Hyungbin Jun

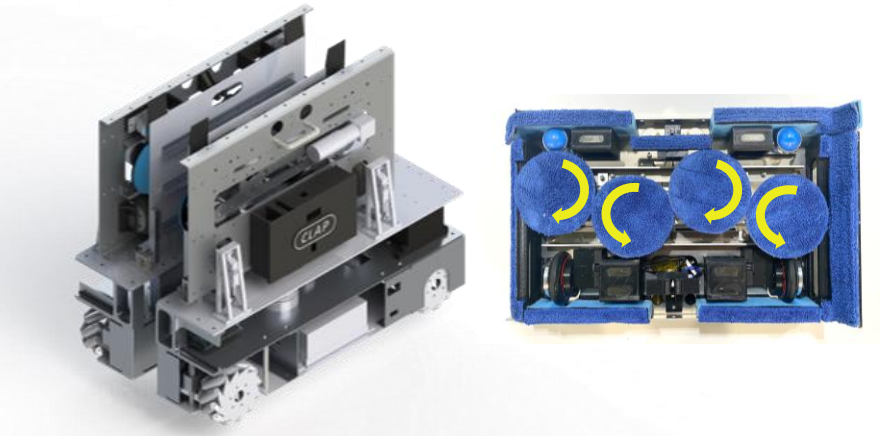
Engineering Portfolio

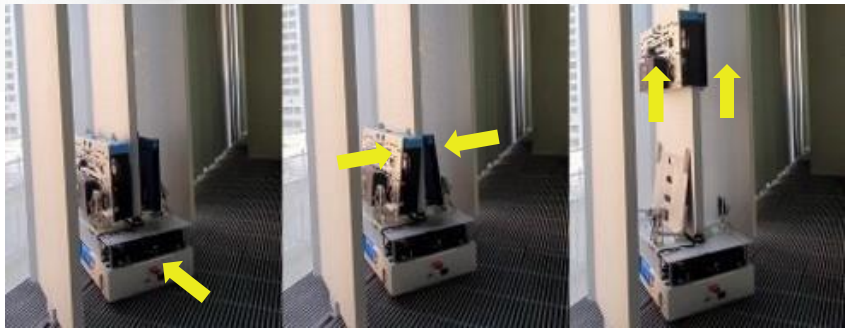
(+1)919-597-0067
hyungbin.jun@duke.edu
315 Towerview Dr, Durham, NC

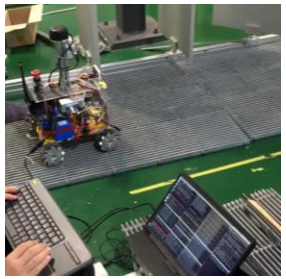
Table of Contents

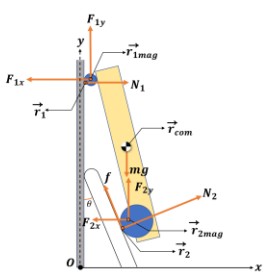
CLAP | Climbing and Cleaning Robot

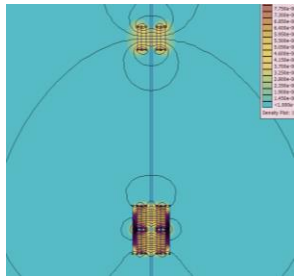
SolidWorks, AutoCAD, 3D Printing, Laser Cutting, Machining, Python











Objective

- To invent a robotic system to autonomously clean ~20,000 louvers (rotating thin walls of aluminum) in a new building for a company, in a team of 5 engineering interns

Technical Details

Mechanical Design

- Designed and created frames, mechanisms, electronic cases, and sensor mounts for a pair of magnetic wall climbing robots and an omnidirectional mobile base
- Developed a ferromagnetic ramp mechanism to connect climbing robots with the mobile base, increasing the power efficiency and reliability

Prototyping

- Utilized rapid prototyping techniques with SolidWorks, AutoCAD, 3D printing, and laser cutting to build 4 stages of prototypes for the system
- Designed experiments to evaluate the climbing and cleaning ability of the robot, built a testbed on a smaller scale for rigorous tests

Simulation & Analysis

- Analysed the stress with SolidWorks Simulation to restrict the deformation of the robot's body frame to 0.1 mm
- Simulated the magnetic field by conducting finite element analysis with Python, ensuring stability and increasing the magnetic force by 60%

