ITU RAKE

Istanbul Technical University IGVC 2025 - KAYRA

Advisor: Prof. Dr. Tufan KUMBASAR

Team Captain: Elif Eylul BAYSAL (baysal22@itu.edu.tr)

Date Submitted: May 15th,2025

STATEMENT OF INTEGRITY

I certify that the Kayra vehicle described in this report was prepared by the students named below, and this work is equivalent to the project in an advanced design course. This report was prepared by the ITU RAKE team under my guidance.

Advisor: Prof. Dr. Tufan KUMBASAR

Team Members

Onder Ihsan KUL (kulo23@itu.edu.tr)

Ahmet Emre AKTAS (aktasah22@itu.edu.tr)

Kagan SEZENGOZ (sezengoz22@itu.edu.tr)

Mustafa Ata ABATA (abatay22@itu.edu.tr)

Semih Adin TURHAN (turhans23@itu.edu.tr)

Melih Emre GUNEY (guneym22@itu.edu.tr)

Table of Contents

1. Conduct of Design Process and Team Identification and Organization

1.1 Introduction

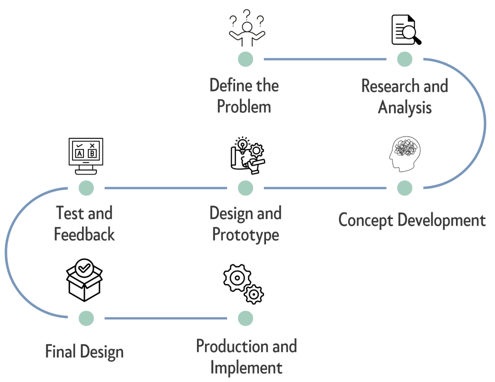
The Istanbul Technical University Robotic Search and Rescue Team (ITU RAKE) will participate in the Intelligent Ground Vehicle Competition (IGVC) AutoNav Challenge for the first time, which will be held from May 30 to June 2, 2025, at Oakland University. Composed of undergraduate students from various engineering disciplines, the team aims to gain knowledge in autonomous vehicle technologies and test the performance of the robot they have developed.
The vehicle is fully autonomous, capable of detecting obstacles and choosing the optimal path to move forward. It also features safety elements such as an emergency light and an emergency stop system. The competition tests the vehicle's ability to detect and overcome environmental obstacles, and the team has been working intensively to achieve this goal.

1.2 Team Organization

The İTÜ RAKE project team consists of a total of 20 members and is divided into four sub-teams: electronics, software, mechanics, and organization. Each sub-team has a leader responsible for ensuring tasks are completed on time, and these leaders report to the team captain. Below are the class, major,role, and total number of person-hours of the team members who will participate in the competition.

Name-Surname	Class	Major	Role	Total Number of Person-Hours
Elif Eylul Baysal	2nd	Management Engineering	Team Captain	250
Ahmet Emre Aktas	2nd	Electronics and Communication Engineering	Electronics Leader	255
Kagan Sezengoz	2nd	Mechanical Engineering	Mechanics Leader	250
Onder Ihsan Kul	2nd	Electronics and Communication Engineering	Software Leader	275
Mustafa Ata Abatay	2nd	Mathematical Engineering	Software Member	175
Melih Emre Guney	2nd	Mechanical Engineering	Mechanics Member	160
Semih Adin Turhan	1st	Electrical Engineering	Electronics Member	150

1.3 Design Assumption and Process



The design process begins with identifying the problem and understanding the competition's requirements. Past reports are reviewed, followed by research, analysis, and brainstorming. Initial concepts are developed, prototyped, and tested using simulation tools. Based on feedback and data, the final design is completed and implemented. Throughout the process, team members coordinated closely and shared regular progress updates.

2.System Architecture

2.1 Mechanical, Power, and Electronic Components

The IGVC vehicle developed by ITU RAKE, named *Kayra*, is designed to operate on both flat and uneven terrains. It features a six-wheeled, skid-steering drive system with a modified rocker-bogie suspension that enhances obstacle negotiation and stability. The chassis is built using aluminum box profiles for their strength-to-weight advantage, supported by steel sheet connectors, sigma profiles, plexiglass, and sheet metal plates to ensure optimal weight and durability.

The vehicle is powered by a 24V 30,000 mAh Li-Po battery and driven by four INTER MOTOR FD-60E-15-12 motors. Each motor is controlled by BTS7960 motor drivers and monitored using high-resolution 600-pulse rotary encoders. The control system includes one STM32F407 and two STM32F103 microcontroller boards. Encoder signals are filtered through an analog circuit to reduce noise and ensure accurate speed feedback.

