Intelligent Ground Vehicle Competition 2025

Monash University

Monash Connected Autonomous Vehicle

Design Report

Asterius MKII

MCAV ESDA Lead
Aryeh Rathner

Faculty Advisors
Pascal Mater   Dr. Hoam Chung   Professor Hai Vu

Electrical	Mechanical	Software	Industrial Design
Ben McGaw	Matthew Lowe	William Tioe	Jessie Li
Hao Yu Wong	Richard Quan	Ricky Zhang	Wilson Hidayat
Harrison Crofts	Hari Guggilam	Yide Tao	Aiman Rahman
Shawn Hoon	Brandon Tran	Samuel Tri	Rohit Lal
Kin Wai Wong	Pellegrina Nheu	Ishaan Chowdhary	
Kenny Nguyen	Finn Didier	Huy Nguyen	
Oliver Papillo	Vidath Jayasooriya	Bhagesh Shetty	
	Alan Wu	Daniel Seow	
	Insaf Imran	Guanyue Li	
	Johnny Ou	Sanithma Mahawithanage	
	Joshua Chan		
	Kynan Reeves		

I certify that the design and development of the vehicle described in this report is equivalent to the work completed in a senior design class.

P.MATER

Date Submitted: 15/05/2025

** Names are also hyperlinks to members' emails.*

Contents

1 DESIGN PROCESS AND TEAM ORGANISATION

1.1 Introduction

Monash Connected Autonomous Vehicle is a student team based in Monash University that researches and develops autonomous and intelligent transport systems. We have a focus on creating efficient, sustainable, and safe means of transport. Our entry to the 32nd IGVC is named after Asterius, a giant of Greek mythology representing our ambition.

1.2 Organisation

The team is divided into four sub-teams: electrical, mechanical, software, and industrial design. Each sub-team is responsible for several sub-projects corresponding to the specific components of the vehicle. Sub-project members met separately on a weekly basis while the whole team met on a fortnightly basis to discuss updates and major issues. Google Drive was used to manage the design cycle documentation including Design Reports, Milestones, Timelining and Bill of Materials, while Notion was adopted for meeting minutes and task management.

1.3 Design Assumptions and Process

Before commencing the research phase, the team outlined the high-level assumptions and requirements for this project. These were determined by an analysis of the IGVC rules and an extrapolation to what would be required to achieve success. To address the assumptions and requirements, the team underwent a rigorous project design cycle.

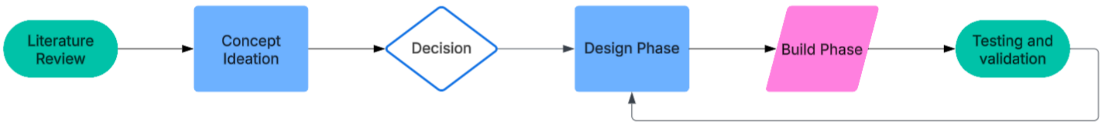

Figure 1: Asterius MKII Design Cycle

The literature review and concept ideation are primarily conducted in sub-projects, with each project sharing their findings and the team making unanimous design decisions moving into the design cycle. Each sub-project identified two viable solutions to meet the project requirements, with the most suitable chosen for further development and the other kept as an alternative.

The design phase involved sub-system design, planning testing, simulations and failure mode identification. Mechanically, subsequent prototypes were constructed to demonstrate proof of concepts. Mechanical manufacturing and electrical assembly were undertaken in the build phase, followed by identifying points of failure and implementing changes where necessary to validate the design.

2 Vehicle Architecture

2.1 Significant Mechanical, Power and Electronic Components

The significant mechanical components include the chassis, swing arms, gear boxes, front wheels, pneumatic shocks, trailing wheel, and electrical component plates. These components are further explored in Section ??.

The electrical components include the flight controller, SwiftNav Piksi Multi PCB, relay board, remote switch, E-Stop, SwiftNav GNSS antenna, ZED 2i stereo camera, Velodyne VLP16 LiDAR, ESCs, NVIDIA Jetson Orin Nano, and motors. The significant power components include the battery and a 12V buck converter. More specifics are detailed in Section ??.

2.2 Safety Devices

Safe operation of the vehicle is achieved with the help of the emergency stop function. This can be activated through the onboard manual E-Stop button or the wireless E-Stop, both of which will cause the onboard computer to cease driving the motors. The robot also has a safety light LED strip around the base of the GPS fin to indicate that it is operating and its operating mode. See Section **??** for further information.

2.3 Significant Software Modules

The significant software modules of the Asterius MkII are:

- A lane and pothole detection module implemented by the ZED SDK and OpenCV.
- Obstacle detection by the LiDAR module.
- Localisation of the robot using Robot Localisation Package, ROS2 SLAM Toolbox, GPS and LiDAR modules.

A Jetson Orin Nano (8 GB) running ROS 2 Humble on Ubuntu 22.04 Jammy Jellyfish integrates data from a ZED2i stereo camera, Piksi GPS, and Velodyne VLP-16 LiDAR. Nav2 then performs SLAM, localisation, and path-planning to issue linear and angular velocity commands to control Asterius MkII's movement.

A high-level view of the Asterius MkII's software stack is shown in Figure 2 and will be explained further in Section **??**.

