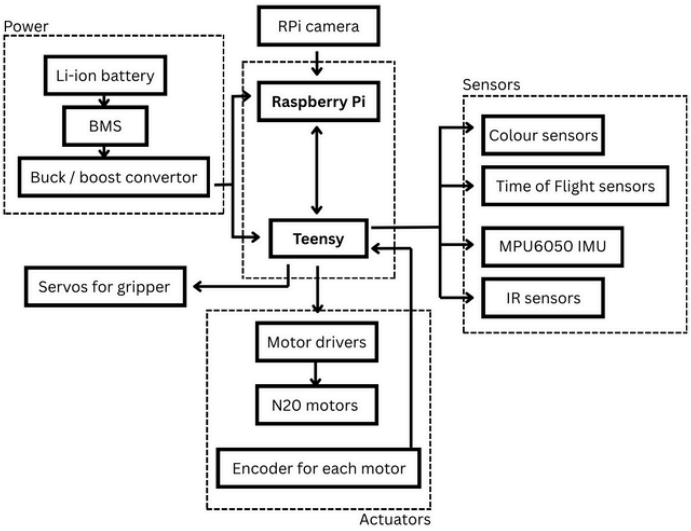
Link to the Youtube video!



Back View

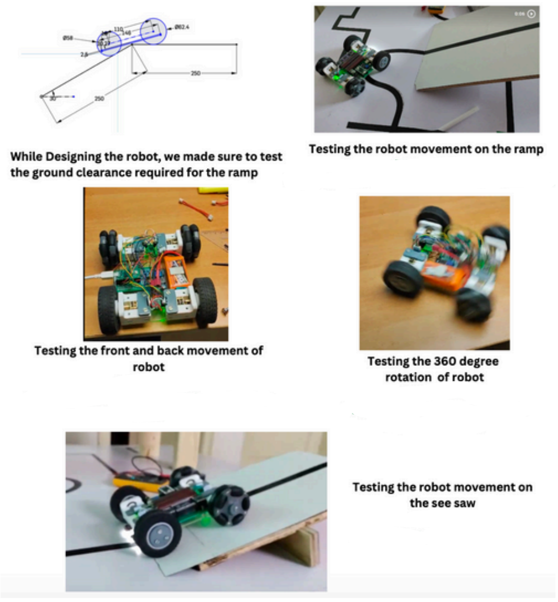
Front View

CAD Design of Complete Model



Power
Li-ion battery
BMS
Buck / boost convertor
RPi camera
Raspberry Pi
Teensy
Servos for gripper
Sensors
Colour sensors
Time of Flight sensors
MPU6050 IMU
IR sensors
Motor drivers
N20 motors
Encoder for each motor
Actuators

High-level system architecture



While Designing the robot, we made sure to test the ground clearance required for the ramp

Testing the robot movement on the ramp

Testing the front and back movement of robot

Testing the 360 degree rotation of robot

Testing the robot movement on the see saw

Testing

Component Selection & Electronics Integration

- **Read multiple datasheets** for sensors (like TCS34725 color sensor, VL53L0X ToF, MPU6050 IMU, IR sensors, N20/GA25 geared motors with encoders) to select parts that were precise, reliable and compatibility with the Teensy and Raspberry Pi.
- Created circuit schematics using **EasyEDA** to ease the assembly and soldering process
- Built and soldered all custom electronics, including **sensors, I2C multiplexers and buck converters**. Managed **power distribution** from Li-ion battery for both 5V logic and 9V motor drivers.
- Used communication protocols like **I2C, UART** and analog for inter device control and feedback.

Control Systems & Microcontroller Programming

- Programmed the **Teensy 4.0 (C++)** to handle line following, ramp detection, **PID motor control** and interface with the raspberry pi.
- Initially attempted **PID tuning using Ziegler-Nichols** method but due to static friction and motor lag, just used a trial and error tuning approach using **motor encoder** feedback.
- Developed **filtering logic** for sensor fusion between IR and color sensors to handle noisy edge cases on green and silver markers.

ML Based Victim Detection (Computer Vision & Optimization)

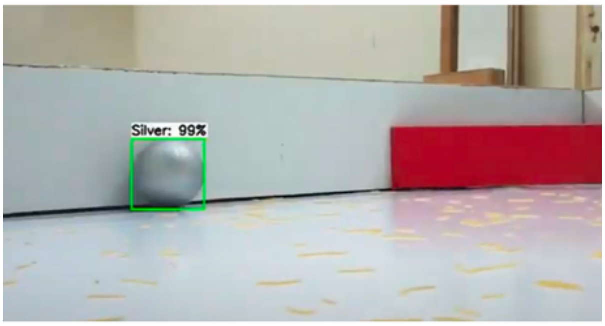
- **Data Collection**: Captured and manually labeled around 250 images of victims (silver balls, black balls, safety zones) under varied lighting.
- **Training & Optimization:** Trained a lightweight **Machine Learning model using TensorFlow on Google Colab**.
- **Deployment:** Optimized the model to run on a Raspberry Pi 4 with **Python**. Making sure to have good real-time performance and decent FPS under tight memory and power constraints.
- Tuned camera resolution, exposure and ROI to improve inference speed and maintain decent confidence levels.

