**ELEC330 Assignment 3 - Executive Summary**

**Team Butterfly - Biomimetic Robot Autonomous Navigation & Object Detection**

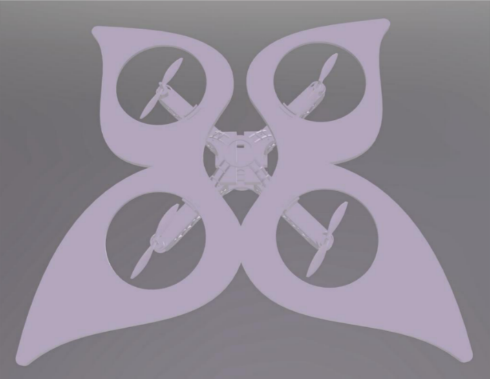
**Abstract**

This report presents the simulation of Butterflybot, a butterfly-inspired quadcopter developed in Gazebo Harmonic using ROS 2 Jazzy. The new model replaces the original flapping-wing design and incorporates a 2D LiDAR, IMU, and bounding-box camera for perception. This robot features SLAM (via Cartographer) and colour-based object detection using OpenCV. Thruster-based control facilitates allow for autonomous flight, while HSV segmentation is implemented to identify targets. Core systems—sensor integration, detection, and initial control—were implemented, establishing a modular foundation for future development and potential real-world deployment, even though 100% autonomy was not attained by the deadline.

**1 Introduction**

This project builds a high-fidelity virtual testbed for “butterflybot,” a bio-inspired aerial robot, by integrating ROS 2 with Gazebo Harmonic. To avoid the complexity of true flapping wings, a quadrotor core is enclosed in lightweight butterfly wings that preserve lift, stealth and low noise. The simulation stack combines rigid-body physics, a quasi-steady wing aerodynamics plugin, noise-injected cameras, LiDAR and IMU sensors, all orchestrated by ROS 2 Jazzy. Cartographer [1] enables 2D SLAM, OpenCV detects coloured fiducials for virtual “foraging,” and RViz visualises sensor data and planned paths. Deliverables include open-source URDF/SDF models, ROS 2 launch files, a Docker CI pipeline and a hardware-in-the-loop bridge.

**2 Design and Methodology**



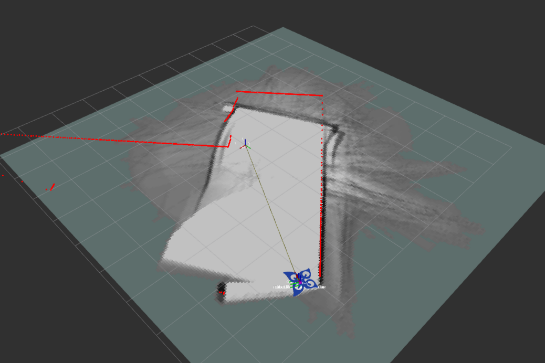
*Fig.2.1: New design with a motor-propeller*

Figure 2.1 shows the new robot design. The updated SDF model incorporates a restructured sensor setup tailored for the quadcopter design. A new IMU sensor was added alongside revised LiDAR and camera configurations. Sensor positions were redefined relative to the new base link, and all relevant ROS 2 bridges and plugin paths were updated to ensure compatibility with the new file structure. Furthermore, OpenCV, Google’s Cartographer and ArduPilot were utilised for object detection, 2D LiDAR mapping, and flight simulation, respectively.

**3 Testing Scenarios and Results**

The project implemented a colour detection algorithm capable of identifying areas in a cat photo, accurately detecting red and orange regions. The detection algorithm uses HSV values for colour matching, with results displayed in green bounding boxes. The object detection system showed promise in a Gazebo simulation, identifying various coloured objects but faced challenges with complex shapes and lighting conditions.

Figure 3.1 shows butterflybot generating a 2D map using Cartographer. Two mapping stages were conducted: static and dynamic. The results indicated that the Cartographer SLAM algorithm showed promise but struggled due to flight instability, leading to issues like overlapping submap outputs and inconsistent data during dynamic mapping.



*Fig.3.1 Producing a 2D occupancy grid map in Rviz.*

Due to time and resource constraints, extensive autonomous navigation tests were not conducted. Instead, evaluation scenarios were developed to assess the quadcopter’s integration with object detection, mapping, and obstacle avoidance systems. Two main scenarios include the Obstacle-rich Path Test, focusing on LiDAR-based mapping and obstacle avoidance, and the Precision Landing Test, utilising colour markers for landing accuracy.

**4 Discussion and Reflection**

This summary delineates the obstacles encountered during the simulation of a novel quadcopter drone design and suggests potential enhancements for the future. The drone was effectively simulated by the team; however, the performance was subpar due to challenges with CAD model export and ROS 2 integration. Although mesh decimation alleviated these concerns, the navigation stability and spatial awareness were all negatively affected by the absence of sensor fusion, the limitations of 2D LiDAR for vertical mapping, compatibility issues with the boundingbox\_camera, and the underutilisation of IMU data. The future work will concentrate on the integration of an RGB sensor, a 3D scanning solution, the completion of the TF tree, the resolution of flight instability, and the utilisation of IMU data for pose estimation to improve the overall performance of the system.

**5 Bill of Materials (BoM)**

The BoM, available via the project’s GitHub repository [2], describes the parts that would be used to build the robot in real life. The BoM features brushless motors, a propellor set, an Arduino Pro Mini microcontroller board, an IMU with 6 degrees of freedom, a 1000 mAh battery, and standard wiring and connectors, all housed by a 3D printed frame. The total cost is £155.48, which is less than half the maximum allowed budget of £350. The remaining balance could be used to add extra sensors to the robot as needed.