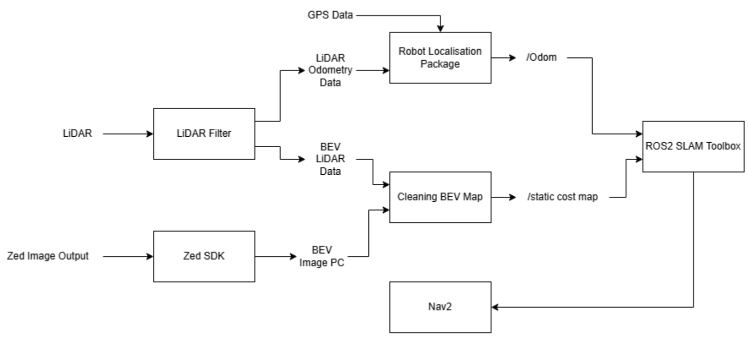

Figure 2: Software Stack Flow for the Asterius MkII

3 Key Innovations

3.1 Mechanical

Custom planetary gearbox: An in-house produced two-stage planetary gearbox that delivers a 64:1 reduction, capable of withstanding high torque without failure. See Section 4.2.1 for more information.

Sheet metal chassis: Integrates industrial design and engineering through the aggressive shark-inspired shapes and lines while ensuring the overall stability and integrity of the vehicle. Section 4.2.2 further explores the chassis.

Independent Suspension: Reverse adaptation of the trailing arm castor and swing arm system developed to provide stability while optimising space. Section 4.2.3 goes into more detail.

Modular electrical component plates: Two separate acrylic plates holding electrical components are attached to the internal chassis using shelf supports, favouring ease of disassembly.

3.2 Electrical

Typically used in unmanned aerial vehicles, drones, and other aircraft, Asterius MkII's motors are controlled by a flight controller (FC). We opted to use a FC due to its compatibility with our sensor suite and its affordability.

The motors are controlled by the FC and Jetson Orin Nano as follows:

- The FC receives commands from the Jetson Orin Nano via MAVROS. MAVROS is a custom ROS2 interface that bridges communication between ArduPilot and the Jetson Orin Nano.
- ArduPilot processes these commands and sends them to the ESCs to control the speed of the motors. ArduPilot is a pre-installed firmware on the FC.

3.3 Software

Unified Lane and Pothole Detection: Creating a scaled costmap followed by Nav2 through performing Inverse Perspective Mapping (IPM) on live camera feed—masking anything considered as lanes or potholes found in the Region of Interest (ROI). This is further detailed in Section 6.1.

LiDAR Obstacle Detection and Odometry: Using the Velodyne VLP16 LiDAR in conjunction with the ZED2i to create a late sensor fusion costmap, and utilising the Iterative Closest Point (ICP) algorithm for odometry. Further exploration can be found in Section 6.2.

4 Mechanical Design

4.1 Overview of Mechanical Design

The Mechanical subteam intended to build a vehicle that creates a robust, reliable, and efficient physical platform to support sensors, actuators, and control systems for safe and effective self-driving operation. It should not only support the integration of electrical components, but also optimise the performance of the software suite.

We utilised Onshape—a collaborative CAD software—to model and iterate on the design prior to physical fabrication to ensure precise fitment of all components. Onshape was also used to conduct simulations for design validation.

Components such as motors, wheels, and fasteners were acquired ready-made, allowing our team to focus design efforts on the custom chassis, drivetrain, and suspension elements. The design process involved close collaboration between the mechanical and electrical subteams to ensure seamless integration of all systems, particularly regarding space allocation and mounting provisions.

4.2 Significant Mechanical Components

4.2.1 Drivetrain

The drivetrain system is centered around two in-hub motors designed to deliver the necessary torque and speed control for the vehicle. To achieve the required 64:1 reduction ratio, a two-stage planetary gearbox is implemented, with each stage providing an 8:1 reduction utilising steel sun, planet and ring gears. The output carrier of the second stage is directly bolted to the 13×5.00–6 buggy wheel rim, eliminating the need for a separate output shaft. The gearbox housing was CNC machined from lightweight aluminum and all gears were made from hardened steel to withstand large loads.

The motor is connected to the gearbox via a modified 15-tooth steel gear secured to its D-shaped shaft with a machined bore and cross-drilled M4 hole. The gearbox's internal components—including the planet gear shafts and the central bearing—are secured using circlips. The motor and gearbox are mounted by incorporating a circular solid steel bracket into the swing arm design, welded concentrically to the end of the RHS arm, illustrated in Figure 3.

To enhance cooling and weather resistance, the design includes 3D-printed ABS motor covers that provide cooling, held in place by magnets. Due to the geometric constraints of the gearbox and the chosen wheel rim, a spacer is required to ensure proper mounting. The selection of 13 × 5.00–6 buggy wheels was based on their spacious internal area, accommodating the in-hub motor design, and the availability of mounting features.