System Level Integration & Debugging

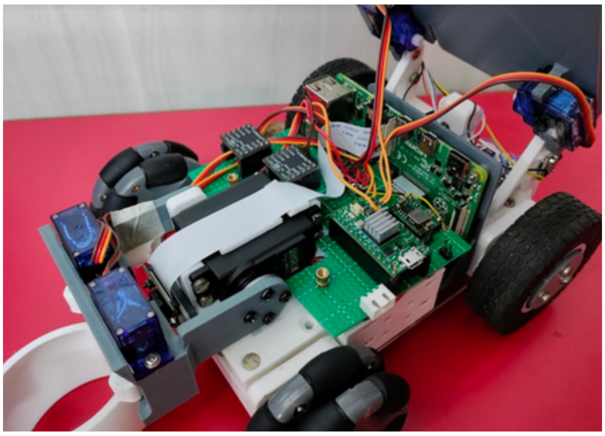
- Set up **serial communication (UART)** between the Teensy and Raspberry Pi to synchronize low level motor logic with high level ML decisions.
- Did **rigorous testing** of each subsystem (line follow, ML detection, gripper, ramp climbing) and **debugged integration issues** related to timing, communication errors and component malfunctions.
- Used an iterative integration process by testing and validating modules one by one before combining them, trying to reduce integration bugs.

RESULTS

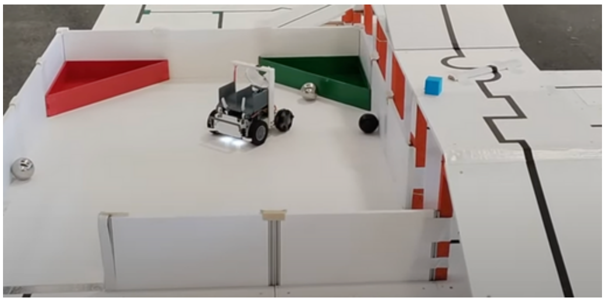
- Successfully completed all challenge tasks under 8 minutes.
- Achieved victim detection with **>85% accuracy** in evacuation zone.
- Recognized as best rescue method at RoboCup France 2023.



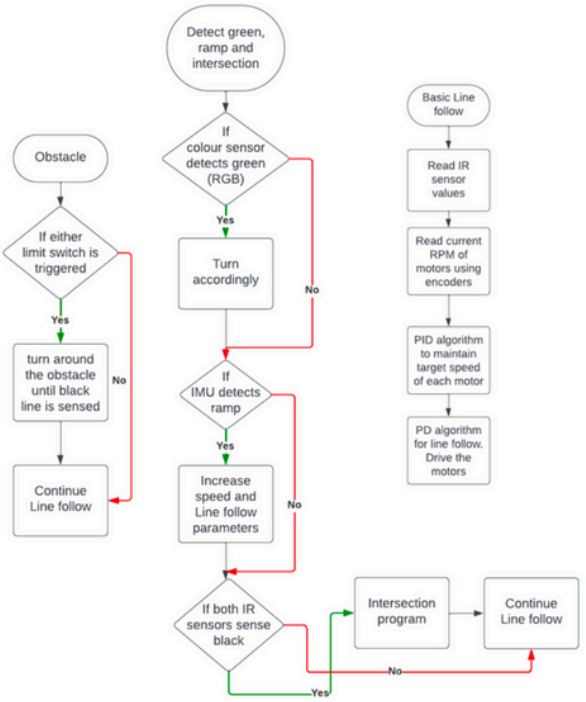
Computer vision and ML algorithm



Final robot model



Our robot on the competiton field


Obstacle
If either limit switch is triggered
Yes
turn around the obstacle until black line is sensed
No
Continue Line follow
Detect green, ramp and intersection
If colour sensor detects green (RGB)
Yes
Turn accordingly
No
If IMU detects ramp
Yes
Increase speed and Line follow parameters
No
If both IR sensors sense black
Yes
Intersection program
Continue Line follow
No
Basic Line follow
Read IR sensor values
Read current RPM of motors using encoders
PID algorithm to maintain target speed of each motor
PD algorithm for line follow. Drive the motors

Flowcharts showing the software logic