2.2 Safety Devices

In order to read sensor data and process external inputs efficiently, the workload is distributed between the STM32 microcontrollers on the mainboard and the Jetson Orin to ensure maximum system performance. The slave STM32F103 boards on the mainboard collect position and velocity data from the wheel encoders and transmit this information to the master STM32F407 board via the I2C communication protocol. The master board then forwards the processed odometry data to the Jetson Orin using the UART protocol for autonomous navigation.

The Jetson Orin not only processes the incoming data from the master board but also continuously collects environmental data through cameras. These inputs are evaluated alongside IMU and GPS data to generate motion commands. The motor commands are sent back to the master board via UART, which then transmits the necessary control signals to the slave boards responsible for motor actuation.

The vehicle also features a wireless emergency stop (e-stop) system that utilizes RF modules connected to both the custom controller and the master board. An Arduino Nano is used in the controller. When the emergency stop button on the controller is pressed, the Arduino reads the signal via its digital pins and transmits it through UART to the RF module. The RF module on the master board receives the signal and communicates the e-stop condition to the board via UART, triggering the emergency stop protocol.

2.3 Software Modules

Software system consists of various interconnected modules classified based on functionality

1. **Perception:** Responsible for image segmentation and object detection using cutting-edge AI architectures
2. **Localization:** Utilizes Dual Extended Kalman Filters with multiple sensors for noise-tolerant robust pose estimation
3. **Grid-Mapping:** creates a light-weight rolling occupancy grid of the robot's surroundings by merging processed camera imagery & localization output
4. **Goal Selection:** Selects intermediate goals for the robot guided by multiple cost parameters for optimized path-planning
5. **Navigation:** Employs path-planning and path-tracking algorithms to control wheel speeds to effectively reach the selected intermediate goal

All software is developed using ROS2, with simulation conducted in Gazebo. The onboard system includes a Jetson Orin NX (16GB), Logitech C922 Pro webcam, BNO055 IMU, and SparkFun 15136 GPS. Wireless communication is handled via 2.4 GHz Wi-Fi, enhanced by an access point.

2.4 Connection and Interaction of Components

Data flow in the system starts with the software-generated control commands sent via UART to the STM32F407 board. This board parses the data and relays it through I2C to the two STM32F103 boards, which apply RAKE's custom PID control algorithms to produce PWM signals for the motor drivers. This architecture enables precise wheel control.

Filtered encoder signals ensure accurate feedback, and the combined sensory input (camera, LiDAR, GPS, IMU) supports real-time localization and obstacle mapping. The hardware and software modules operate in close coordination, ensuring stable, responsive, and safe autonomous navigation.

3. Effective Innovations in Vehicle Design

3.1 Mechanical

Kayra utilizes a customized version of the classic rocker-bogie system, enhancing mobility on both flat and uneven terrains. By positioning the bogie system at the front and leaving the middle wheels unpowered, the number of control parameters is reduced, improving maneuverability. Box profiles were chosen for their higher strength and better weight optimization.

3.2 Electronic

The PID-controlled motor system maintains constant speed despite external factors like friction and slope, while analog-filtered encoders ensure signal accuracy. The e-stop system only cuts power to the motors, allowing the onboard systems to continue processing—an innovative safety approach that ensures uninterrupted autonomy.

3.3 Software

Our local grid-generation system is updated to produce a more accurate, real-time spatial representation of the robot's immediate surroundings to support path-planning algorithms. To achieve comprehensive environmental coverage, we employ a multi-camera setup consisting of two side-mounted cameras and one forward-facing camera, apply inverse perspective transform to get a birds-eye view, and employ cutting-edge vision processing methods to classify each cell as occupied or unoccupied. Enabling effective and low-cost mapping of the environment to guide the pat

4. Description of Mechanical Design

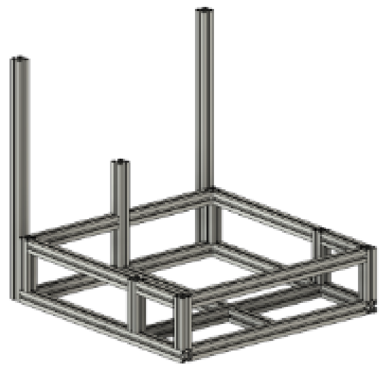
4.1 Overview

Developed by the ITU RAKE team, Kayra is an autonomous ground vehicle designed to be showcased at IGVC. It is specifically engineered to navigate challenging terrains safely and efficiently. With its revised rocker-bogie suspension system and skid-steering mechanism, Kayra offers high maneuverability and stability. Every aspect of its design—from material selection to mechanical structure—has been optimized for effective use in disaster zones and to ensure maximum durability.

