

ENG ME740 Final Project Report
The Design and Development of a Hovercraft-Inspired Robot:
“The Crow’s Nest”

Yousif Alhajji
May 1st 2025

Abstract

A small robotic hovercraft was designed and constructed to investigate the capabilities of monocular vision and attempt to use it as a novel approach in the development of a maze-solving micromouse. The project proved that the system is mechanically capable of traversing flat surfaces and that the communication stream was reliable. However, various issues arose with the autonomous aspect of the project due to oversights in system design and material selection. Ultimately, the system performed satisfactorily during manually controlled testing, and there is room for future improvement.

Introduction and Background

Robot motion and path planning are typically subject to non-holonomic constraints. In addition, the field of amphibious robots is rapidly developing because of their versatility and applicability to environmental research tasks. Many current amphibious robot designs make use of biomimicry to move around. Pliant Energy System’s Velox robot^[1] is an excellent example of this, it is a robotic system that uses soft fins that oscillate similar to a cuttlefish to move across land and swim underwater. The issue with many of these robots is that they tend to focus on underwater exploration rather than anything above the surface. They also tend to be fairly large and require complex control schemes to maneuver around, oftentimes resorting to remote control rather than being fully autonomous. The LORA hovercraft^[2] is an example of what can be accomplished with a surface-traversal autonomous robot, however, a noticeable characteristic of the design is that it is subject to non-holonomic constraints due to how its propellers are set up.

This project aims to design and build a robot that can traverse surfaces omnidirectionally. It will take inspiration from amphibious robots and due to the nature of the robot’s planned motion technique that “floats” across a surface, the primary sensor used was intended to be a monocular camera, therefore, the robot was projected to be able to transmit visual information wirelessly to a personal device for image processing and mapping. In addition, the robot was to be designed to be capable of being remotely controlled if desired. As a constraint, the robot’s dimensions could not exceed 30 cm x 30 cm x 30 cm so it can explore previously unreachable areas.

The idea for this project was originally thought up when doing research for a micromouse competition, and it was realized that most current approaches made use of two drive wheels and maximized friction^[3]. This resulted in a small sense of rebellion, which led to the approach of trying to minimize friction.

In the fall of 2024, a simple system was built in RASTIC to test if a robot could be capable of omnidirectional steering if it “floated” on ball transfers and utilized a lazy-susan mechanism that worked with a toroidal propulsion system to move around. While major flaws were realized with the product, the core concept of using wind propulsion similar to a hovercraft proved to be successful. This project aimed to use that approach by first addressing the mechanical flaws in the system, then adding various software-based functionality.