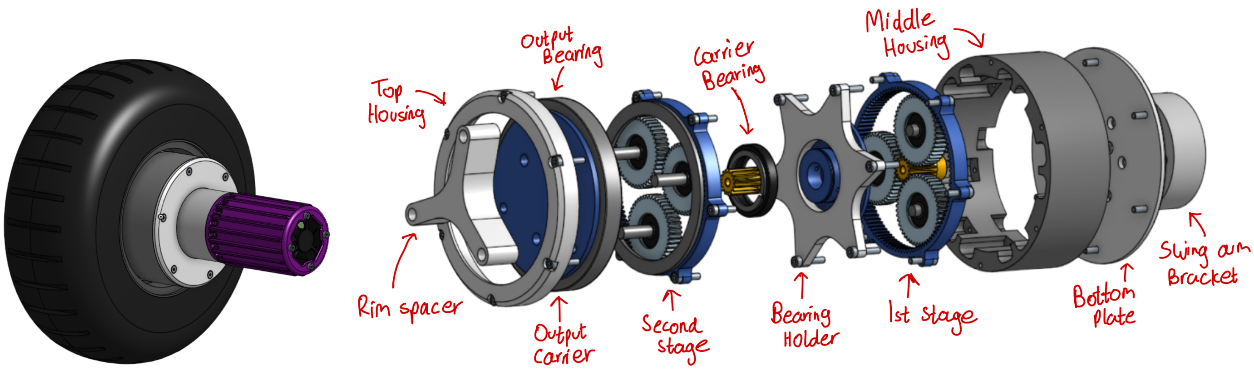

Figure 3: (Left) Rear view of open gearbox, (Right) Exploded view of gearbox

4.2.2 Frame Structure, Housing, and Structural Design

The chassis and lids were laser cut from 2 mm aluminium sheet metal, then bent with a brake press and welded together. The aluminium material and cut-outs throughout the body reduce the overall weight of the vehicle. A welded brace was included in the midsection to reduce bending stresses associated with the chassis's parallelogram shape whilst serving as the mounting point for the acrylic divider.

3D-printed mounting fins for the LiDAR and GPS, along with a camera mount, are positioned on the top surface of the lid to enhance the field of view. The payload is strategically placed in the front section of the chassis for a low center of gravity, secured with straps.

Electrical components are attached onto two acrylic plates held by 3D-printed shelf supports—inspired by bookshelves. The acrylic plates are able to stay in place without needing fasteners. The modular and screwless design allows for convenient access to the components, allowing for easy retrofitting.

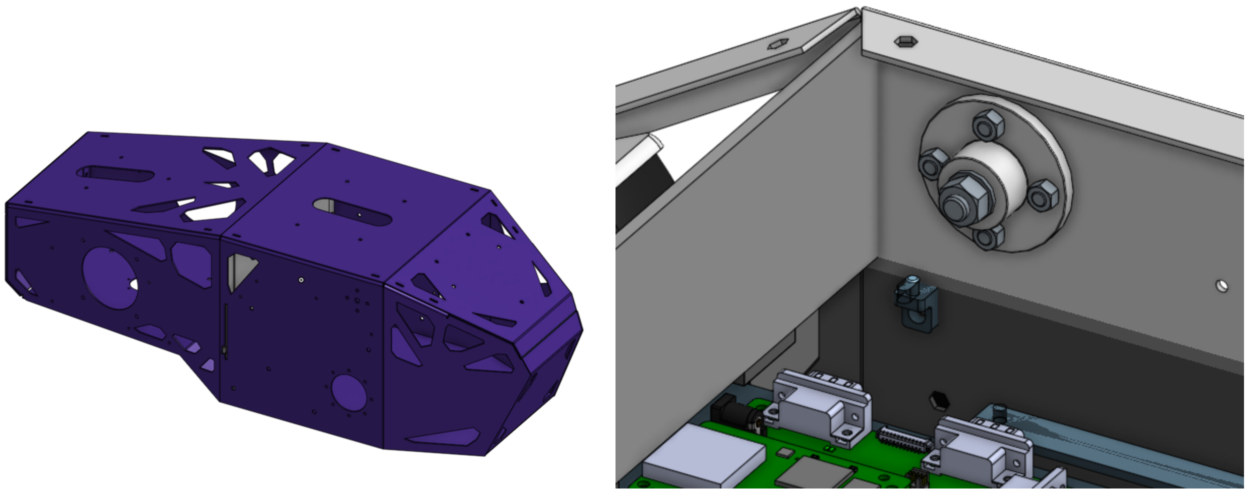

Figure 4: (Left) Aluminium chassis body, (Right) Shelf supports for electrical components

4.2.3 Suspension

To ensure that sensors receive data accurately with minimal vibrational interference, an independent suspension system has been implemented.

Front Drive Suspension: For the two front driving wheels, an entirely unique, reversed adaptation of the trailing swing arm system was developed. Taking inspiration from rear-wheel motorcycle systems, this concept leverages the benefits of independent suspension and adjustable ride height. The inclusion of adjustable pneumatic shocks allows the vehicle to adapt to various terrains and payloads. Furthermore, locating the swing arm and drivetrain assembly on the exterior of the vehicle provides more space within the chassis for the other subsystems.

Castor Wheel Suspension: The rear caster wheel utilises a more conventional trailing arm suspension. The design allows the wheel to follow a radial path in compression, preferred over a simpler vertical path design due to its enhanced reliability on inclines.

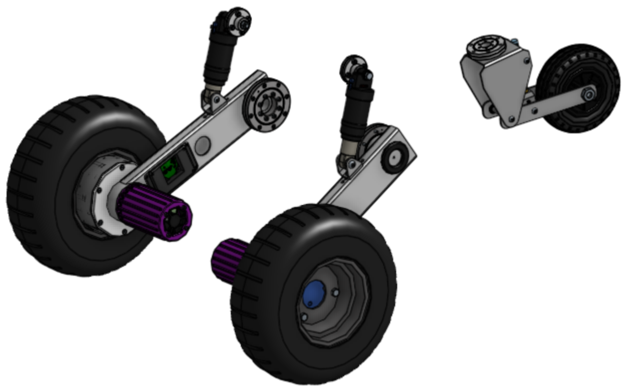

Figure 5: Isolated view of front and rear suspension systems

4.3 Weatherproofing

The chassis is made of aluminium—naturally resistant to corrosion, and is environmentally friendly. To increase its durability and aesthetic, a powder coating layer was implemented to protect against extreme weather conditions and moisture. Additionally, cutouts in the chassis were filled with polypropylene sheets and bonded with double-sided acrylic tape. The use of cable wraps and rubber grommets protect sensitive electrical components and wirings from the external environment.

Air intakes and exhausts are covered by 3D-printed ABS louvres. These angled vents allow air to pass through whilst deflecting direct splashes and larger debris. A dust mesh on each louvre further safeguards the internals from the environment.

5 Electronic and Power Design

5.1 Overview

The electrical design of *Asterius MkII* focuses on precise motor control, appropriate power management, sensor integration, and efficient component placement.

5.2 Power Distribution System

The *Asterius MkII* utilises a 24V 10Ah Li-ion battery with an integrated Battery Management System (BMS) and charging circuit, along with a master switch for safe power engagement. The battery directly powers the ESCs and flight controller, with 30A fuses protecting the ESC lines and 2A fuses on the 12V rail for the LiDAR and Jetson Orin Nano. The 12V rail is supplied through a buck converter.

During typical operation, the motors draw approximately 5A each. All 24V power connections employ XT90S connectors and 8 AWG wiring to ensure reliable high-current delivery, while the 12V components draw up to 2A for stable operation.

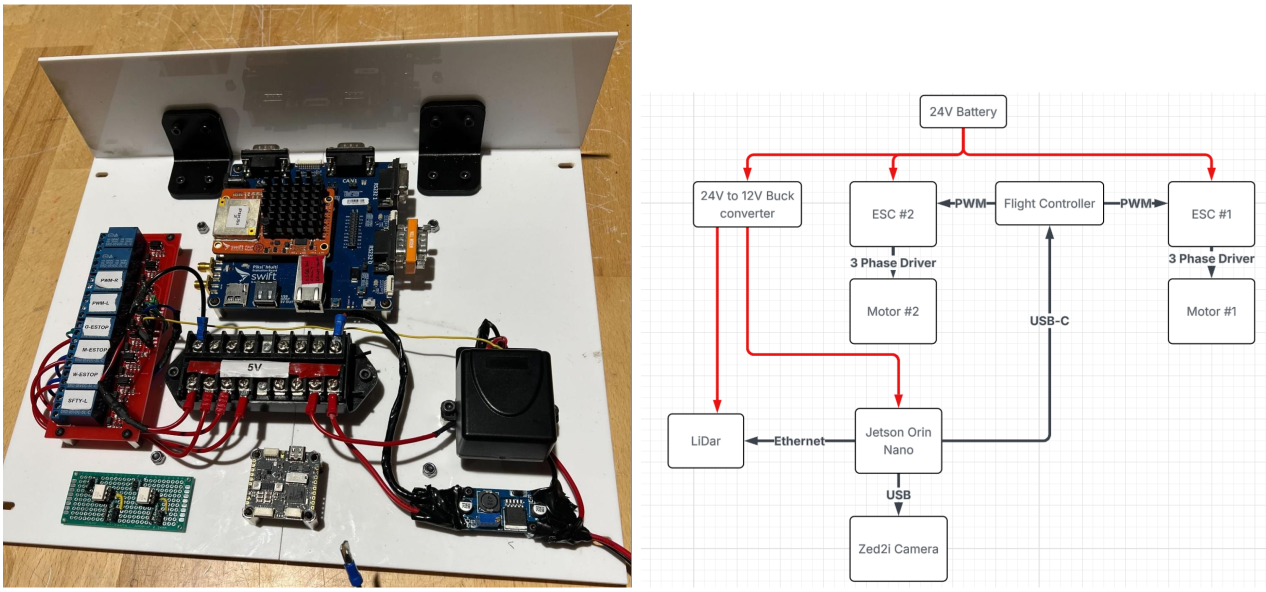

Figure 6: (Left)Acrylic shelf for mounting electrical components (Right) Wiring schematic

Table 1: Battery specifications for the Li-ion battery used in the vehicle

Specification	Value
Capacity	10Ah
Voltage	7S (29.4V)
Discharge Current	11A
Recharge Time	40 minutes
Expected Run Time	10 minutes

5.3 Electronics Suite

The *Asterius MkII* vehicle utilises Castle Mamba X ESCs (7S-capable) controlled through PWM signals from the flight controller (FC) running ArduPilot firmware. The FC receives PWM commands from the Jetson Orin Nano via MAVROS, a communication interface between ROS2 and ArduPilot.

The system incorporates a LiDAR for 3D obstacle detection and environmental mapping, complemented by a Zed2i stereo camera that adds depth perception for enhanced lane tracking. The Jetson continuously processes sensor data and calculates control outputs, which are then transmitted back to the FC. This communication allows for precise motor control through dynamic PWM signal adjustments to the ESCs.

5.4 Mechanical and Wireless E-Stop Systems

The E-Stop subsystem serves as a critical safety mechanism that enables the immediate shutdown of the robot. The onboard E-Stop is a red latching button wired to a relay and is located at the rear of the robot.

Concurrently, a wireless E-Stop was chosen to operate via a radio frequency (RF) communication protocol due to its long range and reliable connectivity. This remote switch communicates with an onboard receiver also linked to the relay, functioning the same as the physical E-Stop.

Activation of either the onboard or wireless E-Stop cuts off the PWM signal from the flight controller to the ESCs, thereby stopping motor operation.

6 Software System

6.1 ZED2i Camera — Lane Detection

The ZED2i camera is *Asterius MkII*'s main method of lane-keeping and pothole avoidance. It integrates with ROS2 via the ZED SDK, allowing various topics to retrieve raw camera images and depth data.

The raw image data is processed with an OpenCV pipeline. Upon receipt of the colour image from the ZED2i, a grayscale conversion is applied over a region of interest (ROI) generated after performing Inverse Perspective Mapping (IPM). The ROI is then masked and passed through a Gaussian blur. Thresholding is applied to detect lane lines or potholes.

The processed ROI is then transformed into a Bird's Eye View (BEV) map. This is later fused with the LiDAR map through a late sensor fusion setup. The resulting output is converted into a costmap for use by the ROS2 SLAM Toolbox and Nav2.

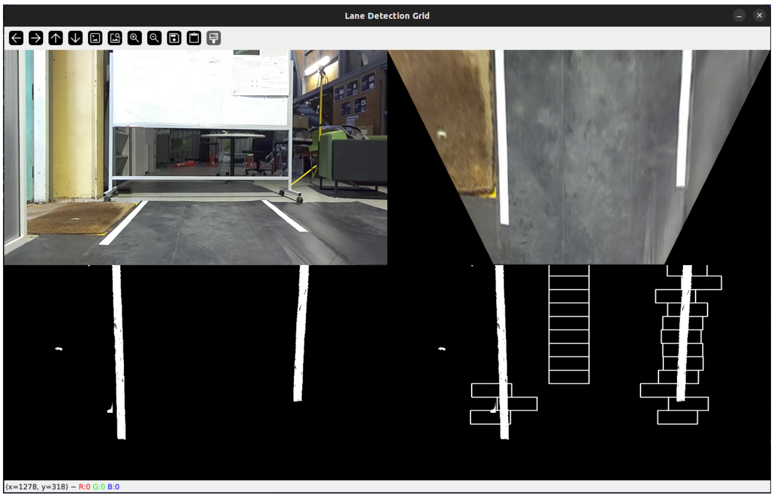

Figure 7: Top left: Live feed from ZED2i. Top right: BEV. Bottom row: Thresholded images detecting and highlighting lanes.

6.2 Velodyne LiDAR — Obstacle Detection and LiDAR Odometry

The Velodyne VLP-16 LiDAR sensor provides 3D point cloud data used for real-time obstacle detection and vehicle localisation through LiDAR odometry.

Raw LiDAR data is filtered using Point Cloud Library (PCL) to reduce noise and exclude points below a height threshold. The remaining data is projected into a 2D BEV point cloud, which forms the static local costmap used in obstacle avoidance.

An Iterative Closest Point (ICP) algorithm compares sequential LiDAR frames to derive odometry. The Robot Localisation Package then fuses this with GPS data to provide accurate vehicle positioning.

Once BEV maps from both the camera and LiDAR are generated, they are merged into a static local costmap. This costmap, together with odometry data, is used to update the global costmap required by Nav2 through the SLAM Toolbox.

6.3 Piksi GPS Module — GPS Approximation

Asterius MkII uses Swiftnav's Piksi Multi GPS modules for centimetre-level positioning. One unit is mounted on the vehicle; another serves as a fixed-position base station.

The rover module receives satellite signals to calculate latitude, longitude, altitude, velocity, and heading. The base station continuously sends correction data to reduce signal errors via differential GPS (DGPS), significantly improving localisation accuracy.

6.4 Robot Localisation Package

This package fuses global GPS data with LiDAR odometry to estimate the robot's position within the global costmap. It uses an Unscented Kalman Filter (UKF) to merge noisy GPS input and smoother LiDAR-derived data, reducing localisation errors from non-linear robot motion and GPS signal noise.

6.5 ROS2 SLAM Toolbox

The ROS2 SLAM Toolbox enables Simultaneous Localisation and Mapping (SLAM), allowing the robot to create and update a map of its surroundings while tracking its location.

Asterius MkII uses SLAM Toolbox to generate a global costmap, continuously refined with data from the Robot Localisation Package and fused sensor inputs (ZED2i and LiDAR). This map feeds into Nav2 for global path planning and dynamic obstacle avoidance.

6.6 Nav2 — Path Planning

Asterius MkII navigates using the Nav2 stack, which manages layered costmaps and coordinates modular planning, execution, and recovery via behaviour trees (BTs).

SLAM Toolbox updates the '/map', which populates the static layer of the Global Costmap Server. The Local Costmap Server receives fused BEV maps from the camera and LiDAR to build a dynamic obstacle layer.

Nav2 uses its built-in BT Navigator with the `NavigateToPose` tree. This structure constantly checks path validity. If issues arise, the Planner Server is triggered via '/compute_path_to_pose' to replan using the global costmap.

The path is then refined by the Smoother Server and tracked by the Controller Server, which compares the path to the live local costmap. The Controller outputs high-rate velocity commands, passed to the Velocity Smoother to filter out jitter and ensure safe accelerations. This finalised velocity is published as the `/cmd_vel` Twist message.

If a failure occurs (e.g., lost path, stalling), the Recover Server activates with tasks like clearing the costmap or repositioning the robot. The Lifecycle Manager orchestrates all Nav2 servers, maintaining correct operational states from startup to shutdown.

7 Cyber Security Analysis Using Risk Management Framework (RMF)

The NIST Risk Management Framework (RMF) is a methodology for iteratively maintaining security in an information system. It aims to effectively triage any cyber-attack or malicious action quickly and effectively.

- **Prepare:** Assign roles and develop a strategy to apply the RMF effectively.
- **Categorise:** Identify and group information systems based on the impact of potential breaches.
- **Select:** Choose security controls to mitigate risks to the system.
- **Implement:** Apply controls and document implementations.
- **Assess:** Test and evaluate control effectiveness.
- **Authorise:** Submit results for approval from competition authorities to operate.
- **Monitor:** Continuously observe risks and update controls as needed.

7.1 Threats and Impact Analysis

A significant threat considered is that of a rival team tampering with vehicle systems in the pit area. Table 2 outlines the systems potentially at risk and the resulting impact on security dimensions.

Table 2: Potential threats for electrical components

System	Threat description	Integrity	Availability	Confidentiality
LIDAR	The Velodyne information box / onboard computer can be modified to transmit inaccurate sensor data.	High	Medium	Low
Wireless E-Stop	If a wireless E-Stop signal is intercepted and misused, a rival team could stop the run.	High	High	Medium
Jetson Orin Nano	If a threat actor is able to obtain privileged access, commands could be sent to modify the vehicle's behaviour.	High	Medium	Medium
SwiftNav GNSS Antenna	Jamming of signal to and from the antenna can compromise vehicle odometry.	Medium	High	Low

7.2 Cyber Controls

The following table outlines key cybersecurity controls applied to safeguard critical components of *Asterius MkII*.

Table 3: List of cyber controls implemented into the vehicle

Control ID	Goal	Implementation	Efficacy
AC-1 (Access Control)	Prevent unauthorized access to vehicle software codebase.	Use of role-based access control (RBAC) and multifactor authentication for privileged accounts (i.e. GitHub, Slack).	**High:** Limits access to software accounts, preventing unauthorised access to sensitive information.
AC-17 (Remote Access)	Secure wireless E-Stop commands.	Use rolling code mechanisms and strong encryption.	**High:** Reduces the risk of interception and misuse of emergency stop commands.
IA-5 (Authenticator Management)	Ensure secure remote access to the Jetson through multiple layers of encryption.	The Jetson can only be accessed with SSH authentication either through an Ethernet connection directly to the robot or an encrypted VPN tunnel (Tailscale).	**High:** Multiple layers of encryption (VPN, SSH) must be bypassed or broken to obtain access to the Jetson.
CM-2 (Baseline Configuration)	In the event that the codebase is accessed, prevent unauthorised modifications to the codebase.	Implement branch protection rules, require pull requests for changes, and enable audit logging.	**High:** Ensures accountability and traceability of all changes, reducing the risk of unauthorised modifications.

8 Vehicle Analysis

8.1 Lessons Learnt During Construction and System Integration

The biggest challenge our team faced was realistically creating a timeline to follow. Our design cycle was 10 months long, with the condition that the mechanical build is completed before the electrical and software teams can make significant progress. As a result, in the last few weeks leading up to the competition, many hours were spent catching up on wiring and writing code.

For future iterations, the team will use Asterius MkII as a starting point to base designs off of to reduce time needed to research and thus allowing for faster solution generation. Starting the design process earlier grants more time for testing and refinement. The three subsystems will also be working simultaneously instead of sequentially, as there are many methods of testing without needing the vehicle built and finalised.

8.2 Top Hardware Failures and Mitigation Strategies (Pre-Competition)

Table 4: List of potential hardware failures, causes and possible mitigations

Hardware Failure	Cause	Mitigation
Electrical		
Loss of electrical connections between components	Loose wire connections	Solder connections where possible or use multiple pin connectors.
Electrical Shorts	Exposed wire causing sparks	Protect wire using heat shrink or electrical tape around the exposed sections.
Overvoltage	Excessive current draw from the motor	Use fuses for the motor power wires. Replace fuses with higher rated fuses and/or reduce the speed of the car.
Malfunctioning signals and control	Wires are not connected to the correct pins	Reconnect the wires to the correct pins and then update either the schematics or assembly guide.
Mechanical		
Gearbox Failure	Gear stripping/shearing due to non-concentric parts, lack of lubrication or failed press-fits	Use high-strength gear materials (steel), ensure proper lubrication, use manufacturing processes and physical validation that ensure concentricity and fitment.
Fastener Failure	Fastener failure	Use thread-locking compounds (Loctite), safety washers, appropriate tightening torques, lock nuts and an appropriate number and size of fasteners.
Swing Arm Failure	Swing arm physically buckles	Use high-strength steel with appropriate dimensions. Validate design using CAD and FEA.
Caster Wheel Failure	Mounting spacer compresses, causing the bracket to rub against the chassis, providing friction and preventing movement	Use a machined aluminium mounting spacer that will not compress.

8.3 Safety, Reliability and Durability During Design

Throughout the duration of this project, many different considerations were addressed involving safety, reliability and durability. Two types of E-Stops that have been added to the vehicle: wireless and mechanical. The mechanical E-Stop allows for a guaranteed shut down of the vehicle by pushing a button and cutting the connection to the circuit. The wireless E-Stop allows for the vehicle to be turned off remotely by sending a signal to a switch.

Safety lights are also crucial to the design of the vehicle, allowing for users to view the status of the vehicle—with flashing lights being autonomous and solid lights being manual.

The battery used for the vehicle comes in with an inbuilt BMS that controls the amount of power utilised. The electrical circuit is grounded to the chassis to mitigate excess current. Moreover, a fuse was used to safeguard any damage to vital electrical components.

Weatherproofing ensures water and dust do not damage the electrical components. To prevent such an event, most electrical components are fully enclosed within the chassis. Cut-outs are covered with polypropylene sheets and double sided acrylic tape to compartmentalise the chassis interior.

During the testing of the vehicle, it is expected that the electrical components may heat up. For this reason, there is an opening for passive cooling.

8.4 8.4 Key Hardware Failure Points & Modes and Resolutions (During Competition)

Table 5: List of hardware failures and resolution strategies

Hardware failure	**Resolution strategies**
Gearbox failure	Disassemble and reassemble to address possible gear misalignment. Inspect gears throughout; if wear or shearing is identified, replace with spare gears. Clean thoroughly to remove any debris and reapply lubricant. Replace gaskets and reapply RTV sealant if necessary.
Swing arm air shock locking/sticking	Manually adjust shock stiffness and/or air volume to ensure correct functionality.
Acrylic mounting plate or 3D part breaking	Replace with spare parts or repair using tape or glue.
Fastener failure	Replace the fasteners if damaged. Tighten any loose bolts and nuts, ensuring lock nuts are used.
Flat tyre/s	Assess and repair leak or tear using tyre slime or a tube patch repair kit and use a hand pump to refill the tyre.
Wire damage	Cut and strip new wires and replace faulty wire to the necessary locations.
Electrical component damage	Fix any damaged components if possible by resoldering connections; otherwise replace with spare parts if available.

8.5 Analysis of the Predicted Performance of the Vehicle

Table 6: List of performance criterion and predictions of its performance

Performance criterion	**Prediction**
Speed	2 m/s
Ramp climbing ability	25 degrees
Reaction times	∼200ms
Using GPS for waypoint navigation and localisation	0.010 m + 1 ppm
Battery life	15 mins
Distance at which obstacles are detected	100m for LiDAR
How the vehicle handles complex obstacles	Remapping logic triggered within 6 seconds of stalling
How vehicle failure points and modes are identified and addressed	Pre and post run inspections
Accuracy of arrival at navigation waypoints	±0.5 m with RTK GPS

8.6 Software Testing, Bug Tracking and Version Control

All software testing and debugging were performed on personal devices alongside the Nvidia Jetson Orin Nano 8GB Developer Kit. GitHub was relied upon for version control and project management functionality.

Testing was done either individually or as a team depending on the bug. All changes were pushed to GitHub at the end of every working session. All updates were also logged on GitHub, including any issues encountered during work. GitHub was also used for version control, which gives the members freedom to create their own branches without worrying about overwriting the work of others. Once a software module was completed, it was discussed and voted on before branches were merged for an integration test. Merging of the branches and maintaining the GitHub repository was the main responsibility of the software lead.

8.7 Physical Testing and Simulations

8.7.1 Mechanical Simulations

Chassis Finite element analysis tested the chassis's structural integrity under payload and self-weight. Loads simulating electronics, sensors, and suspension forces were applied. Results guided lightweighting via cut-outs in non-critical areas, such as the divider panel, while ensuring strength under 1000–3000N forces across key chassis components including the swing arm mounts.

Suspension Simulation SolidWorks Dynamic Motion Testing assessed the suspension system's footprint and stability. 2D simulated movement across various terrain profiles enabled accurate prediction of required spring damping and stiffness for stable motion.

FEA assessments evaluated the structural integrity of the swing-arm suspension system, in dealing with the weight loads of the chassis combined with the reaction forces from the wheel.

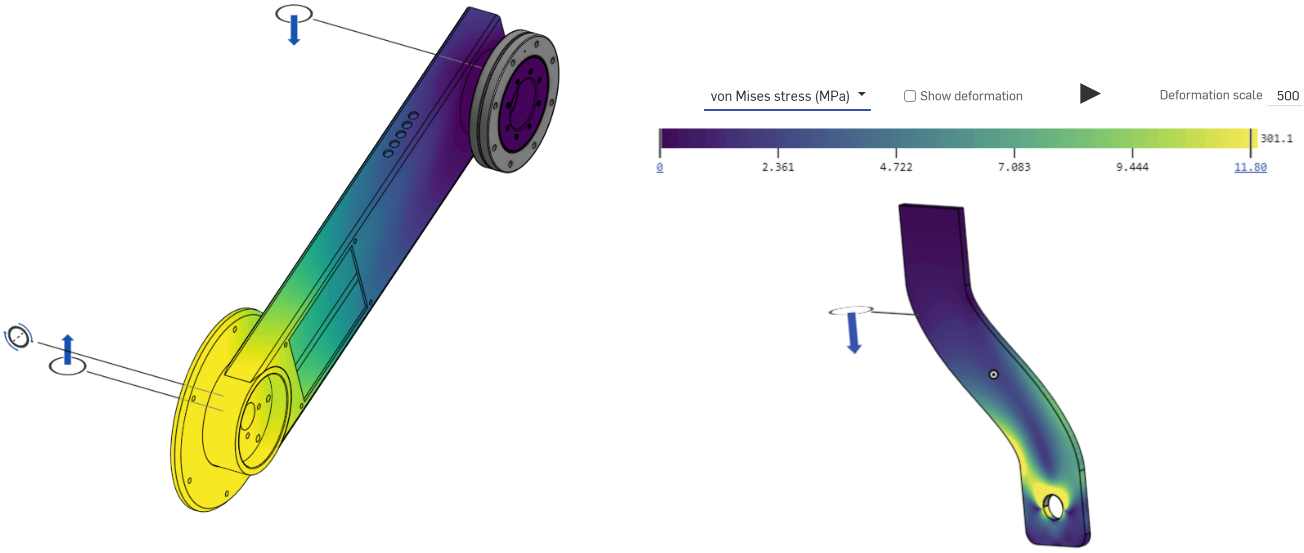

Figure 8: FEA results of Swing-Arm and Caster Support Arm Suspension Systems

The swing-arm suspension system was tested under three conservative load conditions: a 400 N bending moment from wheel contact, a 5000 Nmm reaction force, and a 500 N weight load from the chassis. The simulation revealed a maximum deformation of just 0.384 mm, and a maximum stress of only 49.76 MPa, with a minimum safety factor of 1.7. This confirms the design is not only suitable for operation but robust under unexpected and severe forces.

Static FEA on the caster wheel configuration used a 200 N load—intentionally higher than expected operational loads. All components maintained safety factors well above 3, confirming that material choice

and thickness were appropriate.

Physical Testing

- **Swing Arms – Moderate Impact Test:** A 20 kg weight was dropped from 150 mm onto the suspension system to simulate impacts like potholes.
  Response: Absorbed impact with no distress.
  Deformation: Negligible (<0.1 mm), full elastic recovery.
  Result: **Pass**

- **Caster Wheel Assembly – Static Load Compression Test:** 200 N applied vertically.
  Deformation: 0.5 mm at max stress point.
  Safety Factor: Estimated > 4.
  Result: **Pass**

These tests confirmed the FEA simulations and demonstrated high durability under both static and dynamic loads.

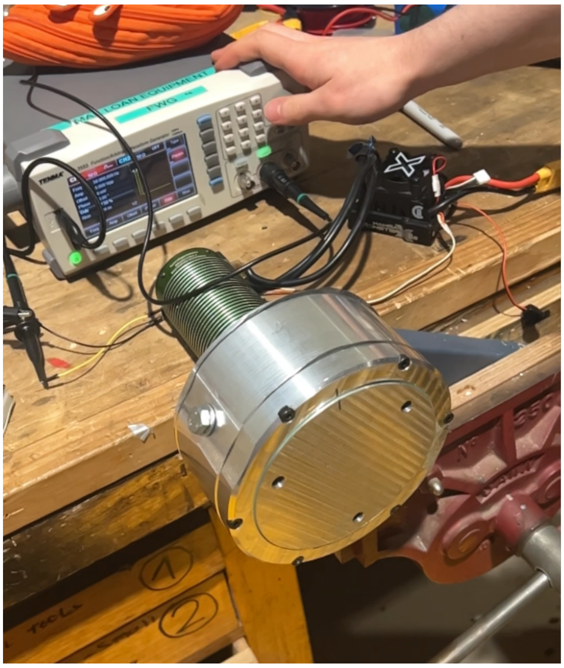

Figure 9: Your caption goes here

Drivetrain The gearbox assembly was tested by mounting the swing arm in a vice and powering the ESCs via a function generator. PWM duty cycle adjustments varied motor RPM. Audio analysis was used to detect irregularities indicating misalignments. Post-test disassembly revealed faults in gearbox carriers, planet gear concentricity, and output bearing seals, prompting redesign and remanufacture.

8.7.2 Software Simulations

Gazebo Fortress was used to test the software stack. A simulated course included lane lines and pothole blobs for camera testing and barrels and cones for LiDAR verification. The Asterius MkII model was spawned into the simulation world to evaluate camera vision, LiDAR-based mapping, and Nav2 path-planning modules.

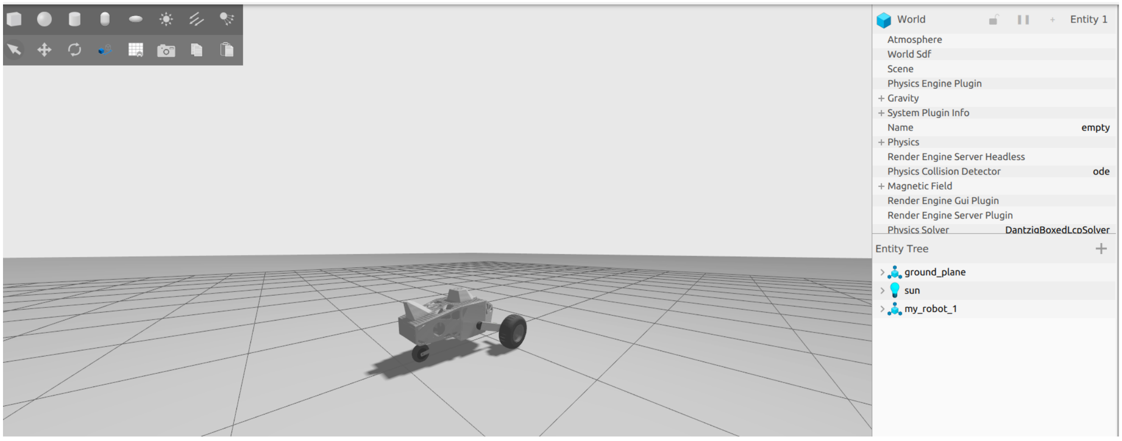

Figure 10: A model of the Asterius MkII spawned in Gazebo Fortress

9 Unique Software, Sensors and Controls for AutoNav

The unique features specifically tailored for the AutoNav challenge include the **Unified Lane and Pothole Detection Software Module**. This module is designed to focus solely on detecting lanes and potholes and generates a bird's-eye view (BEV) map. This map is then converted into a costmap, where both lanes and potholes are treated as "obstacles" alongside those identified by the LiDAR.

The **LiDAR** is also integrated with the **Iterative Closest Point (ICP)** algorithm to enhance odometry performance, providing more accurate localization. This complements the data obtained from the **Piksi GPS modules**, resulting in more reliable navigation performance for the Asterius MkII.

10 Initial Performance Assessments

Currently, the mechanical and electrical subsystems of the vehicle are functioning as intended, which has been verified through individual system-level and integration testing.

Software testing has primarily been conducted through simulation environments and controlled sensor validation. However, due to unforeseen circumstances, the vehicle must be shipped overseas earlier than expected. As a result, opportunities for full-scale physical testing are limited.

To address this constraint, a sensor-mounted testing rig is being employed to validate simulation results and ensure the software modules behave as expected in real-world scenarios. This approach allows us to continue refining performance predictions even in the absence of full vehicle trials.