4.2 Description of the Significant Mechanical Components

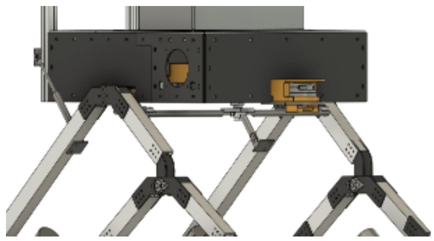
Kayra utilizes a six-wheeled system based on the rocker-bogie mechanism. On one side of the vehicle, the rocker section allows three wheels to move around the same axis, while the bogie section on the other side enables two wheels to rotate around their own axes. As a result, the vehicle employs a total of two rocker and bogie sections on each side. Unlike traditional rocker-bogie systems, the front of Kayra is formed by the bogie wheels, while the middle wheels are not powered by motors but are designed to rotate freely. This configuration reduces the number of parameters, making the vehicle easier to control and allowing it to navigate obstacles more effectively. For the steering mechanism, the vehicle adopts the skid-steering model, where the wheels on both sides rotate in opposite directions. This eliminates the need for additional maneuvering joints and reduces the number of motors, thus reducing both the complexity and weight of the system while preventing resistance issues in weak maneuvering axes.

4.3 Frame Structure, Housing, and Structure Design



The overall structure of Kayra has been designed to withstand the loads and stresses it will encounter during operation. The twisting stress generated by the skid-steering mechanism, as well as the bending moments at the rocker and bogie sections, have been carefully considered. To address these forces, box profiles were chosen for their high bending resistance, being 67% more resistant than tube profiles of similar dimensions. These profiles are made of aluminum, which helps to optimize the vehicle's weight, and the connecting parts are made of sheet steel, which provides the required strength.
The vehicle's chassis is constructed using sigma profiles,plexiglass sheets, and steel plates. The sigma profiles form the main load-bearing skeleton of the vehicle, while the plexiglass and steel components are used to protect the electronic systems and organize the internal layout. This combination provides mechanical strength while also offering protection against external elements, ensuring the vehicle's electronics are safe.

4.4 Suspension

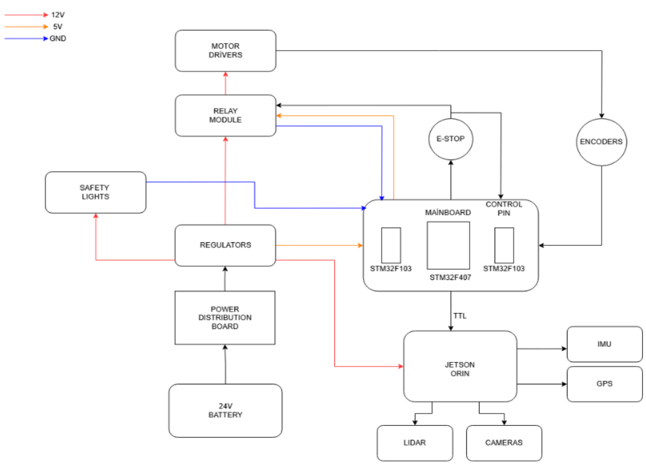


Kayra's rocker-bogie system uses rocker arms mounted on a fixed center, providing passive suspension. This ensures constant wheel-ground contact, enabling smooth obstacle traversal and load stability on rough terrain.

4.5 Weather Proofing

Kayra is specifically designed for use in disaster areas, meaning it must be capable of withstanding various environmental conditions. The use of aluminum box profiles, sheet steel connecting parts, and protective plexiglass helps ensure the vehicle is resistant to dust, water, and mechanical impact. The electronic systems are protected with sealed compartments that prevent damage from rain, mud, and other external factors. This ensures the vehicle will maintain its performance even during prolonged outdoor use.

5. Description of Electronic and Power Design



5.1 Overview

The Kayra vehicle can be controlled both autonomously and via remote control. The core components of the system manage the vehicle's motors and sensors, ensuring it moves safely and efficiently. The electronic system controls the speed and direction of the motors, enabling the vehicle to operate stably across various terrains and conditions.

5.2 Description of the Significant Power and Electronic Components

At the heart of the system are STM32F407 and STM32F103 microcontrollers. These microcontrollers control the vehicle's movement. The 24V 30,000 mAh Li-Po battery provides long-lasting power. BTS7960 motor drivers deliver power to the motors, enabling them to move. 600 pulse Rotary Encoders monitor the speed of the vehicle.

5.3 Power Distribution

The vehicle's power system is designed to ensure efficient operation of the motors and other components. The Li-Po battery allows for extended operation and quick charging. For safety, the emergency stop system cuts off power to the motors, stopping the vehicle. However, this system allows data processing to continue, ensuring the vehicle's functionality isn't compromised.