**6 Conclusions**

The butterflybot project successfully replaced its unstable flapping-wing prototype with a lightweight, ∞-shaped quad-rotor airframe that aligns its mass and geometric centers for precise attitude control while retaining its signature butterfly wings. In simulation, the team integrated a URDF/SDF model into ROS 2 Jazzy and Gazebo Harmonic, bridged LiDAR, IMU and camera sensors, and validated color-based object detection and Cartographer SLAM for mapping in GPS-denied environments. Attempts at end-to-end autonomy highlighted compute and interface bottlenecks—propeller joint bridging and workstation performance limited ArduPilot SITL integration—leaving full Nav2 trajectory tracking as future work. With a bill of materials under £175, the design meets budget constraints and allows for additional sensors. Moving forward, the team will implement LiDAR-IMU-vision fusion via an Extended Kalman Filter, migrate to hardware-in-the-loop flight controllers, and deploy Nav2 for six-degree-of-freedom path-following. These steps will bridge the sim-to-real gap and pave the way for a silent, agile inspection platform.

**Appendix**

**Group Member Contributions:**

* Alvaro – I helped by debugging important simulation files and working with Gideon to add LiDAR, IMU and camera sensors. We addressed issues with launching ROS2 and Gazebo and made the system more stable. I was responsible for changing the design from butterfly to quadcopter, suggesting a propulsion system that relies on thrust. After that, I set up ArduPilot and tested it with a simulated drone, confirming that it worked, but hardware issues made me stop the project. I then suggested and tested a way for the robot to move using thrusters. Since I was given the autonomous flight task late in the project, I investigated several solutions and kept a record of my progress on GitHub to support future work.
* Gideon – As deputy project manager, I led initial planning for Assignment 3 and collaborated with Alvaro on mapping the simulated environment. We faced several coding issues due to misaligned transforms, which I worked to correct by debugging various configuration files and the .sdf files. To enhance group communication, I regularly checked on group members and ultimately encouraged the creation of an up-to-date GitHub repository. I also outlined tasks for the group before bench inspection. I drew inspiration from existing packages to create mapping sub-packages, successfully visualised the drone in RViz post-bench inspection, and integrated the cartographer\_ros package for mapping. Finally, I created templates for the Executive Summary and Main Report, and helped to make the final work more cohesive. Although we couldn't resolve the drone's flight issues, which adversely impacted the quality of the maps, I valued the progress, learning, and teamwork throughout the project.
* Junyang – I led the conversion of refined STL meshes into URDF/SDF for Gazebo, defined link/joint hierarchies, configured LiDAR, IMU, and bounding-box camera sensors, ros\_gz\_bridge, consolidated launch scripts, attempted flapping-wing stabilization, then transitioned to a quadrotor. I resolved model conflicts, created test worlds, built TF/RViz setups, and documented design decisions and results.
* Yanzhang – I oversaw the environmental setup, sensor configuration and dynamic flight simulation in the origin bionic butterfly project. After discovering that Gazebo was unable to handle fluid dynamics, I participated in converting the design into a butterfly-shaped unmanned aerial vehicle to achieve SLAM. I designed an object recognition system based on HSV, was responsible for handling technical issues, and managed the GitHub repository and documentation writing. Although the final result of the project was not satisfactory, this experience made me deeply understand the value of teamwork and learn the importance of effective division of labour and communication.
* Yifan – I proposed replacing the original flapping-wing concept with a four-propeller architecture, carried out the necessary research, and completed the full quad-rotor design and SolidWorks model. I also handled the workflow for bringing the CAD data into ROS: exporting the STL assemblies to URDF, editing the corresponding SDF files, and verifying that the model launches correctly in the ROS 2–Gazebo environment. In parallel, I surveyed the literature on autonomous navigation, compiled potential solutions, and—together with Alvaro—implemented and tested several alternative approaches for waypoint flight in simulation.
* Zijin – Reconstructed the simulation model by converting STL files into URDF and SDF formats, and migrated components from the previous design. Updated the sensor configuration, including the addition of an IMU and modifications to LiDAR and camera settings. Authored the initial launch file, resolved ROS-Gazebo bridge issues, contributed to environment modelling, and co-developed the TF tree and RViz visualization setup.

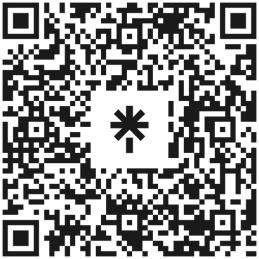
**References:**

[1] “ROS 2 Cartographer — ROS 2 workshop documentation,” *Readthedocs.io*, 2022. <https://ros2-industrial-workshop.readthedocs.io/en/latest/_source/navigation/ROS2-Cartographer.html>

[2] Y. Wang, G. Tladi, A. Mesa Giner, Z. Wu, Y. Wang, and J. Xiao, “ELEC330\_Butterfly/BoM at main · YanzhangWang/ELEC330\_Butterfly,” *GitHub*, 2025. https://github.com/YanzhangWang/ELEC330\_Butterfly/tree/main/BoM (accessed May 18, 2025).

**Multimedia:**

Access the project team’s photos and videos via the QR Code or link below:



If unable to scan the QR Code, or if it does not function as expected, kindly access the group’s multimedia via the following links:

* Linktree: <https://linktr.ee/NectarWings>
* GitHub Repository: <https://github.com/YanzhangWang/ELEC330_Butterfly>
* Google Drive Folder: <https://drive.google.com/drive/folders/1MfXtSQJq7_PgpVzn3WEl8bAbTliOOvM7>