Product Design

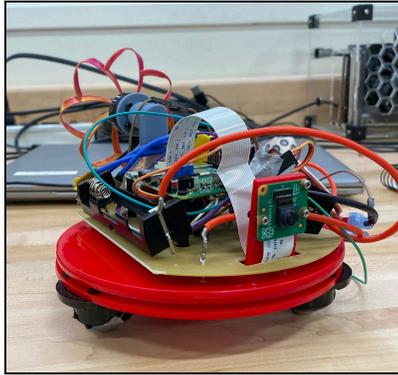


Figure 1: Old Design

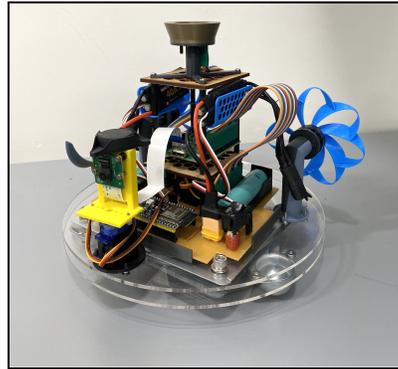


Figure 2: Current Design

The project was designed around two key components: sensing and propulsion. A Raspberry Pi Camera Module 2 was selected for its affordability and simplicity. However, while the camera alone could theoretically provide the necessary data, using an Inertial Measurement Unit (IMU) in tandem made state estimation more reliable. To ensure accuracy, the IMU was positioned near the robot's center of mass in a custom 3D-printed "Crow's Nest," allowing it to capture motion data effectively. A single Raspberry Pi was centrally mounted to maintain balance, with the Crow's Nest positioned above it. Unlike previous designs, where the camera remained fixed relative to the main body, this prototype attached it to a 9G servo motor, enabling dynamic reorientation. This modification aimed to compensate for the Pi camera module's limited field-of-view, enhancing its ability to track moving objects while keeping the system simple and cost-effective.

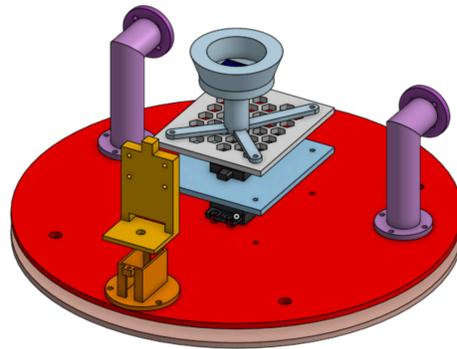


Figure 3: Mechanical Structure CAD of the Project

In terms of the mechanical system, the flaws in the previous design were addressed successfully. Most notably, a lazy-susan thrust bearing was implemented to minimize friction between the two spinning plates that make up the robot's main body. Additionally, two propellers were implemented this time around with the expectation that they would help control the system using torque vectoring in addition to the main body's swivel. This time around, because of the expected increase in mass of the robot, a 25G servo was used for the main body swivel. There were ideas explored that involved using a 9G servo connected to a gear transmission to boost its torque, but ultimately settled on the 25G to avoid unnecessary complexity. Most parts were laser-cut for rapid iterative testing, such as the rotary plates and the component platforms. However, more intricate parts were printed out of PLA because it was lightweight, stiff, and easy to fabricate using a 3D printer. Three Ball Transfers were used to ensure proper contact with the floor at all times^[4], which was intended to prevent "table wobble".

System Modelling

To model the system, a diagram was constructed as shown in Figure 4 below:

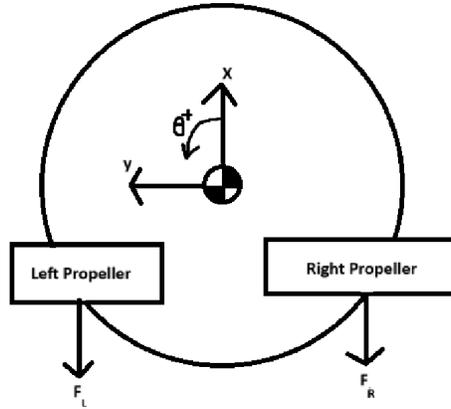


Figure 4: Diagram of the Differential Thrust from the Propellers

The state matrices and control inputs were found to be as follows:

$$\text{State Vector} = \begin{bmatrix} x \\ \dot{x} \\ y \\ \dot{y} \\ \theta \\ \dot{\theta} \end{bmatrix}$$

$$\text{Control Input Vector} = \begin{bmatrix} F_R \\ F_L \end{bmatrix}$$

Which led to the following state space model:

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{y} \\ \ddot{y} \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = A \begin{bmatrix} x \\ \dot{x} \\ y \\ \dot{y} \\ \theta \\ \dot{\theta} \end{bmatrix} + B \begin{bmatrix} F_R \\ F_L \end{bmatrix}$$

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 \\ \frac{1}{m} & \frac{1}{m} \\ 0 & 0 \\ 0 & 0 \\ \frac{d}{I} & -\frac{d}{I} \end{bmatrix}$$

Where “m” is the mass of the robot, d is the perpendicular distance between the thrust vectors and the centroid, and I is the rotational inertia of the system.