5.4 Electronics Suite

The Kayra's electronic package includes important components for autonomous operation. The Jetson Orin NX computer handles image processing and environmental sensing, helping the vehicle detect obstacles and navigate its surroundings. The STM32F407 and STM32F103 microcontrollers manage motor control, ensuring the vehicle moves at the desired speed and direction. The BTS7960 motor drivers control the motors' speed, and the 600 pulse Rotary Encoders measure the vehicle's speed with high precision. These components work in sync to ensure the vehicle operates smoothly, making Kayra fully controllable both autonomously and remotely.

5.5 E-stop Systems

The Kayra's safety is ensured by a specially designed e-stop system. This system can be activated either by the physical e-stop button on the vehicle or remotely via the remote control. When the e-stop is activated, the power to the motors is cut, stopping the vehicle. However, this does not affect the data processing on the mainboard or the image processing on the Jetson Orin NX. This feature allows the vehicle to stop safely while continuing other critical functions. Additionally, the wireless e-stop system enables remote activation, allowing the vehicle to be safely stopped when needed.

5.6 Sensor Suite

Side Cameras (Logitech C922 Pro Webcam): Used for lane detection, obstacle classification, and road surface segmentation via image processing. Semantic segmentation is applied using deep learning models, determining whether each pixel is drivable or not. This allows classification of elements like lanes, road edges, and obstacles.

Front Camera(Logitech Brio 270 Webcam): Used to cover blind-spots of the side cameras due robot chassis height. All the imagery is merged in the vision transformer unit to generate a complete local grid of the robot's surroundings.

IMU (BNO055): Measures the acceleration, tilt, and orientation of the vehicle to support motion estimation. These values are utilized in the localization process.

GPS (SparkFun ZED-F9P): Provides global position data, allowing the vehicle to determine its general location and navigate toward waypoints.

6. Description of Software System

6.1 Overview

The Software System of Kayra is built on ROS2, an industrial standard for building complex robotics systems. We prioritized modular design of specific functionalities and rigorous testing to ensure stable performance. Each component of the system is aware of the states of others to increase operational efficiency and dynamic decision-making.

6.2 Image Transformation & Grid-Mapping

Preprocessing

Two Logitech C922 cameras capture frames from the top-left and top-right sides of the robot, while a Logitech Brio 270 camera is mounted beneath the chassis; all cameras operate at 12 FPS. Each frame enters a vision-processing pipeline, where a box blur is first applied to reduce image noise. Then, the region of disinterest (RoD) is masked out for each camera to focus on relevant areas.

Vision Transformation & Grid generation

The resulting preprocessed image is passed to a UNet-based segmentation model, which detects road lanes, barrels, and pylons, and classifies each pixel as occupied or unoccupied. Following segmentation, each pixel is projected onto a bird's-eye view using Inverse Perspective Transform (IPT). The combined outputs are then converted into a 60×60 occupancy grid that spans 6 meters ahead and 3 meters to each side of the robot.

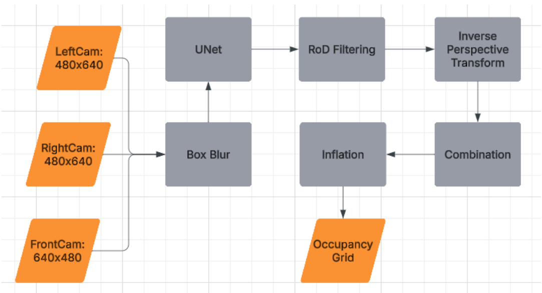


Figure 1: Vision Transformer Diagram

Inflation Layer

Finally, a gradient-based inflation layer is applied to the grid, expanding obstacle boundaries to support safer and more robust path planning.

6.3 Dual-Rate EKF Sensor Fusion

Kayra employs a Dual-Rate Extended Kalman Filter (EKF) architecture for robust, noise-resistant pose estimation. The Local EKF integrates high-frequency sensor data—specifically from the IMU and wheel odometry—to provide a continuous, real-time estimate of the robot's pose.

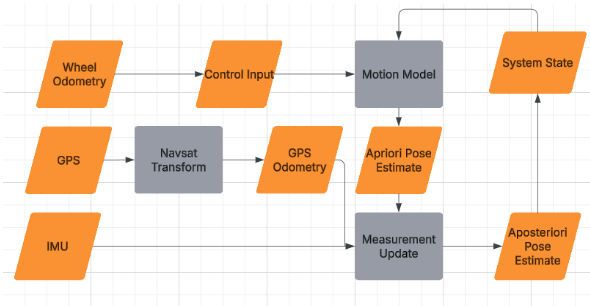


Figure 2: Dual-Rate EKF Sensor Fusion Diagram

In parallel, the Global EKF performs asynchronous corrections using low-frequency, absolute position updates from the GPS sensor. Both the process noise and measurement noise covariances are carefully tuned through simulation and validated in real-world tests to ensure stable and reliable localization. This dual-layered filtering approach allows Kayra to maintain accurate and continuous pose estimates, which are essential for downstream tasks such as path planning and dynamic goal selection.