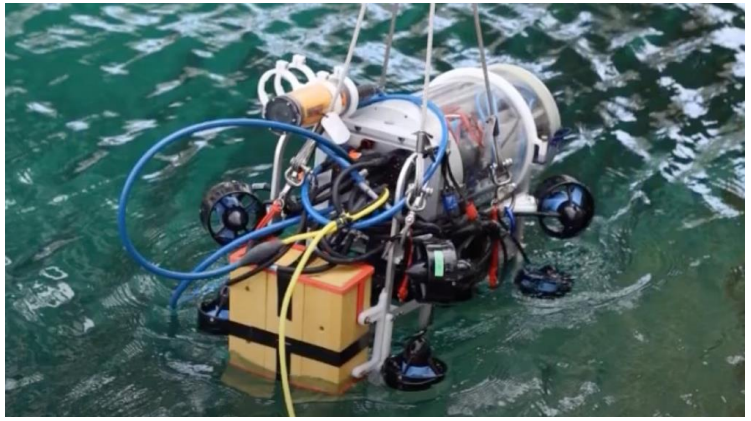
Results

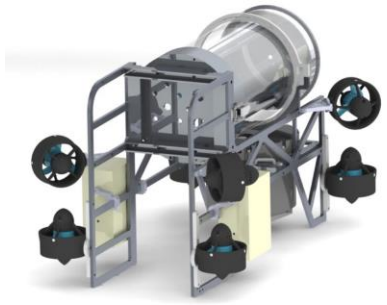
- A robotic system that navigates through narrow aisles and cleans each louver within 2 min, able to decrease the cleaning process for the building from 5 days to a single day, and reduce the portion of the uncleaned surface from 30% to 0%
- Patent pending for the invention of a modular robotic system with ground navigating and wall climbing capabilities

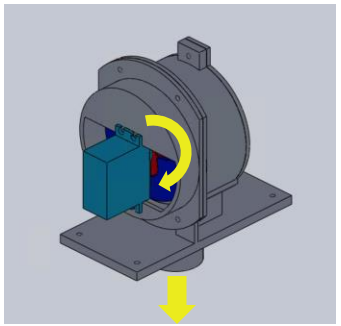
Link to video

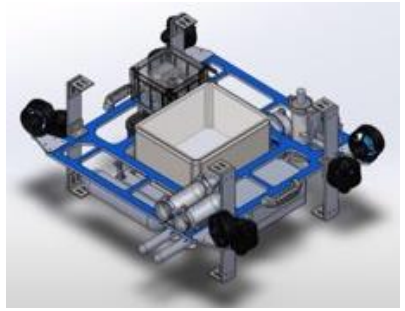
Cthulu | Autonomous Underwater Vehicle

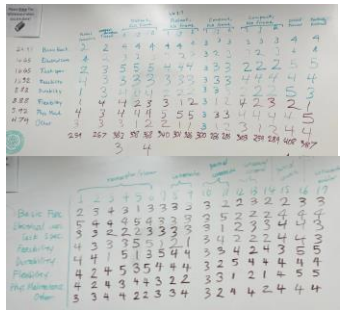
SolidWorks, 3D Printing, Waterjet Cutting, Machining, Python, OpenCV











Objective

- To develop autonomous underwater vehicles for RoboSub international competition, collaborating with 30+ members in Duke Robotics Club

Technical Details

Mechanical Design

- Improved the versatility of the new robot by doubling the space for additional actuators, increased maintainability by making the electronics case 20% more compact and its detaching time 50% shorter
- Ensured all electronics, actuators, and connectors are IP68 waterproof, and sensors have maximum angle of view
- Developed a marker dropper mechanism for placing metallic balls on an underwater target with <5cm offset

Computer Vision

- Developing a Python and ML-based computer vision software that identifies and estimates distance to an underwater gate under limited visibility and sunlight

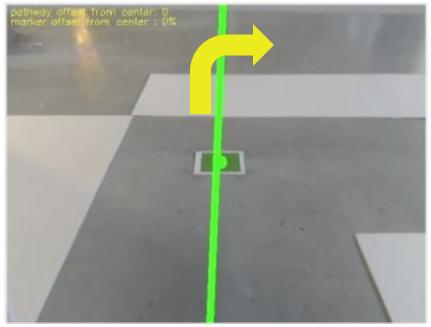
Results

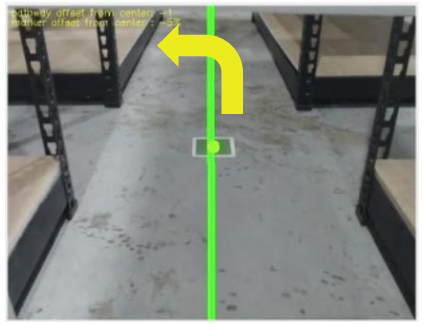
- Completed one version of the robot to be ready for pool tests and competition, finished the first stage of design for the new version
- Created design reports, skills videos, and a well-organized design documentation for all parts and assemblies for collaboration

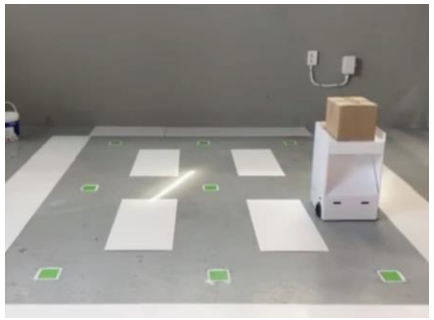
Link to video

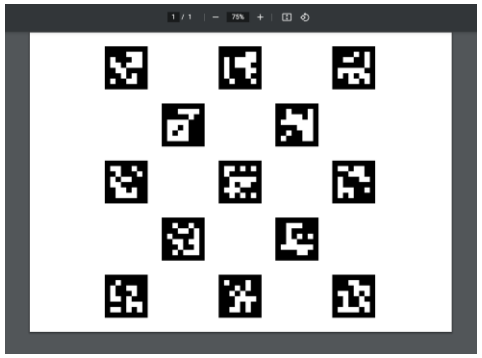
MiniCoCo | Autonomous Mobile Robotic System

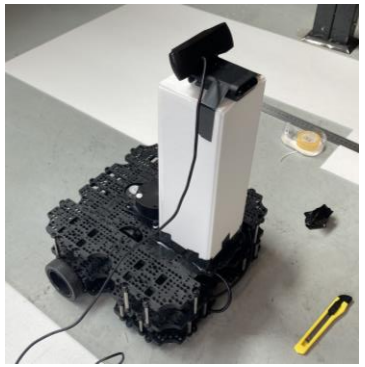
Robot Operating System, Linux, Python, OpenCV











Objective

- To develop and test a self-navigation software of mobile robots for logistics automation

Technical Details

Programming

- Implemented features including navigating to the desired location and streaming the robot's point of view to the user
- Minimized the risk of path overlaps between multiple robots by using an algorithm that treats every aisles as one way
- Reduced the printing preparation time for the software team by over 90% by creating a Python package to automatically generate PDF of Aruco Markers

Computer Vision

- Used OpenCV with Python to create a computer vision software that locates the aisles and colored planar markers, giving scroes from 0 to 100 to determine how well the robot is following the path
- Improved the lane detection algorithm for self-driving cars to ensure stability in environments with obstacles with various sizes and shapes

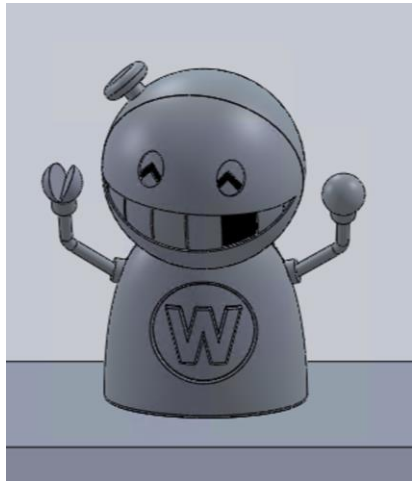
Results

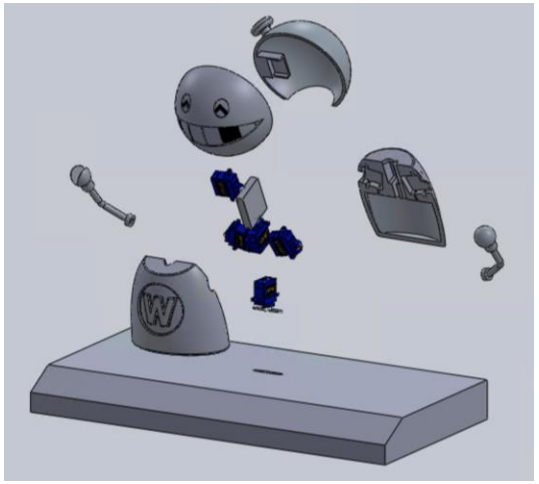
- Completed a ROS2 package for mobile robots to navigate within grid-like warehouses of any size, requiring no modifications of the warehouse
- Created a basis for software engineers to test navigation algorithms, higher-level computer vision software, and AWS server synchronization for fleet control, saving weeks of developing time

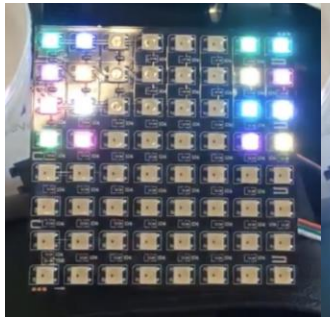
Link to video

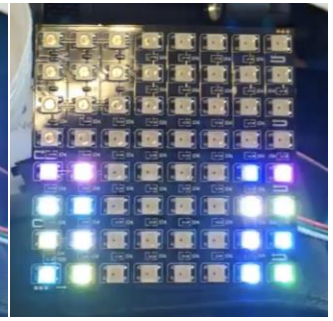
Concert-Bot | Music-Synced Robot Performer

SolidWorks, 3D Printing, Robot Operating System, C++, Python, Arduino *ongoing project









Objective

- To design an automated performer robot that generates dancing movements given user's music input

Technical Details

Mechanical Design

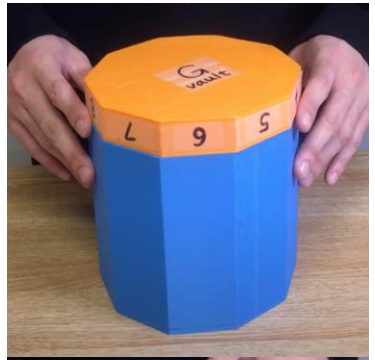
- Designed a 5-axis character figure robot hardware with SolidWorks surface modeling

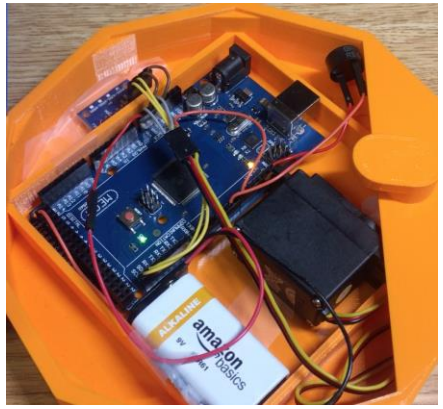
Programming

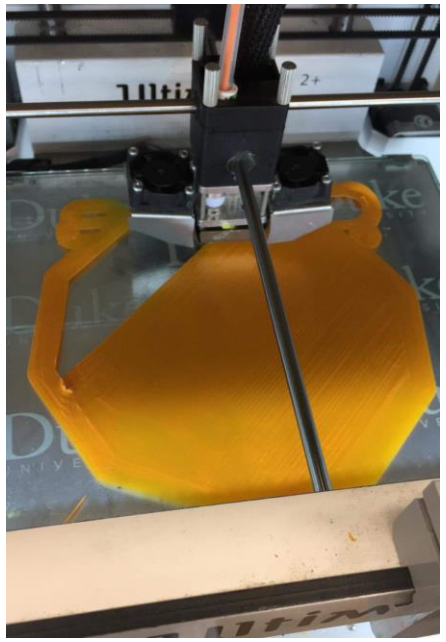
- Currently implementing C++, Python, and ROS2 based software for bpm detection, genre classification, and generation of natural dancing movements
- Currently implementing C++ based software for LED matrix control for displaying graphic output according to the genre

G-Vault | Arduino-based Locker Project

SolidWorks, 3D Printing, C++, Arduino







Objective

- To create a vault that utilizes its own orientation for password input

Technical Details

Mechanical Design

- Designed using SolidWorks and 3D printed a vault with a single-servo locking mechanism and an electronics case

Programming

- Implemented algorithms in C++ for locking/unlocking, alerting the user for wrong password, and resetting the password
- Utilized IMU sensor values for password input of 0~9, *, and #

Results

- A working prototype and a demonstration video of a novel vault design

Link to video