Using the aforementioned model, combined with suitable cost matrices Q and R, it became possible to iterate through the Riccati Equation^[5] and come up with a K-matrix that could be used to control the inputs to the system based on the desired states.

Circuit Board Design

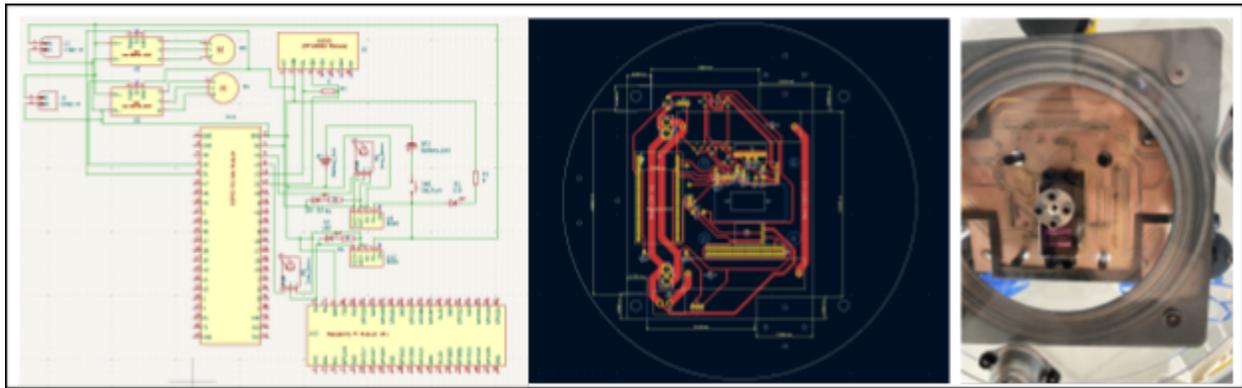


Figure 5: Development Process for PCB Manufacture
(Clearer Images Shown in Appendix)

The printed circuit board (PCB) for the robot acted as a power distribution network for the two 18650 Lithium-Ion cells, at 3.7V, which were connected in series to produce 7.4V. To account for the expected inrush current draw of the brushless DC motors, which could reach 9A, 5mm trace widths on 1 oz/ft copper were used for the path through the batteries^[6] with high current XT60 connectors acting as the interface. Additionally, time constraints and political decisions resulted in great difficulty when attempting to source the boards from international suppliers, which meant they had to be manufactured in-house. The PCB was CNC milled and all components were hand-soldered as shown in Figure 5. Furthermore, two Buck Converters were integrated into the circuit to step down the battery voltage to a consistent 5V for the Raspberry Pi, ESP32-S3, Servo Motors, and IMU. Three indicator LED lights were used to provide a visual cue as to the state of each power supply; if all three were on, it meant everything was working fine, if any were off, then it would be an indication that a fault occurred with the Buck Converters or the Battery.

The secondary function of the PCB was to allow reliable and high-fidelity signal transmission between electronic components. The signal traces were milled at 1mm widths to minimize resistance and spaced out in such a way that electromagnetic interference was negligible. The signal transmission portion of the design centered around the ESP32-S3 Development Module, which controlled all the actuators of the system. The IMU also made use of this signal network as it communicated with the ESP32 using an I²C protocol that required a 120Ω resistor between its SCL and SDA lines.

The Board was cut to a non-rectangular shape and mounted on a 3D printed standoff to prevent accidental shorting across traces and allow it to fit on top of the existing top plate design of the robot. Due to the positioning of the Raspberry Pi off the plane of the PCB, a row of header pins was used in tandem with a ribbon assortment of Dupont cables to make the connection between the single-board computer and the copper plane.