6.4 Goal Selection & Path Planning

GPS Waypoint Following

The goal-selection system continuously generates and evaluates intermediate goals within the robot's local costmap to facilitate efficient navigation toward the final destination. GPS data is first transformed into the robot's local coordinate frame using the Navsat Transform package, ensuring consistent alignment with onboard localization.

Path Planning

A custom cost function is then applied to evaluate candidate goals. This function incorporates multiple factors: the heading difference between the intermediate goal and the next global waypoint, the angular deviation between the robot's current orientation and the candidate goal, and the Euclidean distance from the robot to the goal. Once the optimal intermediate goal is selected, an A* path-planning algorithm computes a collision-free path through the occupancy grid. The resulting trajectory is then smoothed into a series of linear segments to reduce path curvature and minimize travel time, enabling the robot to follow the path more efficiently and with greater stability.

6.5 Motion Generation and Monitoring

Once the path is divided into linear segments, the motion system ensures smooth following. If the heading error exceeds a threshold, a P controller adjusts orientation; otherwise, another P controller drives forward motion. On detecting obstacles, the robot first backs up and tries to replan. If unsuccessful, it rotates in place to find new paths. Global planning runs at 5 Hz, local planning at 10 Hz, ensuring responsive and accurate tracking in dynamic environments.

6.6 Robot Simulation

Leveraging competition footage and reference materials from previous years, we developed a detailed digital twin of the IGVC environment within the Gazebo simulator. This virtual replica was designed to mirror real-world conditions and allowed us to anticipate and address practical challenges before physical deployment.

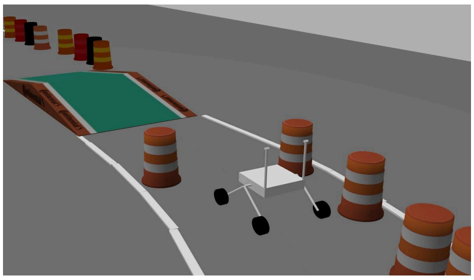


Figure 3: IGVC Course simulated in Gazebo

The simulated environment strictly follows the specifications outlined in the rulebook, including course dimensions, minimum path clearance, and obstacle constraints. To further enhance realism and robustness, we incorporated variable lighting conditions and introduced synthetic sensor noise, enabling us to evaluate the system's adaptability and noise tolerance. A simulated view of Kayra navigating this environment is shown in the image below.

6.7 Lifecycle Manager & System Modes

Environment & System Modes

Kayra's software architecture is built around a network of **Discrete Functional Units (DFUs)**, which communicate bidirectionally to enable modular development, rapid testing, and streamlined system design. Prior to system startup, an **environment mode** is selected—**simulation**, **training**, or **competition**—defining the overall context and purpose of the run. Simultaneously, the robot operates under a designated **operation mode—Autonomous**, **Manual**, **E-stop**, or **Shutdown**—which governs its internal control behavior. These two layers of state work in tandem to prepare the system for any navigation scenario with clarity and control.

Lifecycle Modes For DFUs

In addition to global system states, each software component (i.e., node) operates within its own **lifecycle state**, such as **unconfigured**, **configured**, **active**, or **aborted**, following ROS2 lifecycle management principles. For instance, before the robot initiates its course, the **vision transformer node** remains in the **configured** state.

Once the course begins, it transitions to the **active** state upon receiving a system-wide signal. In the event of communication loss or internal failure, the node moves to the **aborted** state and reports the error accordingly.

By clearly defining and exposing these states across the system, our team was able to manage complex behaviors more efficiently, reducing debugging time and improving the reliability of system interactions.

6.8 Additional Creative Concepts

Configuration & User Interface

To ensure optimal performance and optimally setting our robot for the competition, continuous system monitoring and seamless real-time configuration were essential. To address this need, we developed a web-based user interface that allows comprehensive monitoring of sensor data, real-time diagnostics of system nodes, and dynamic parameter tuning.

The interface is organized into multiple dedicated pages: a dashboard for visualizing ad-hoc system information, a vision page for inspecting vision processing modules, a configuration page for adjusting and testing system parameters in real time, and a logging page that displays categorized messages—including warnings, errors, and informational logs—from each active node. Implementing this interface early in development significantly accelerated our idea → design → implementation cycle, enabling rapid iteration and validation.

7. Cyber Security Analysis

7.1 NIST Risk Management Framework(RMF)

1. prepare: Establish the context, priorities, and scope of the risk management effort, including organizational risk tolerance and devising a strategy

2. categorize: Define the system's criticality and sensitivity based on potential impact levels (confidentiality, integrity, availability) following FIPS 199.

3. select: Choose baseline security controls from NIST SP 800-53 tailored to the system's categorization and specific requirements.

4. implement: Apply and document the chosen controls within the system architecture and operational environment.

5. assess: Evaluate the implemented controls to determine if they are correctly operating and producing the desired security outcomes.

6. authorize: Senior officials decide to accept risk and formally authorize the system to operate based on the assessment findings.

7. monitor: Continuously observe the system's security posture, reviewing control effectiveness and adapting to changes or emerging threats.

7.2 Model The Threats & Analyse Their Impact

Threat Description	Confidential	Availability	Integrity	Control Code
Physical Damage or Modification	low	medium	medium	PE-18
Access to runtime software	medium	high	medium	AC-3
Communication Interruption	medium	medium	medium	AC-4
Unauthorized Access	medium	high	medium	AC-17
Misuse of Privileges	high	high	medium	AU-2
Malware insertion via USB or debug port	high	high	high	CM-7
Configuration change without authorization	low	medium	high	CM-3, AC-3
Sensor spoofing	low	medium	high	SI-4

7.3 Cyber Controls & Description of Their Implementations

Threat	Control Code	Applicable Control
Misuse of Privileges	AU-2	All the events in Jetson are monitored and logged into a log file, any critical events are priority events will be monitored by the admins periodically
Communication Interruption	AC-4	Use message filters or access control on communication interfaces (e.g., restrict which IPs/ports can send commands to the robot).
Malware insertion via USB/debug	CM-7	System components are reviewed periodically, unused ports are disabled and each executable is registered with a GPG key to prevent malware insertion
Unauthorized Access	AC-17	firewall is configured to only allow data sent from robot static IP, wifi network is configured for MAC-filtering,
Physical Damage or Modification	PE-18	Robot is securely stored in supervised areas, in case of any physical damage, source code is kept and updated on github
Configuration change without auth	CM-3	Use git for version control, restrict write permissions on important config files, only allow authorized users to apply changes to software
Configuration change without auth	AC-3	In competition environment, software is only allowed to be uploaded to the robot only via ssh with a physical ethernet connection
Sensor spoofing	SI-4	Jumps on GPS sensor data is checked and only sensor data under a threshold is accepted, unexpected behaviour is logged immediately

8. Analysis of Complete Vehicle

8.1 Lessons Learned During Construction and System Integration

- The revised rocker-bogie system improved mobility over obstacles, and switching to skid-steering reduced motor usage and simplified control. Encoder signal issues during integration were mitigated by analog filters. PID control enhanced stability against surface variations.
- One of the most valuable lessons learned during the development process was the importance of effective communication between teams, particularly among team captains. Misunderstandings, even minor ones, can unexpectedly hinder progress. Holding brief but regular face-to-face meetings proved highly effective in keeping our design and implementation phases on track this year.

8.2 Top Hardware Failures That Would Prevent Competition Success and Mitigations

- Potential failures include encoder noise and motor power loss. These were addressed with signal filtering and a selective e-stop system that only cuts motor power, allowing the rest of the system to continue operating. STM32 boards are modularly distributed for resilience.

- High power load on the Jetson Orin NX caused by the RGB cameras was resolved by using a powered USB hub, which both offloaded the USB ports and supplied adequate power to the cameras.
- To address overheating issues in the microcontrollers, voltage regulators, and onboard computer, heat-sensitive components were strategically positioned near cooling fans during the vehicle’s design process.
- To prevent overcurrent and overvoltage on sensitive components, the robot's battery is equipped with a Battery Management System (BMS) that limits current draw. All components are powered through voltage regulators with built-in current and voltage protection. Spare regulators are kept on hand to quickly replace any that overheat during operation.

8.3 Safety, Reliability, and Durability Considerations and Design Problems Encountered

- To ensure an extensive and reliable emergency stop (E-stop) range, an HC-12 module was used with RF communication, providing up to 1000 meters of range in outdoor environments.
- Aluminum box profiles provided strength with low weight, while steel connectors reinforced load-bearing areas. Torsional stress during turns was handled by using box-profile materials for improved rigidity.
- To prevent memory overload in the global costmap, a rolling window costmap was implemented. This approach reduces memory usage by dynamically allocating resources only to the relevant area around the robot while retaining essential navigation information.
- To meet the demands of real-time autonomous navigation, a lightweight perception stack was developed using RGB cameras combined with efficient, state-of-the-art AI models. This setup enables fast and resource-efficient grid mapping without the overhead of heavier sensor suites, ensuring responsiveness within the system’s real-time constraints
- To sustain an evolving and complex robotic software architecture, the system was modularized into Discrete Functional Units (DFUs). A dedicated lifecycle manager and state machine framework were implemented to ensure intuitive system design, streamline development, and enable robust coordination between modules

8.4 Key Hardware Failure Points and Failure Modes and On-Site Resolution Strategies

- In case motors fail during testing phases, replacement motors, encoders, and mounting equipment are kept on hand for quick repairs.
- To ensure connection reliability, all robot wiring is soldered with clip-on connectors used; necessary electronic components for checking, wiring, and soldering are available if disconnections occur during the competition trials.
- Component protection against electrical overload is provided by the robot's battery management system (BMS) which restricts current flow, while voltage regulators with built-in limits safeguard all components; replacement regulators are available if overheating occurs.

8.5 Virtual Environment Testing & Gazebo Simulation

We employed Gazebo as our physical simulation platform, allowing continuous testing under diverse environmental conditions including variable lighting and world configurations. We developed a digital twin replicating prior competition environments, conforming to specified course dimensions, minimum path clearance requirements, and other constraints outlined in the rulebook.

Our simulation framework supports testing object detection capabilities and can incorporate simulated sensor noise to enhance sensor fusion robustness. This tool enables us to calibrate and optimize the robot for virtually any scenario, facilitating comprehensive system testing at any time

8.6 Physical Testing to Date

Software: Our tests focused on verifying sensor calibration, accurate sensor fusion, environment-agnostic occupancy grid generation and effective path planning. Camera and IMU calibration took more time than estimated. UNet model performed semi-optimally on lower lighting conditions. In autonomous navigation, we observed

oscillations in the path-planner in some of the rides. Our team focused on resolving these issues and pruning the navigation stack for the last month before competition.

Electronics: During physical testing, the electronics team primarily focused on two critical aspects: ensuring sufficient power delivery to the motors and maintaining reliable data communication between subsystems.Initially, the motors were not receiving the expected amount of power under load. Upon investigation, this issue was traced back to the current limiting configuration on the buck converters. The current thresholds were accordingly adjusted to allow adequate current flow, resolving the power delivery issue without requiring hardware replacement.Additionally, intermittent communication losses were observed between the master controller and peripheral boards. To address this, the software architecture on the master microcontroller was modified to prioritize communication management tasks in a way that maintains system integrity, ensuring robust and consistent data exchange throughout operation.Based on these findings, our team successfully implemented and validated the corresponding solutions during the final rounds of testing.

Mechanical: In the initial tests for Kayra, a 6-wheeled rocker bogie system was used with only 4 wheels (excluding the middle ones) powered by motor-gearbox units delivering a nominal torque of 3.8 Nm. The system was limited to a maximum current of 40 amps. Under these conditions, Kayra successfully demonstrated straight-line motion and moderate slope climbing but failed in turning maneuvers.
Following these results, the torque output of the powered wheels was increased to 9.54 Nm. With this configuration, the vehicle performed better in turning and obstacle negotiation on a PVC surface. Later, all six wheels were powered, further improving turning, obstacle traversal, and steep slope climbing capabilities.
Subsequent tests were conducted on asphalt, where Kayra maintained good forward motion and slope climbing but again showed poor turning performance. To address this, the system was stiffened. During this phase, the existing battery was found insufficient, prompting a switch to a new one with an 80-amp current limit.
Although the new battery and stiffened structure led to some improvements in asphalt turning, full success was not achieved. Further analysis revealed that the current to the motors was still being limited. After increasing this current limit, Kayra successfully executed ideal tank turns on PVC and completed turning tests on asphalt as well, concluding the mechanical testing phase.

8.7 Predicted Performance Analysis

- **Speed**: Estimated at 1–1.5 m/s.
- **Ramp Climbing**: Capable of handling 20–30° slopes.
- **Reaction Time**: Expected to be 200–300 ms.
- **Localization**: EKF with GPS, IMU, odometry provides ~0.3–0.7 m accuracy.
- **Battery Life**: 2–3 hours with 24V 30000mAh Li-Po.
- **Obstacle Detection**:Up to 6 meters with Multi-camera Perception Module
- **Complex Obstacles**: Handled via UNet Segmentation Model
- **Waypoint Accuracy**: 0.25 meter tolerance
- **Failure Handling**: Modular design, recovery behaviour and e-stop mechanism provide quick recovery.
- **Testing Comparison**: Some encoder noise and mechanical stress observed; mitigated via filtering and material improvements.

8.8 Software Testing Bug Tracking & Version Control

For version control and remote storage, our team implements git and GitHub. The git platform provides pull requests, branches, and issues sections, facilitating simultaneous development across multiple developers. GitHub serves as our remote repository host, storing both legacy and current software versions accessible to all team members. Upon initialization, Kayra automatically retrieves code from the main branch's latest version—a branch that can only be modified through pull requests that undergo review by authorized team leaders before implementation.

9. Unique Software, Sensors and Controls Tailored for AutoNav

9.1 Overview

To enable robust and real-time autonomous navigation, *Kayra* is equipped with a series of custom-designed subsystems that collectively ensure accurate perception, reliable localization, intelligent goal selection, and precise motion control. Each component was engineered to address the unique challenges posed by the IGVC AutoNav competition, such as limited compute budgets, dynamic environments, and the need for high-frequency decision-making.

9.2 Vision-Based Occupancy Grid Generation

Kayra employs a lightweight yet powerful grid-mapping approach using a tri-camera setup—two Logitech C922s positioned on the top-left and top-right of the chassis, and a Logitech Brio 270 camera mounted underneath the robot. These cameras stream synchronized RGB frames at 12 fps, which are passed through a vision processing pipeline tailored for real-time operation.

The pipeline begins with noise-reduction via box blur and filters out regions of disinterest before the images are processed by a UNet-based segmentation network. This model identifies lane markings, pylons, and barrels and classifies each pixel as occupied or free. Inverse Perspective Transform (IPT) then maps image pixels to world coordinates, which are fused into a 60×60 local occupancy grid covering 6 meters in front and 3 meters laterally. A gradient-based inflation layer is applied to expand obstacle zones, helping the path-planner generate safer, smoother trajectories.

9.3 Robust Sensor Fusion with Dual-Rate EKF

For accurate pose estimation, Kayra utilizes a Dual-Rate Extended Kalman Filter (EKF) architecture, explicitly tailored for noise-immune sensor fusion in real-world outdoor settings. The **local EKF** runs at a high frequency to integrate data from the IMU and wheel odometry, ensuring continuous updates to the robot's pose even in GPS-denied zones.

Meanwhile, the **global EKF** corrects long-term drift using low-frequency but globally referenced GPS measurements. Both process and measurement noise matrices were finely tuned in simulation and real-world tests to guarantee consistent, reliable localization across different terrains and operating conditions. This robust localization backbone enables the planner and control layers to function with high precision.

9.4 Autonomous Goal Selection and Adaptive Path Planning

The goal selection module continuously generates intermediate waypoints within the robot's local costmap, optimizing for smooth progression toward the final destination. By transforming GPS-based waypoints into the robot's local frame using navsat_transform, the system evaluates multiple candidate goals through a cost function incorporating factors like heading alignment, angular deviation, and Euclidean distance.

Once an optimal goal is selected, the A* algorithm generates a path which is later smoothed into linear segments to reduce trajectory complexity and improve velocity tracking. This dynamic goal selection and path refinement ensure the robot adapts fluidly to changing environments and obstacle configurations.

9.5 Modular Control Architecture for Real-Time Motion Execution

Kayra's motion generation system is designed for high-frequency path tracking with built-in recovery mechanisms. For each path segment, the robot generates angular velocity commands to align itself with desired orientation using a P controller, if angular difference is smaller than a threshold, it adjusts its speed with another p controller for smooth tracking.

When unexpected obstacles are detected, the robot executes a hierarchical recovery behavior: first attempting a backup routine to re-plan a valid path, followed by a rotation recovery if needed. The global planner runs at 5 Hz, while the local planner operates at 20 Hz, enabling responsive, low-latency adjustments to the robot's trajectory. These control layers are tightly integrated with the central microcontroller via serial communication, which executes per-wheel RPM commands using a PID loop, ensuring smooth and accurate motion.

9.6 Integrated Web-Based Interface for Realtime Monitoring and Configuration

To facilitate system reliability and ease of debugging during both development and competition phases, Kayra includes a custom web-based UI. This interface allows real-time monitoring of sensor streams, component states, and system logs, as well as live configuration of parameters such as control gains and vision thresholds. Separate tabs for data inspection, vision output, parameter tuning, and node logging helped significantly shorten the idea-design-implementation cycle and enabled rapid testing and troubleshooting during field deployments.

10. Initial Performance Assessments

Avg. Run Time	4 mins 55 seconds
Average Speed	4 mins 35 seconds
Average Speed	0.7 m/s
Max Speed	1.1 m/s
Battery Life	180 mins
Waypoint Accuracy	0.1 meter tolerance