Remote Controlled Functionality

The first piece of functionality implemented on the system was the remote-controlled aspect. Unfortunately, the ESP32-S3 module turned out to be incompatible with all the handheld video game controllers immediately available. It should be noted that common ESP32 modules can work with controllers such as Sony's DualSense^[7], however, the ESP32-S3 was most readily available at the time of assembly. To address this, a custom controller was built using protoboards as shown in Figure 6 and was named the "DualZap". This controller had an ESP32-S3 module mounted to be able to communicate using the robot's microcontroller wirelessly using a low-latency ESP-NOW protocol^[8].

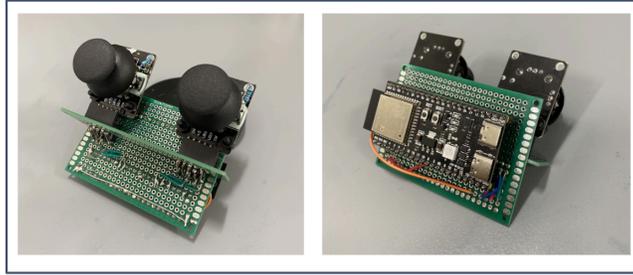


Figure 6: DualZap Controller

Once the remote controller worked and the inputs were mapped to their correct outputs, the robot was tested for its mechanical capabilities. An iterative P-test^[9] was performed to tune the mechanical portions such that they performed as intended. This process involved going through many physical prototypes and the replacement of materials until it worked.

Camera Streaming

Once the robot was able to move sufficiently well, it was time to implement the camera vision. This started with mounting the camera module onto its swivel motor and connecting it to the Raspberry Pi as shown in Figure 7. Following this, a Python program was written such that the Pi would stream the camera's view to a locally hosted webpage that could be accessed by any devices connected to the same network. Since the OpenCV library was used for this, it meant that the colorspace of the camera's output was in Blue-Green-Red rather than Red-Green-Blue, which meant additional processing was required to flip the bits.



Figure 7: Camera Mount

Point Cloud Generation

After the camera was up and running, an attempt was made to generate a 3D point cloud of a space using the robot. This made use of MATLAB's computer vision toolbox and various structure-from-motion techniques^[10]. Unfortunately, these attempts were unsuccessful as the camera had a low field-of-view and couldn't be positioned with precision, given the materials used on the current system.

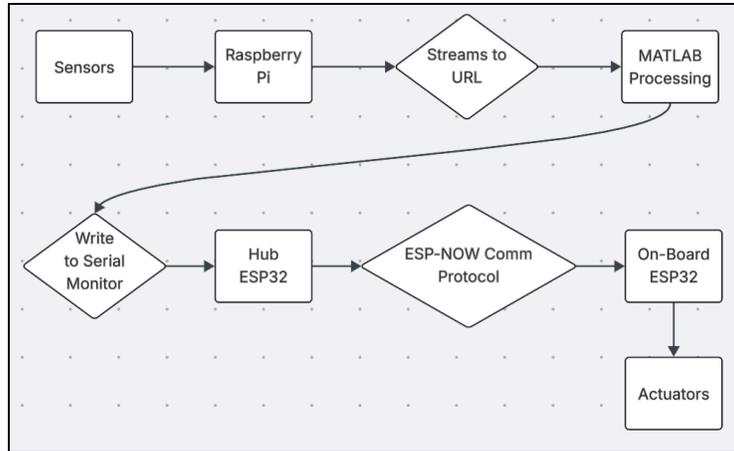


Figure 8: Communication Pathway from Sensors to Actuators

Object Following Functionality

Once attempts at point cloud generation were foiled, a MATLAB script was developed that would be able to take snapshots of the webpage and process the images within. The script was designed to isolate various sections of the images, convert them to binary matrices if the pixels were within certain limits of a Hue-Saturation-Value (HSV) colorspace. HSV was used for its reliability and effectiveness in color separation within images.

The color segmentation aspect worked, and the camera was able to follow a carrot being waved around in front of it reliably once a PD controller was implemented for the motions. However, this success only applied when the robot was static. During dynamic testing, the camera's low field-of-view, combined with the fact that it was rigidly mounted to the frame without motion dampeners, meant that the camera shook around with the main body of the robot and consistently lost view of the carrot it was meant to follow.



Figure 9: Carrot Isolation Program Highlighting a Carrot's Outline

Concluding Remarks

The system shown below was built and controlled successfully using a custom-built handheld controller. It was able to run off two Lithium batteries for hours, and the propellers were able to provide the necessary thrust and control to maneuver the system in all directions.

Unfortunately, attempts at automating aspects of the project were unsuccessful. Much more work and accurate modeling are needed for a system as complex as this. There is potential that this project will continue as a personal project and a challenge to develop skills in computer vision and system control.

In future iterations, the plan is to implement a Kalman filter using the IMU to minimize issues associated with strapdown navigation as discussed in Woodman's paper on the subject. Additionally, optical flow techniques will be explored to investigate whether they can be accurately used alongside suitable large language models to estimate depth from monocular vision.

Overall, though this project did not meet all of its objectives, it was key in developing a better understanding in multiple areas of designing robotic systems, and it succeeded in fostering a newfound appreciation and interest in intelligent mechatronics and vision-based navigation.

References

- [1]: Kai Ren, Jiancheng Yu, Research status of bionic amphibious robots: A review, *Ocean Engineering*, Volume 227, 2021, 108862, ISSN 0029-8018, <https://doi.org/10.1016/j.oceaneng.2021.108862>.
- [2]: Roubieu, F. L., Serres, J. R., Colonnier, F., Franceschini, N., Viollet, S., & Ruffier, F. (2014). A biomimetic vision-based hovercraft accounts for bees' complex behaviour in various corridors. *Bioinspiration & biomimetics*, 9(3), 036003. DOI: 10.1088/1748-3182/9/3/036003
- [3]: S. G. Kibler, A. E. Hauer, D. S. Giessel, C. S. Malveaux, and D. Raskovic, "IEEE Micromouse for mechatronics research and education," 2011 IEEE International Conference on Mechatronics, Istanbul, Turkey, 2011, pp. 887-892, doi: 10.1109/ICMECH.2011.5971240. keywords: {Mobile robots;Educational robots;Education;Wheels;Micromouse;Mechatronics;Research;Education}
- [4]: Shi, G. Contact theory. *Sci. China Technol. Sci.* 58, 1450–1496 (2015). <https://doi.org/10.1007/s11431-015-5814-3>
- [5]: “Kybernetika Vladimír Kučera a Review of the Matrix Riccati Equation.” n.d. Accessed May 3, 2025. https://dml.cz/bitstream/handle/10338.dmlcz/124651/Kybernetika_09-1973-1_4.pdf.
- [6]: Tran, T.T. (2023). Printed Circuit Board (PCB) Layout. In: High-Speed System and Analog Input/Output Design. Springer, Cham. https://doi.org/10.1007/978-3-031-04954-5_14
- [7]: Murphy, Sheila C, and Adam Lefloïc-Lebel. 2023. “Controllers.” *Routledge EBooks*, April, 17–24. <https://doi.org/10.4324/9781003214977-4>.
- [8]: Pasic, Roberto, Ivo Kuzmanov, and Kokan Atanasovski. n.d. “ESP-NOW Communication Protocol with ESP32.” <https://doi.org/10.37886/ip.2021.019>.
- [9]: Ullman, David. n.d. “The Mechanical Design Process.” <https://miemagazine.com/sample/Technology/TC1-100/TC050/sample%EF%BC%8DThe%20Mechanical%20Design%20Process%205th%20E%20David%20Ullman.pdf>.
- [10]: Ullman, S. 1979. “The Interpretation of Structure from Motion.” *Proceedings of the Royal Society of London. Series B. Biological Sciences* 203 (1153): 405–26. <https://doi.org/10.1098/rspb.1979.0006>.

Mounted Copper PCB:

