



Conceptual Design Report

Version 1.0
29.12.2024

Design Studio Section:

6

Presented by:

Murathan Kutaniş
Alperen Şahin
Enescan Çelebi
Hacer Ayça Yılmaz
Sümeyra Arıcan
Öykü Özyurt



PROPRIETARY STATEMENT

This document contains proprietary information of OSEAM AG and, except with previous written authorization of OSEAM AG, such information shall not be published or disclosed to others, and the document shall not be duplicated in whole or in part. The proprietary information disclosed herein may be used by the Addressee solely to evaluate the application. If the document is submitted as part of an application to be submitted by the Addressee to another Customer, OSEAM AG hereby authorizes the Addressee to transfer the data in their proposal with the same reservations. If the licence is awarded to this application or as result of or in connection with the submission of this document, the Addressee shall have the right to use or disclose such information or duplicate this document to the extent provided in licence grant.

The above statement is not applicable if the proprietary information disclosed herewith have already been made generally available to and/or known by the public and/or is already known by or in possession of the Addressee or is thereafter rightfully received or becomes available from a third party with no secrecy obligations thereto.

Company Information

| | |
|---|---|
| Economic Operator Complete Name, Legal Nature and Address | OSEAM AG Middle East Technical University Üniversiteler Mahallesi, Dumlupınar Bulvarı No:1 06800 Çankaya Ankara/TÜRKİYE |
| Headquarters | Ankara |
| Place of Jurisdiction | Local Court Ankara |
| Legal representatives | |
| Contact | oseam2024@gmail.com |

Table 1-1: Company Information

Project Information

| | |
|-----------|----------------------------------|
| Addressee | Middle East Technical University |
| Title | Business State Report |
| File ID | BUS-OSEAM-0001 |

Table 1-2: Project Information

Signatures

Approved by Approved by Approved by Approved by Approved by Approved by

Table 1-3: Signatures

| Alperen Şahin | Enescan Çelebi | Hacer Ayça Yılmaz | Murathan Kutaniş | Öykü Özyurt | Sümeyra Arıcan |
|---------------|-------------------|----------------------|---------------------|-------------|-------------------|
| | | | | | |

Contents

| | |
|---|----|
| Executive Summary | 3 |
| Introduction | 4 |
| - Literature Review | 4 |
| - Motivation of the Project | 5 |
| - Current Status of the Project | 6 |
| - Scope | 6 |
| - Organization | 6 |
| Problem Statement | 7 |
| Objectives And Requirements | 7 |
| - Objectives | 7 |
| - Functional Requirements | 8 |
| System Design | 9 |
| Relevant Standards and Compliance: | 12 |
| 3D Drawing of the System: | 13 |
| Sub-Systems: | 15 |
| - Standards Compliance | 16 |
| 1. Inertial Sensor Suite | 16 |
| 2. Camera | 25 |
| Plan B for Subsystems | 26 |
| Test Results | 27 |
| - IMU Translational Accuracy Tests | 36 |
| - Rotational Accuracy Tests | 37 |
| - Combined Accuracy Tests | 38 |
| - IMU Test Results Discussion and Evaluation | 39 |
| - Camera Translational Accuracy Tests | 40 |
| - Camera Rotational Accuracy Tests | 41 |
| - Camera Test Results Discussion and Evaluation | 42 |
| Planning | 43 |
| - Schedule | 43 |
| - Ordering the Supplementary Components | 43 |
| - Development of the Algorithms | 43 |
| - Simulations and Tests of the Subsystems & Overall System | 44 |

| | |
|---|----|
| - Hardware Setup & Final Improvements | 44 |
| - System Integration & Complete Design Tests | 44 |
| - CDR Report - Presentations | 44 |
| - Project Presentation Setup | 44 |
| - Risk Analysis | 44 |
| - FMECA Methodology | 44 |
| - Sensor Fuse Algorithms | 57 |
| - Power Systems | 57 |
| - Communication | 59 |
| - Adding A Lidar Sensor | 59 |
| - Camera | 60 |
| - OpenCV SFM Module | 60 |
| - Harness | 61 |
| - Deliverables and Cost Analysis | 63 |
| - Ethical Consideration in System Design | 64 |
| Test Plans | 64 |
| - Component Tests | 64 |
| - Subsystem Tests | 66 |
| Conclusion | 69 |
| REFERENCES | 70 |
| APPENDIX | 71 |

Executive Summary

This report details the development of the "Inside-Out Tracking Sensor Suite," an advanced system designed to address the challenges of real-time head pose estimation for virtual reality (VR) applications. By leveraging sensor fusion of an Inertial Measurement Unit (IMU) and a camera, the system ensures accurate six-degree-of-freedom (6-DOF) tracking with low latency, offering users an immersive and flexible VR experience. The system is designed to eliminate the constraints of traditional tracking methods that rely on external sensors or fixed play zones by utilizing a compact, self-contained design.

The visual tracking component of the system is powered by a custom visual odometry algorithm that enables precise pose estimation. The algorithm uses Shi-Tomasi corner detection to identify key features in image frames, which are then tracked using the Lucas-Kanade sparse optical flow method. To estimate relative camera motion, the essential matrix is computed using a 5-point solution, with RANSAC employed to remove outliers. Pose recovery is achieved through Singular Value Decomposition (SVD), ensuring

accurate rotational and translational estimates that are further validated using chirality tests. By dynamically updating keyframes, the algorithm adapts to changing environments, maintaining robustness even in low-feature or dynamic settings.

Complementing the visual tracking, the IMU subsystem integrates data from the MPU6050 IMU and GY-271 magnetometer, processed through advanced algorithms such as the Extended Kalman Filter (EKF) and Madgwick Filter. These algorithms mitigate drift and noise, providing high-frequency updates for stable and reliable orientation and position tracking. Initial system tests have demonstrated strong performance in rotational accuracy and most translational scenarios, with plans to refine performance in challenging environments.

Future work focuses on the integration and testing of the subsystems under dynamic and diverse conditions, improving environmental adaptability and overall system robustness. Ergonomic improvements and user-friendly calibration methods will ensure a seamless user experience. The project is committed to maintaining cost-effectiveness, with the system design constrained to a \$300 budget. A detailed risk analysis ensures the system adheres to safety and performance standards, reducing potential failure modes to an As Low as Reasonably Practicable (ALARP) level.

The "Inside-Out Tracking Sensor Suite" represents a significant advancement in VR technologies, combining innovative algorithms and efficient hardware integration to deliver a high-performance, compact, and accessible tracking system. This project sets a strong foundation for future innovations in immersive and interactive technologies.

Introduction

- Literature Review

Estimation of the head pose in virtual reality (VR) applications is critical to creating immersive and seamless user experiences. Accurate pose estimation ensures precise alignment between physical movements and the virtual environment, minimizing discomfort and enhancing usability. Early systems relied heavily on external tracking solutions, such as infrared markers and external cameras, which required fixed setups and limited user mobility. Examples include the OptiTrack system and Vicon cameras, which provided high precision but constrained user interaction to specific zones.

Recent advancements have shifted focus to inside-out tracking methods, leveraging onboard sensors like inertial measurement units (IMUs) and cameras. Research by Scaramuzza et al. (2011) demonstrated the use of visual odometry for real-time camera pose estimation in robotics, which later influenced VR applications. Similarly, Newcombe et al. (2011) introduced KinectFusion, which reconstructed 3D environments using RGB-D cameras, showcasing the potential for spatial understanding in VR tracking systems.

Sensor fusion techniques have also gained prominence in overcoming challenges like drift and noise. Studies by Madgwick et al. (2011) proposed a computationally efficient filter for orientation estimation, combining gyroscope, accelerometer, and magnetometer data. Another widely referenced work is the use of the Extended Kalman

Filter (EKF) for fusing inertial and visual data, as described by Mourikis and Roumeliotis (2007), which set the foundation for robust 6-DOF pose estimation.

Visual odometry algorithms have advanced significantly in the VR domain. For example, Klein and Murray (2007) introduced Parallel Tracking and Mapping (PTAM), which performs simultaneous localization and mapping (SLAM) for real-time camera tracking. The integration of optical flow methods, such as the Lucas-Kanade algorithm, further improves feature tracking and motion estimation, as demonstrated in subsequent works.

These literature examples highlight the evolution of pose estimation technologies, moving toward compact, accurate, and user-friendly solutions. The insights gained from these studies form the basis for developing modern VR tracking systems like the Inside-Out Tracking Sensor Suite, which combines advanced visual and inertial data processing to achieve high performance in diverse environments.

- Market Review

The market for VR systems has experienced exponential growth, driven by advancements in hardware and software technologies and an expanding range of applications beyond gaming, including education, healthcare, and industrial training. Inside-out tracking systems, which eliminate the need for external sensors, are becoming increasingly popular due to their ease of setup and enhanced user mobility. Industry leaders such as Oculus (Meta), HTC, and Sony have integrated such tracking technologies into their latest VR headsets, setting new benchmarks for performance and affordability.

A key market trend is the increasing demand for lightweight, ergonomic VR headsets with high accuracy in head pose estimation. Consumers prioritize systems that offer low latency, wide compatibility, and seamless integration with existing platforms. Cost considerations remain a critical factor, especially in emerging markets. While high-end systems dominate the professional and enterprise segments, budget-friendly alternatives are gaining traction among casual users and educational institutions.

The competitive landscape underscores the importance of innovations that balance performance and affordability. Systems leveraging sensor fusion and advanced algorithms, like the Inside-Out Tracking Sensor Suite, are well-positioned to address these market demands. By offering a compact and adaptable solution with a target cost of \$300, the proposed system aligns with current market trends and has the potential to capture a significant share in both consumer and enterprise sectors.

- Motivation of the Project

The motivation behind this project is to enhance the user experience in virtual reality by eliminating the limitations of traditional tracking systems that rely on external sensors or dedicated play spaces. Inside-out tracking provides users with greater flexibility and freedom to move naturally without being confined to specific zones or setups. By integrating a compact, self-contained tracking system within the VR headset, we aim to create a more immersive and realistic experience that allows users to interact fluidly with virtual environments. This project seeks to address the challenges of real-time head pose

tracking in VR applications by leveraging advancements in sensor fusion technology to improve accuracy and responsiveness.

- Current Status of the Project

At this stage of the project, we have identified the fundamental algorithms of our two main components: camera and IMU. We have defined the scope, objectives and also the supplementary components of the system. We have finalized the core subsystems and planned the future course of events.

Based on the results of the tests, we have determined optical configurations and solutions for the subsystems. We prioritize real-time processing, and data accuracy; thus, alternative improvements will yet to be made.

Currently, we are refining our main solution by integrating additional components and selecting the most suitable approach for the sensors separately and altogether. Following this, we will mostly focus on communication channel development, and sensor fusion algorithm that will enable us to achieve an accurate and real-time camera pose estimation.

In the next steps, we will carry out subsystem and overall system tests and simulations rather than component-wise approaches. This includes testing the integration of the systems and testing under environmental challenges. Later, we will continue to optimize the functionality, accuracy, and robustness of our system.

- Scope

This report focuses on the development and integration of the Inside-Out Tracking Sensor Suite for virtual reality applications. The primary goal is to achieve accurate, real-time head pose estimation using sensor fusion technology, combining data from IMUs, cameras, and additional sensors such as a magnetometer and LiDAR. The system addresses technical challenges, including synchronization, drift correction, environmental adaptability, and cost-effectiveness, while ensuring compliance with ergonomic and electronic standards. The report provides a comprehensive analysis of the project's objectives, system design, testing results, and risk mitigation strategies, ensuring the solution meets both performance and usability requirements.

- Organization

The report is structured as follows:

1. **Introduction:** Provides an overview of the project's motivation, problem statement, and objectives, highlighting the need for enhanced tracking solutions in VR applications.
2. **System Design:** Details the architecture and functionality of the system, including the integration of IMU, camera, and additional sensors like LiDAR and magnetometer.
3. **Sub-System Descriptions:** Discusses each subsystem in depth, including the inertial and visual sensor suites, sensor fusion algorithms, and their implementation.
4. **Test Results:** Presents the results of individual and integrated system tests, analyzing performance, accuracy, and compliance with predefined requirements.

5. **Cost Analysis:** Provides an overview of the materials used, their procurement timeline, and a detailed cost breakdown.
6. **Planning and Schedule:** Outlines the project timeline, milestones, and responsibilities for each task.
7. **Risk Analysis:** Identifies potential failure modes and risks, with proposed mitigation measures to ensure system reliability and performance.
8. **Conclusion:** Summarizes the findings, design decisions, and future steps to optimize and finalize the system.

Problem Statement

The development of the Inside-Out Tracking Sensor Suite for virtual reality applications poses several critical engineering challenges. From a technical perspective, the primary problem lies in achieving accurate, real-time head pose estimation using a combination of IMU sensors, cameras, and data fusion algorithms. Ensuring low-latency, six-degree-of-freedom (6-DOF) tracking requires precise synchronization and integration of asynchronous sensor data, often prone to noise, drift, and environmental variability. Additionally, designing a compact, ergonomic, and lightweight wearable system that meets stringent power and thermal constraints presents further challenges in balancing performance and user comfort. Adapting the system to operate reliably under diverse environmental conditions, such as low-light or featureless spaces, while maintaining affordability within a \$300 budget, exacerbates the complexity. These technical problems demand innovative approaches in hardware selection, algorithm optimization, and system design to ensure the product delivers accurate, seamless, and immersive VR experiences without compromising usability or cost-effectiveness.

Objectives And Requirements

- **Objectives**
 - **Real-Time Tracking**
Implement real-time six-degree-of-freedom (6-DOF) head pose estimation, integrating rotational and translational parameters to deliver immediate responsiveness.
Justification: Real-time tracking is crucial for maintaining immersion and ensuring the virtual environment responds seamlessly to user movements.
 - **High Accuracy**
Ensure precise head pose estimation by fusing data from the camera and IMU, achieving sensitivity levels of $\pm 0.01^\circ/\text{s}$ for the gyroscope and $\pm 0.01 \text{ m/s}^2$ for the accelerometer.
Justification: Accurate measurements eliminate errors, reduce drift, and enhance the realism of VR interactions.
 - **Compact Wearable Design**
Develop a lightweight, ergonomic head-mounted sensor suite for extended wearability and comfort.
Justification: A compact design ensures ease of use and makes the system suitable for wearable applications.
 - **Energy Efficiency**
Optimize power consumption to $\leq 12\text{W}$ for all components to support portable

and battery-operated use.

Justification: Minimizing energy usage extends operational time and ensures thermal safety.

- **Environmental Adaptability**

Achieve reliable operation across varying indoor environments, including low-light conditions (50 lux) and dynamically changing lighting scenarios.

Justification: Adaptability ensures robust performance in diverse settings, enhancing usability.

- **Cost Efficiency**

Maintain a cost-effective design, limiting production expenses to \$300 or less without compromising quality.

Justification: Cost efficiency broadens accessibility and market appeal.

- **Ease of Use**

Provide an intuitive setup process and maintain reliable functionality within a minimum operational space of 2m x 2m.

Justification: User-friendly design promotes adoption and reduces setup complexity.

- **Modularity and Scalability**

Design the system with modular components to facilitate optional features like augmented reality or gaze tracking.

Justification: Modularity supports future upgrades and tailored applications.

- **Functional Requirements**

- **Continuous Image Processing**

Perform near real-time processing on continuous image streams captured by the camera for timely updates.

- **IMU Measurement Acquisition**

Acquire angular position, velocity, and acceleration from the IMU to track dynamic movements effectively.

- **Sensor Fusion**

Fuse data from the camera and IMU to enhance responsiveness and reliability of head pose estimation.

- **Scene Analysis and Collision Warning**

Analyze the scene to issue collision warnings, ensuring safety and situational awareness.

- **Camera View Transmission**

Transmit a first-person camera perspective to an external computer for centralized processing.

- **Head Model Rendering**

Render the acquired sensor data on a head model to visually reflect user movements.

- **Augmented Reality Integration**

Seamlessly render the user's view locally for augmented reality applications.

- **System Requirements**

1. **Camera**

1. Resolution: 720p or 1080p
2. Frame Rate: >30 FPS
3. Working Distance: >2 meters
4. Field of View (FOV): 70°–90° horizontal, 60°–80° vertical

2. **IMU Specifications**

1. Gyroscope Sensitivity: ±0.01°/s or better
2. Accelerometer Sensitivity: ±0.01 m/s² or better

3. Sampling Rate: ≥ 200 Hz

3. Performance Requirements

1. Data Fusion Latency: ≤ 10 ms per sensor update cycle
2. Tracking Frequency: ≥ 60 Hz
3. Lighting Tolerance: 50 lux (low light) to 100,000 lux (sunlight)

4. Power Consumption

1. ≤ 12 W for the entire system

5. Physical Requirements

1. Lightweight and ergonomic wearable form factor
2. Wired communication between sensors and the processing unit

System Design

The system is built around the Raspberry Pi 5, integrating an Inertial Sensor Suite and a Visual Sensor Suite to achieve precise motion tracking and pose estimation. The Inertial Sensor Suite includes the MPU6050 and GY-271 sensors, which provide essential data for understanding motion and orientation. Rotation is determined using quaternion orientation derived from the gyroscope and accelerometer data, while translation is estimated through an Extended Kalman Filter. This approach helps mitigate errors such as drift, ensuring more reliable measurements over time.

The Visual Sensor Suite employs the Raspberry Pi Camera Module 3, which captures video and processes it using optical flow techniques. Optical flow is used to track changes between consecutive frames, enabling the calculation of both rotational and translational movements. The system dynamically updates keyframes to ensure robust performance even in changing environments.

By combining inertial data with visual odometry, the system achieves a balance between high-frequency updates from the sensors and drift-free corrections from the camera. This integration ensures accurate and stable pose estimation, making it suitable for applications that demand reliable motion tracking and navigation in dynamic scenarios. The fusion of these outputs will enhance performance while maintaining flexibility across various conditions.

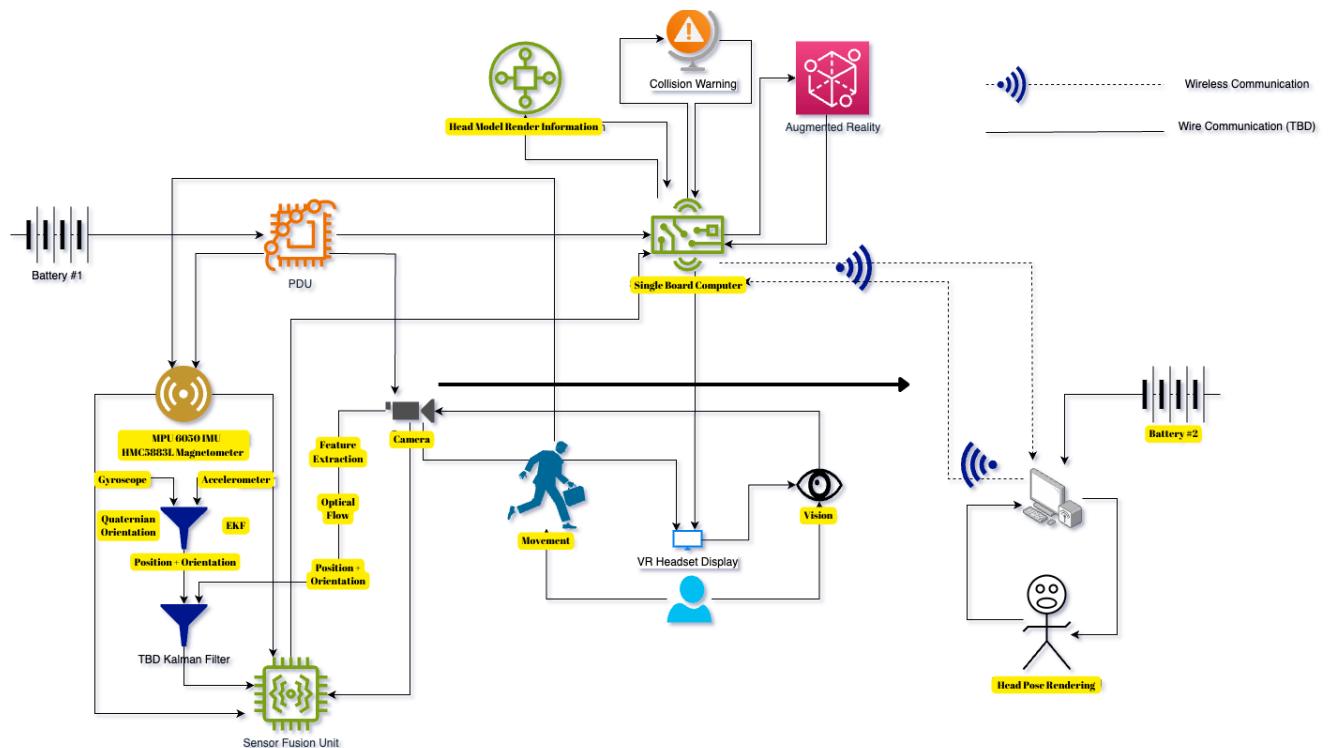


Figure 1: Main Design - Processing Parts are Highlighted

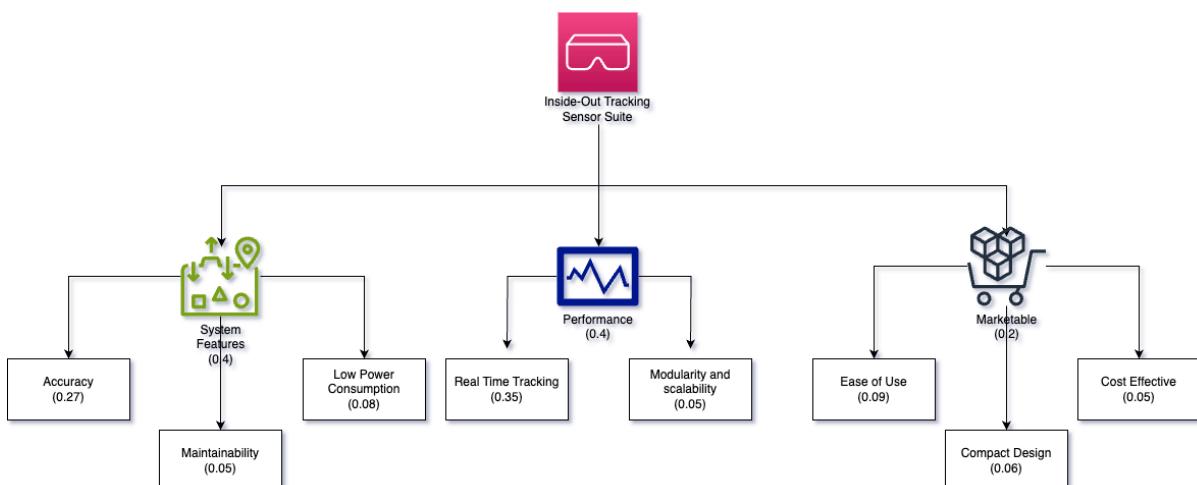


Figure 2: Weighted Objective Tree

The proposed solution for the Inside-Out Tracking Sensor Suite aligns well with the weighted objective tree by addressing each critical category—**System Features**, **Performance**, and **Marketability**—as follows:

System Features (0.4)

1. Accuracy (0.27):

- o The system integrates the **MPU6050 IMU** and **HMC5883L Magnetometer** for precise rotational and translational motion tracking. Sensor fusion algorithms like the **Madgwick Filter** enhance accuracy by mitigating sensor noise and drift.

- The use of a **LiDAR sensor** will ensure improved depth perception and spatial accuracy, addressing environmental variability and featureless spaces.

2. Low Power Consumption (0.08):

- The system is built on a power-efficient platform using the **Raspberry Pi 5** and optimized algorithms like the Madgwick Filter, which reduce computational demands without compromising performance.

3. Maintainability (0.05):

- Modular subsystems (e.g., IMU, camera, and LiDAR) simplify troubleshooting and allow individual component upgrades or replacements, ensuring long-term maintainability.

Performance (0.4)

1. Real-Time Tracking (0.35):

- The solution prioritizes real-time processing with the integration of an **Extended Kalman Filter (EKF)** and high-frequency IMU outputs. Testing has shown minimal latency in tracking head poses, enabling seamless virtual reality experiences.
- The camera's optical flow and visual odometry algorithms ensure real-time pose estimation, even in dynamic environments.

2. Modularity and Scalability (0.05):

- The system's modular design supports future enhancements, such as adding higher-resolution cameras or more advanced sensors. Scalability is ensured by the compatibility of the Raspberry Pi 5 with a variety of peripherals and software frameworks.

Marketability (0.2)

1. Ease of Use (0.09):

- The compact and ergonomic design ensures user comfort. Clear calibration protocols, such as automated sensor calibration and user-friendly interfaces, simplify setup and operation.
- The solution includes visual and auditory feedback mechanisms to guide users during setup and usage.

2. Compact Design (0.06):

- All components, including the IMU, magnetometer, and LiDAR, are integrated into a lightweight and compact package. The system is designed to be wearable, aligning with market expectations for portability.

3. Cost Effectiveness (0.05):

- By selecting cost-efficient components like the MPU6050 and leveraging bulk purchasing strategies, the system remains within the target budget of \$300 while delivering high performance.

Table 1: Objectives

| Objective | Functional requirement | Performance requirement |
|------------------------------|--|--------------------------------------|
| Real-time Tracking | Sensor Fusion | Data Fusion and Processing |
| | Continuous Image Processing | |
| Accuracy | IMU Measurement Acquirement | IMU Accuracy and Sensitivity |
| | Sensor Fusion | Head Pose Rendering |
| | Six Degree of Freedom Head Pose Estimation | |
| Low Power Consumption | Power Consumption for All Components | |
| Indoor Compatibility | Environment and Lighting Adaptability | |
| Cost Efficiency | Continuous Image Processing | IMU Accuracy and Sensitivity |
| | IMU Measurement Acquirement | Power Consumption for All Components |
| Ease of Use | Transmitting camera view to computer | Head Pose Rendering |
| | Head Model Rendering | |
| Maintainability | Power Consumption for All Components | |
| | Environment and Lighting Adaptability | |

Relevant Standards and Compliance:

To ensure the Inside-Out Tracking Sensor Suite is marketable, it is designed to comply with industry-relevant standards, addressing safety, performance, and usability requirements. Key standards include:

1. Ergonomic Standards (ISO 9241):

- Compliance ensures user comfort and safety during extended use, particularly for wearable systems. The design considers weight distribution, adjustability, and ease of wear to meet these requirements.

2. Electronic Safety Standards (FCC and CE Certifications):

- The product adheres to regulations for electromagnetic compatibility and electronic device safety, ensuring it can be legally distributed in various markets. Compliance with these certifications prevents interference with other devices and guarantees safe operation.

3. Environmental Standards:

- The system is designed to be robust and operational in diverse environments, addressing challenges such as low-light conditions and featureless spaces. Components like the **Camera** and **IMU** meet reliability standards for industrial and consumer electronics.

Expected weight, dimensions, and total power consumption

The expected weight of the Inside-Out Tracking Sensor Suite is approximately 1500 g, ensuring it remains lightweight and comfortable for extended use. The dimensions are designed to be compact, with an estimated size of 300 mm x 200 mm x 100 mm, allowing for easy integration into wearable systems. The total power consumption is calculated to be around 20 watts, accounting for the Raspberry Pi 5, IMU, camera, and other components, making it efficient and suitable for portable operation with standard power supplies.

3D Drawing of the System:

Current Visual sensor suite consists of single RGB camera which is, Raspberry Pi Camera Module 3. It can give video output at 1920x1080 resolution at 50 frames per second. It has a focal length of 7.4mm, horizontal and vertical field of views of 66 and 41 degrees. The image output is RGB array format.

In order to extract pose estimation, custom visual odometry algorithm is used, which finds 3 rotational angles, and translational direction vector which indicates the direction of motion.

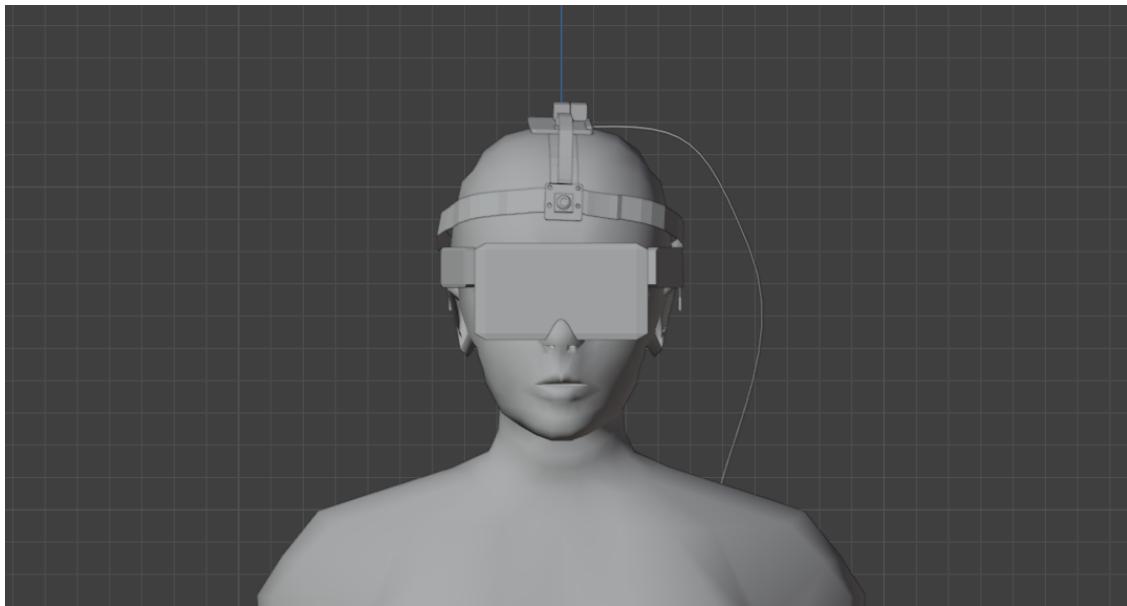


Figure 3 Headset design

Fig. 3 shows the front view of suite design with the main sensors: IMU and camera. The camera module will be attached to 3 flexible adjustable bands surrounding the head. Raspberry pi is located on the top with IMU components. An additional setup for the phone holder is designed for user view implementation. It contains an inner hole in which the phone will be put through sliding from the side entering.

Fig. 3 shows the front view of suite design with the main sensors: IMU and camera. The camera module will be attached to 3 flexible adjustable bands surrounding the head. Raspberry pi is located on the top with IMU components. An additional setup for the phone holder is designed for user view implementation. It contains an inner hole in which the phone will be put through sliding from the side entering.

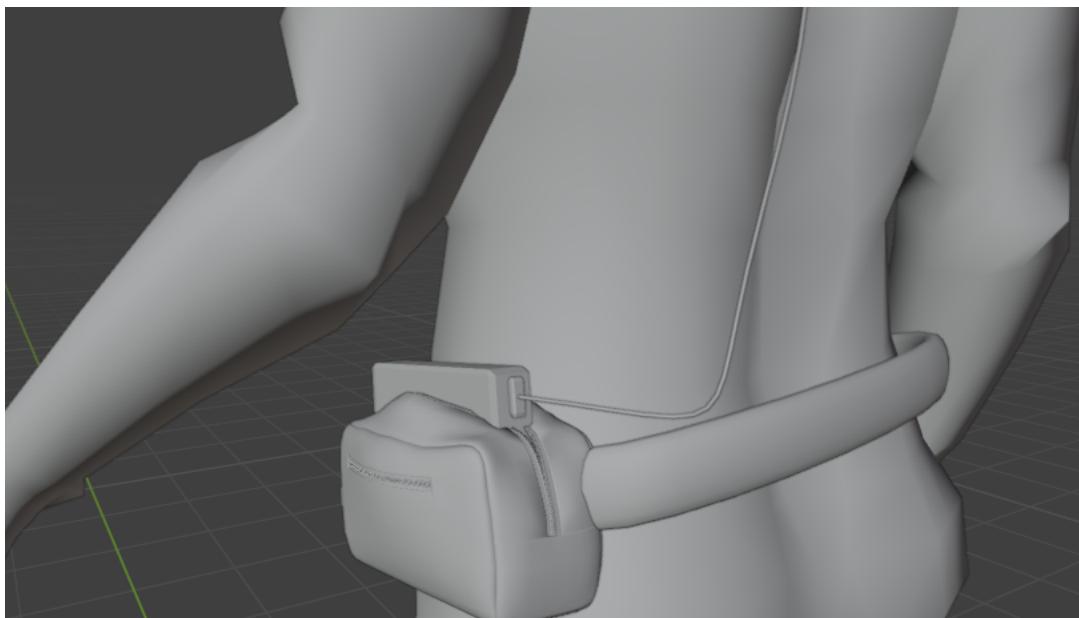


Figure 34 Power Unit Connected to Headset

Fig. 4 presents the power unit for the system components. A pouch connected to the waist band will contain the power bank. In case of any supplementary components, the waist bag will be used for the storing of the additional batteries.

Other than the phone holder, which will be printed using a 3D printer, flexible waist band and the head bands will provide comfortable clothing for the user. In fact, the headset is easily adjustable, which allows a satisfying VR experience. Additional design pictures from various angles are given in Appendix.

Sub-Systems:

Table 2 Subsystem-Level Requirement

| Subsystem-Level Requirement | IMU Subsystem Requirement | Camera Subsystem Requirement |
|--|---|--|
| Accurate and reliable pose estimation | The IMU must provide accurate rotational and translational data with an error margin of $\pm 4^\circ$ and ± 6 cm, respectively. | The camera must accurately track movement and provide pose estimation with an error margin of ± 6 cm and $\pm 4^\circ$. |
| Low latency for real-time tracking | The IMU data output rate must be at least 1 kHz for real-time updates. | The camera must capture video at a minimum resolution of 1920x1080 pixels with a frame rate of 50 fps. |
| Seamless sensor fusion for enhanced tracking accuracy | The IMU must support precise synchronization with other sensors using standardized communication protocols like I ² C. | The camera must integrate seamlessly with the IMU via sensor fusion algorithms for drift correction. |
| Robust performance in various environmental conditions | The IMU must maintain accuracy in environments with vibrations, temperature variations, and dynamic movements. | The camera must ensure robust tracking in low-light conditions and featureless environments. |
| Energy-efficient operation | The IMU must operate within a power consumption limit of 0.5 W to ensure overall system efficiency. | The camera must operate within a power consumption limit of 2 W to meet the total system power budget. |

| | | |
|--------------------------------|---|---|
| Compact and lightweight design | The IMU must not exceed a weight of 20 g to maintain portability. | The camera must not exceed a weight of 50 g to ensure ergonomic design. |
|--------------------------------|---|---|

- Standards Compliance

The IMU subsystem aligns well with recognized standards for sensor-based motion tracking in VR, such as the IEEE standards for inertial navigation systems, particularly for rotational tracking. The discrepancies in the camera subsystem's translational accuracy suggest a gap in meeting comparable standards for optical tracking, such as ISO standards for pose estimation systems, emphasizing the need for algorithmic and hardware improvements.

1. Inertial Sensor Suite

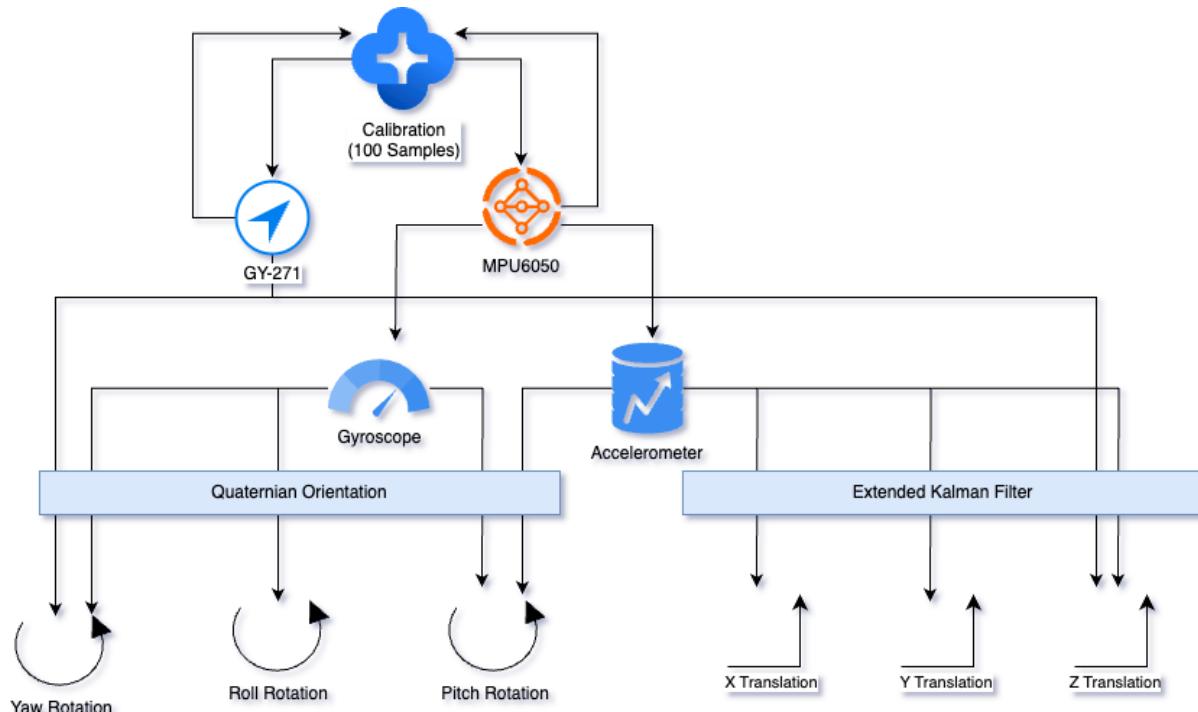


Figure 5.4 Inertial Sensors Algorithm Design

Translation measurements from the Inertial sensor suite

Translation measurements from an inertial sensor suite, such as one incorporating the MPU6050 IMU, can be improved using an Extended Kalman Filter (EKF) and techniques

like zero-velocity updates (ZUPT). The IMU is initialized via I2C communication and provides raw accelerometer and gyroscope data as a 16-bit signed integer, which is then scaled by 16384.0 to get the physical acceleration range $\pm 2g$. The first 100 samples are taken to calculate the bias of the system in the beginning. The data is then given to EKF to be processed to estimate the system's state, including position and velocity. The EKF operates in two main steps: prediction, which estimates the next state (position and velocity) based on the current state and time step, and update, which refines the prediction using new measurements from the sensor. However, due to sensor noise, bias, and drift, translational measurements often suffer from cumulative errors during double integration of acceleration data.

The zero-velocity update (ZUPT) technique is implemented to constrain the sensors' error. ZUPT uses the fact that a stance phase appears in each step at zero velocity to correct IMU errors periodically [12]. ZUPT addresses this by exploiting periods of detected stationarity, resetting the velocity to zero when the system is stationary enhances the accuracy of position estimates. However, the ZUPT has a drawback in our case the translation can be calculated when the system's acceleration is greater than a stationarity threshold. Stationarity threshold can be decreased but the bias is greater in this case. Combined with robust sensor calibration and error modeling, this approach significantly improves translational measurements.

Rotational measurements from the Inertial sensor suite

The primary purpose of the rotational measurement algorithm is to compute the orientation of a device in 3D space by fusing data from multiple sensors, including an accelerometer, gyroscope, and magnetometer. Accurate orientation estimation is critical for virtual reality (VR) applications, where even minor errors can lead to a disjointed user experience. This implementation combines sensor readings to estimate the roll, pitch, and yaw angles, which define the device's orientation in real-time.

IMU and Magnetometer Fusion: Purpose and Necessity

In orientation tracking systems, data from an Inertial Measurement Unit (IMU) and a magnetometer are fused to achieve accurate and stable orientation estimation. An IMU, typically consisting of a gyroscope and an accelerometer, measures angular velocity and acceleration, while a magnetometer provides data about the Earth's magnetic field. The fusion of these sensors allows the system to compute the device's orientation in terms of roll, pitch, and yaw.

The gyroscope in the IMU tracks rotational motion, offering high-frequency updates for orientation changes. However, gyroscopes suffer from drift, where small inaccuracies in measurements accumulate over time, leading to significant errors in orientation. The accelerometer, on the other hand, measures the direction of gravity and provides a stable reference for roll and pitch but is sensitive to linear motion and vibrations. A magnetometer complements these sensors by providing a reference for the yaw (heading) relative to the Earth's magnetic field, effectively eliminating drift in that axis.

Sensor fusion combines these complementary strengths, correcting the weaknesses of individual sensors. The gyroscope provides smooth, instantaneous updates, while the accelerometer and magnetometer anchor the orientation to real-world references, correcting for drift and maintaining long-term stability. This combination is critical for applications like virtual reality, where accurate and consistent orientation is essential for an immersive user experience.

The Necessity of Sensor Fusion

The necessity of fusing IMU and magnetometer data becomes evident when considering the limitations of using either sensor in isolation. For example, relying solely on the IMU would lead to drift over time, rendering the orientation estimation unreliable. Similarly, using only a magnetometer or accelerometer would result in incomplete orientation tracking, as they cannot capture rotational motion or handle dynamic movements effectively.

Sensor fusion algorithms, such as the Madgwick Filter, address these issues by integrating data from all three sensors. The accelerometer stabilizes roll and pitch, the magnetometer anchors yaw, and the gyroscope ensures smooth real-time updates. Together, these sensors create a robust orientation tracking system capable of operating in dynamic environments with minimal errors.

By combining a standalone IMU and magnetometer, we achieve the same level of functionality as a higher-cost integrated sensor, while maintaining flexibility and cost efficiency. This trade-off makes it an ideal solution for projects where budget constraints are a priority, but accuracy and stability cannot be compromised.

Sensor Fusion with Madgwick Filter

The core of the algorithm is the Madgwick Filter, a computationally efficient algorithm for orientation estimation. It uses a gradient descent method to minimize the error between measured and estimated sensor data, producing a quaternion-based representation of the orientation. This filter is particularly well-suited for low-power and embedded systems, as it delivers robust results without requiring significant computational resources.

Sensor fusion is achieved by integrating data from the accelerometer, gyroscope, and magnetometer. The accelerometer provides information about the direction of gravity, the gyroscope tracks rotational motion, and the magnetometer determines the device's orientation relative to the Earth's magnetic field. Combining these inputs mitigates the limitations of individual sensors, such as drift in the gyroscope or noise in the accelerometer.

Quaternion Theory and Its Use in the Algorithm

Quaternions are mathematical constructs that represent rotations in 3D space. They consist of four components: one scalar part (q_0) and three vector parts (q_1, q_2, q_3). This structure allows quaternions to represent orientations without the pitfalls of traditional methods like Euler angles, which suffer from issues such as gimbal lock [1]. In this algorithm, quaternions are used to track the orientation of the device relative to a fixed reference frame. By leveraging quaternion algebra, the algorithm can compute rotations, combine them, and update orientation in real-time with high numerical stability.

The use of quaternions is particularly advantageous for real-time systems. Unlike matrices, quaternions require fewer computations for rotation operations and are easier to normalize, ensuring that the orientation remains valid over time. This makes them well-suited for resource-constrained systems like the Raspberry Pi, which processes data from multiple sensors simultaneously.

Madgwick Filter: Overview and Structure

The Madgwick Filter is an orientation estimation algorithm designed for efficient sensor fusion in systems using an Inertial Measurement Unit (IMU) [2]. It combines data from the accelerometer, gyroscope, and magnetometer to calculate the orientation of the device. The algorithm is quaternion-based, avoiding singularities and ensuring smooth calculations, even in complex rotations. It is computationally lightweight, making it suitable for real-time applications like virtual reality, robotics, and motion tracking.

At its core, the filter integrates angular velocity data from the gyroscope to estimate orientation changes over time. However, gyroscope data is prone to drift, causing the orientation to deviate from the actual position over extended periods. To address this, the Madgwick Filter employs a feedback mechanism. This involves comparing the gyroscope-based orientation with measurements from the accelerometer and magnetometer, which provide absolute references for gravity and the Earth's magnetic field, respectively.

Sensor Data Fusion and Error Correction

The Madgwick Filter begins by normalizing sensor data to ensure consistent scaling, a crucial step for accurate calculations. The gyroscope data is integrated to compute the orientation, represented as a quaternion. To correct for drift, the algorithm calculates the error between the estimated orientation and the orientation implied by the accelerometer and magnetometer readings. This error is represented as a gradient and is used to adjust the quaternion update rate, ensuring alignment between the estimated and actual orientations.

The correction process is controlled by a parameter β , which determines the weight of the feedback. A higher β value leads to faster corrections but may introduce instability, while a lower β value provides smoother adjustments at the cost of slower convergence. This balance ensures that the system adapts quickly to changes while maintaining stable orientation estimates.

Computational Efficiency and Real-Time Performance

The Madgwick Filter is optimized for real-time applications, prioritizing computational efficiency. It avoids matrix operations and relies on quaternion mathematics to update orientation with minimal overhead. The algorithm's feedback mechanism uses a gradient descent approach, which is less computationally intensive than Kalman filters while still providing robust performance.

This efficiency makes the Madgwick Filter ideal for applications where processing power is limited but real-time responsiveness is critical. By combining the strengths of gyroscopes, accelerometers, and magnetometers, the filter delivers accurate and stable orientation tracking, even in noisy or dynamic environments. This capability is particularly valuable for VR systems, where precise orientation is essential for an immersive user experience.

Comparison Between Madgwick Filter and Kalman Filter

The Madgwick Filter and Kalman Filter are two prominent algorithms used for orientation estimation through sensor fusion, but they differ significantly in terms of computational complexity, noise handling, and overall accuracy. These differences determine their suitability for specific applications, such as real-time tracking in embedded systems or high-precision orientation estimation in robotics and aerospace.

The Madgwick Filter is designed to be computationally efficient, making it ideal for resource-constrained devices like microcontrollers or single-board computers. It uses quaternion mathematics to represent orientation, ensuring smooth rotations without issues like gimbal lock. The filter integrates gyroscope data for real-time orientation tracking and corrects for drift using accelerometer and magnetometer measurements. Its simplicity lies in employing a gradient descent method for error correction, which minimizes the discrepancy between estimated and actual sensor readings. However, the Madgwick Filter has limited noise modeling capabilities and relies on a fixed parameter, β , to balance responsiveness and stability. This can make it less adaptable in environments with highly variable noise or dynamic conditions.

In contrast, the Kalman Filter offers a probabilistic approach to sensor fusion by explicitly modeling sensor noise and system uncertainties. It combines gyroscope, accelerometer, and magnetometer data using a prediction-correction cycle, delivering highly accurate orientation estimates. The Kalman Filter dynamically adjusts its parameters based on the noise characteristics of the sensors, making it more robust in noisy or rapidly changing environments. However, its accuracy comes at the cost of computational complexity, as it requires recursive matrix operations and a detailed understanding of the system's dynamics. For applications with sufficient computational resources, such as robotics or aerospace systems, the Kalman Filter provides superior performance. Nonetheless, when implemented with Euler angles instead of quaternions, it risks encountering gimbal lock, a problem that can destabilize orientation tracking.

Gimbal lock is a well-known issue in 3D orientation estimation that occurs when using Euler angles to represent rotations. It happens when two of the three rotational axes align, resulting in a loss of one degree of freedom and leading to ambiguous or unstable calculations. Quaternions inherently avoid gimbal lock because they represent orientation as a four-dimensional structure, combining a scalar and a three-dimensional vector. Both the Madgwick Filter and the Kalman Filter can use quaternions to circumvent gimbal lock, but the Madgwick Filter is specifically designed around quaternion mathematics, simplifying its implementation and avoiding any reliance on Euler angles.

In summary, the Madgwick Filter is better suited for real-time applications on systems with limited computational power, such as VR devices or embedded systems. Its simplicity and efficiency make it a practical choice for scenarios where quick and reliable orientation tracking is essential. On the other hand, the Kalman Filter excels in high-accuracy applications where noise and system dynamics need to be explicitly modeled, provided there is adequate computational capacity. Both filters effectively avoid gimbal lock when quaternions are used, ensuring stable and continuous orientation tracking in 3D space.

Table 3.3 Comparison between Madgwick Filter and Kalman Filter

| Feature | Madgwick Filter | Kalman Filter |
|---------------------------------|---|--|
| Computational Efficiency | High (low processing requirements) | Moderate to low (requires matrix operations) |
| Accuracy | Good for real-time, less noisy environments | Excellent, especially in noisy conditions |
| Complexity | Simple to implement | Complex, requires accurate modeling |

| | | |
|------------------------------|--|--|
| Noise Handling | Limited noise modeling | Explicit noise and uncertainty modeling |
| Gimbal Lock Avoidance | Naturally avoids with quaternions | Avoidable only if using quaternions |
| Use Case | Low-power embedded systems, real-time VR | High-accuracy systems, robotics, aerospace |

Calibration Algorithm for Sensor Accuracy

Calibration is a vital process for ensuring the accuracy and reliability of sensor measurements, particularly for devices like gyroscopes, accelerometers, and magnetometers used in orientation tracking systems. Without calibration, sensor outputs often contain biases, offsets, or scaling errors that can distort the results and degrade the performance of sensor fusion algorithms. The calibration algorithm addresses these issues by systematically processing raw sensor data to establish baseline corrections and remove distortions.

The gyroscope calibration focuses on eliminating bias, which is the non-zero reading that the gyroscope may produce even when the device is stationary. This is achieved by collecting a set of raw gyroscope measurements over a fixed period while ensuring the sensor remains motionless. The algorithm calculates the average value of these readings for each axis, representing the gyroscope's static offset. These offsets are then subtracted from subsequent readings to ensure that the gyroscope outputs zero when no angular velocity is present, reducing cumulative drift during orientation estimation. A low-pass filter is applied to the gyroscope data to smooth out high-frequency noise and improve the stability of the orientation estimates. This preprocessing step enhances the reliability of the system, especially in environments with significant sensor noise or rapid movements.

For the accelerometer, calibration ensures that the sensor accurately measures both the magnitude and direction of acceleration, including the gravitational force. The process involves placing the accelerometer in known orientations, where one axis aligns with gravity. By comparing the measured acceleration to the expected value (approximately 9.81 m/s^2), the algorithm identifies any scaling errors or offsets. These discrepancies are corrected by adjusting the readings, and the corrected values are normalized to unit vectors. This step ensures that the accelerometer provides reliable data for stabilizing roll and pitch during sensor fusion.

The magnetometer calibration addresses distortions caused by hard iron effects (permanent magnetic disturbances) and soft iron effects (distortions due to nearby ferromagnetic materials). The sensor is rotated through various orientations to record the magnetic field in all directions. Hard iron distortions shift the magnetic field readings from the origin, which the algorithm corrects by recentering the data. Soft iron effects, which distort the data's spherical shape into an ellipse, are corrected by reshaping the data into

a sphere using a scaling matrix. After applying these corrections, the magnetometer readings are normalized to provide consistent and accurate heading information.

1.HMC5883L Magnetometer

The **HMC5883L** is a 3-axis digital magnetometer designed for precise magnetic field sensing and navigation applications. It uses anisotropic magnetoresistive (AMR) technology to measure magnetic fields in the x, y, and z axes

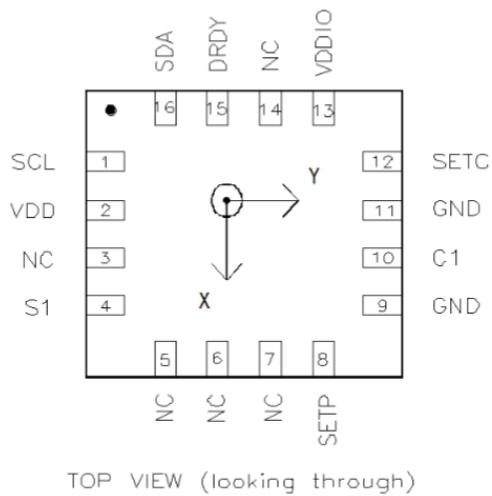


Figure 6.5 HMC5883L Magnetometer

Table 4. 4 Technical Information about Magnetometer

HMC5883L**SPECIFICATIONS** (* Tested at 25°C except stated otherwise.)

| Characteristics | Conditions* | Min | Typ | Max | Units |
|-----------------------------------|---|--------------|--------------|----------------|----------------|
| Power Supply | | | | | |
| Supply Voltage | VDD Referenced to AGND VDDIO Referenced to DGND | 2.16 1.71 | 2.5 1.8 | 3.6 VDD+0.1 | Volts Volts |
| Average Current Draw | Idle Mode Measurement Mode (7.5 Hz ODR; No measurement average, MA1:MA0 = 00) VDD = 2.5V, VDDIO = 1.8V (Dual Supply) VDD = VDDIO = 2.5V (Single Supply) | - - | 2 100 | - - | µA µA |
| Performance | | | | | |
| Field Range | Full scale (FS) | -8 | | +8 | gauss |
| Mag Dynamic Range | 3-bit gain control | ±1 | | ±8 | gauss |
| Sensitivity (Gain) | VDD=3.0V, GN=0 to 7, 12-bit ADC | 230 | | 1370 | LSb/gauss |
| Digital Resolution | VDD=3.0V, GN=0 to 7, 1-LSb, 12-bit ADC | 0.73 | | 4.35 | milli-gauss |
| Noise Floor (Field Resolution) | VDD=3.0V, GN=0, No measurement average, Standard Deviation 100 samples (See typical performance graphs below) | | 2 | | milli-gauss |
| Linearity | ±2.0 gauss input range | | | 0.1 | ±% FS |
| Hysteresis | ±2.0 gauss input range | | ±25 | | ppm |
| Cross-Axis Sensitivity | Test Conditions: Cross field = 0.5 gauss, Happlied = ±3 gauss | | ±0.2% | | %FS/gauss |
| Output Rate (ODR) | Continuous Measurement Mode Single Measurement Mode | 0.75 | | 75 160 | Hz Hz |
| Measurement Period | From receiving command to data ready | | 6 | | ms |
| Turn-on Time | Ready for I2C commands Analog Circuit Ready for Measurements | | 200 50 | | µs ms |
| Gain Tolerance | All gain/dynamic range settings | | ±5 | | % |
| I ² C Address | 8-bit read address 8-bit write address | | 0x3D 0x3C | | hex hex |
| I ² C Rate | Controlled by I ² C Master | | | 400 | kHz |

Figure 7. Technical Information for HMC5885L [9]

2.MPU6050 IMU

The MPU6050 is a 6-axis Inertial Measurement Unit (IMU) that combines a 3-axis accelerometer and a 3-axis gyroscope in a single chip.

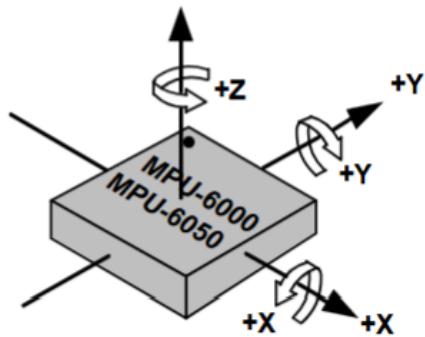


Figure 8. 6 IMU Orientation of Axes of Sensitivity and Polarity of Rotation

5.1 Gyroscope Features

The triple-axis MEMS gyroscope in the MPU-60X0 includes a wide range of features:

- Digital-output X-, Y-, and Z-Axis angular rate sensors (gyroscopes) with a user-programmable full-scale range of ± 250 , ± 500 , ± 1000 , and $\pm 2000^{\circ}/sec$
- External sync signal connected to the FSYNC pin supports image, video and GPS synchronization
- Integrated 16-bit ADCs enable simultaneous sampling of gyros
- Enhanced bias and sensitivity temperature stability reduces the need for user calibration
- Improved low-frequency noise performance
- Digitally-programmable low-pass filter
- Gyroscope operating current: 3.6mA
- Standby current: 5 μ A
- Factory calibrated sensitivity scale factor
- User self-test

5.2 Accelerometer Features

The triple-axis MEMS accelerometer in MPU-60X0 includes a wide range of features:

- Digital-output triple-axis accelerometer with a programmable full scale range of $\pm 2g$, $\pm 4g$, $\pm 8g$ and $\pm 16g$
- Integrated 16-bit ADCs enable simultaneous sampling of accelerometers while requiring no external multiplexer
- Accelerometer normal operating current: 500 μ A
- Low power accelerometer mode current: 10 μ A at 1.25Hz, 20 μ A at 5Hz, 60 μ A at 20Hz, 110 μ A at 40Hz
- Orientation detection and signaling
- Tap detection
- User-programmable interrupts
- High-G interrupt
- User self-test

Figure 9. 7 Information about Gyroscope and Accelerometer

5.3 Additional Features

The MPU-60X0 includes the following additional features:

- 9-Axis MotionFusion by the on-chip Digital Motion Processor (DMP)
- Auxiliary master I²C bus for reading data from external sensors (e.g., magnetometer)
- 3.9mA operating current when all 6 motion sensing axes and the DMP are enabled
- VDD supply voltage range of 2.375V-3.46V
- Flexible VLOGIC reference voltage supports multiple I²C interface voltages (MPU-6050 only)
- Smallest and thinnest QFN package for portable devices: 4x4x0.9mm
- Minimal cross-axis sensitivity between the accelerometer and gyroscope axes
- 1024 byte FIFO buffer reduces power consumption by allowing host processor to read the data in bursts and then go into a low-power mode as the MPU collects more data
- Digital-output temperature sensor
- User-programmable digital filters for gyroscope, accelerometer, and temp sensor
- 10,000 g shock tolerant
- 400kHz Fast Mode I²C for communicating with all registers

Figure 108 .Technical Information about IMU [10]

2. Camera

In order to extract pose estimation, custom visual odometry algorithm is used, which finds 3 rotational angles, and translational direction vector which indicates the direction of motion.

Visual Odometry Algorithm Breakdown

Visual odometry algorithm, relies on finding the feature based epipolar geometry between the two frames. In initialization the first frame taken as “keyframe” which features are extracted, then for each frame new locations of the “key features” are found using sparse optical flow, and essential matrix and pose difference between the two frames are found based on these features. If the current frame is distinguished enough to can't find enough corresponding features, feature extraction is repeated on current frame, and keyframe is replaced by the current frame.

Feature Extraction

For the feature extraction Shi-Tomasi corner detector [10] is used. In order to corner detection, RGB array is turned into grayscale, and based on the gradient differences of the grayscale image, edges can be found. If a location on the image is “edge” in both x and y directions location is considered to be a corner.

Optical Flow

In order to find corresponding locations of the features in the keyframe on current frame, Lukas-Kanade sparse optical flow algorithm [12] is used. The algorithm assumes that the motion of pixels in a small neighborhood around a point is constant and solves the “optical flow equation”, which indicates 3D motions correspondence on the 2D image, for this motion by minimizing the error in the brightness constancy constraint.

Essential Matrix Calculation

The essential matrix encapsulates the epipolar geometry between two views of the same scene captured by a calibrated camera. The essential matrix relates corresponding points in two images and is used in 3D reconstruction and camera pose estimation. In order to calculate it we use two sets of corresponding 2D points, the intrinsic camera matrix

(containing focal length and principal point information). By solving the equation $x'^T E x = 0$ Where E is the essential matrix, x and x' are the coordinates of the corresponding features. We specifically use 5-point solution discussed in [15]. Also, in order to eliminate the features that are independent of the camera movements, outliers, we use random sample consensus RANSAC algorithm, which iteratively analyzes the consistency of subsets of the points to overall points and eliminates the inconsistent ones.

Pose Recovery from Essential Matrix

Once the essential matrix is calculated, in order to obtain translation and rotation, Singular Value Decomposition [7], which decomposes essential matrix into rotational matrix and translation array. But this decomposition yields multiple probable solutions. Then using cheirality test [4] which triangulates one point and two camera positions and checks whether the points depth is positive.

Hardware Specifications For Camera Subsystem:

1. RaspberryPi Module 3 Camera

Current Visual sensor suite consists of single RGB camera which is, Raspberry Pi Camera Module 3. It can give video output at 1920x1080 resolution at 50 frames per second. It has a focal length of 7.4mm, horizontal and vertical field of views of 66 and 41 degrees. The image output is RGB array format.

Table 55. Technical Specification of Camera Subsystem

| | |
|-------------------------------|--|
| Sensor: | Sony IMX708 |
| Resolution: | 11.9 megapixels |
| Sensor size: | 7.4mm sensor diagonal |
| Pixel size: | 1.4µm x 1.4µm |
| Horizontal/vertical: | 4608 x 2592 pixels |
| Common video modes: | 1080p50, 720p100, 480p120 |
| Output: | RAW10 |
| IR cut filter: | Integrated in standard variants; not present in NoIR variants |
| Autofocus system: | Phase Detection Autofocus |
| Dimensions: | 25 x 24 x 11.5mm (12.4mm height for Wide variants) |
| Ribbon cable length: | 200mm |
| Cable connector: | 15 x 1mm FPC |
| Operating temperature: | 0°C to 50°C |
| Compliance: | FCC 47 CFR Part 15, Subpart B, Class B Digital Device Electromagnetic Compatibility Directive (EMC) 2014/30/EU Restriction of Hazardous Substances (RoHS) Directive 2011/65/EU |
| Production lifetime: | Raspberry Pi Camera Module 3 will remain in production until at least January 2030 |

Plan B for Subsystems

To mitigate risks associated with critical subsystems, contingency plans have been established to ensure system functionality in case of failures:

1. IMU Subsystem:

- Primary Issue: Inaccuracy or failure due to sensor drift, noise, or environmental factors.
- Plan B: Replace the MPU6050 IMU with a higher-precision alternative, such as the BNO055, which integrates advanced sensor fusion directly on-chip, reducing dependency on external algorithms. Additionally, implement redundant IMU units to cross-verify data and improve reliability.

2. Camera Subsystem:

- Primary Issue: Poor performance in low-light or featureless environments, or hardware malfunction.
- Plan B: Integrate a secondary camera module, such as a stereo camera or an RGB-D sensor (e.g., Intel RealSense), to enhance depth perception and robustness in challenging scenarios. Alternatively, use a software-based enhancement, such as adaptive brightness correction or feature augmentation algorithms, to compensate for hardware limitations.

Test Results

The test procedures and results for the subsystems of the "Inside-Out Tracking Sensor Suite" have been systematically evaluated to assess their performance against defined requirements. The primary subsystems tested include the IMU-based inertial tracking subsystem and the camera-based visual tracking subsystem. These tests were conducted in the D Block study room, ensuring a controlled environment with no vibrations, consistent light intensity to minimize external disturbances and a feature-rich view for the camera. The IMU tests were completed on **25.12.2024**, while the camera tests were conducted on **28.12.2024**. The all test table can be found in the appendix-2.



Figure 11. Feature-rich view for the camera

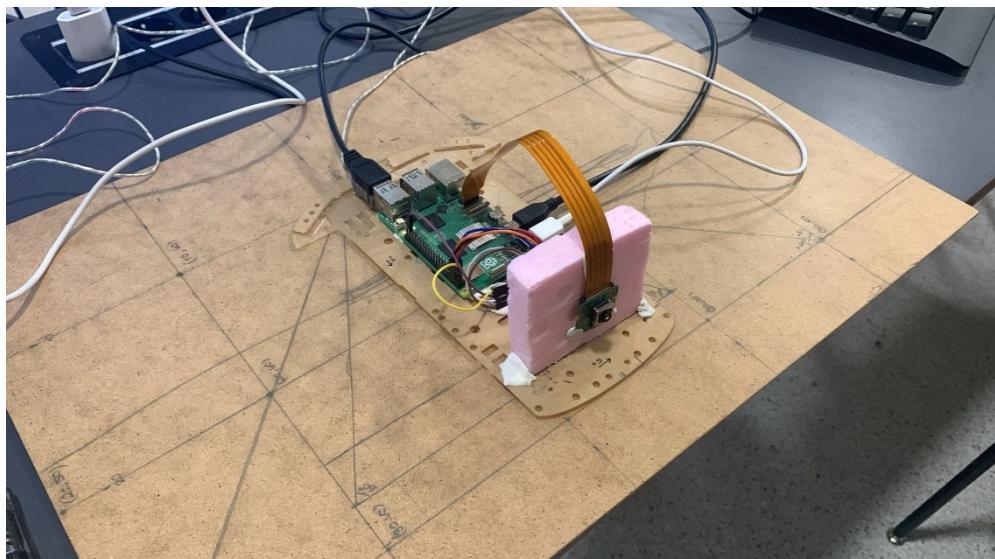


Figure 12. Vibration Free Table

Table 6. Test Results

| Test ID Case | Status | Test Campaign ID | Verification Method | Description | Expected Results (Pass/Fail Criteria) |
|-------------------------|--------|------------------|---------------------|---|--|
| OSEAM-TC-C/DT-01 | Passed | SNR-IMU | Review Of Design | <p>The head pose estimation system, utilizing the MPU6050 IMU and HMC5883L magnetometer, as well as the equivalent-level tests to validate the design, are reviewed on both subsystem and component levels to ensure compliance with all associated requirements. This review includes verifying the integration, calibration, and testing of the sensors as required by the system's concept of operations (CONOPS), referencing engineering drawings, designs, and requirements. Critical parameters such as sensor alignment, data accuracy, availability of signals, and compatibility with pose estimation algorithms are checked and validated.</p> | <p>The head pose estimation system, including interfaces between the MPU6050 IMU and HMC5883L magnetometer, as well as the processing unit and associated software, undergoes validation through its entire development cycle (i.e., proof of concept, design, development, assembly, and testing). This validation ensures compliance with internal documentation and approval from the system owner.</p> |
| OSEAM-TC-C/DT-02 | Passed | SNR-IMU | Inspection | <p>The MPU6050 IMU and HMC5883L magnetometer are subjected to a general visual inspection upon integration into the VR system to ensure the physical quality of the components and their suitability for head pose estimation.</p> | <p>The MPU6050 IMU and HMC5883L magnetometer show no signs of physical damage, deformation, corrosion, or any other visual defects. All labels and markings are intact and legible, ensuring the sensors are in optimal condition for use in the VR system.</p> |

| | | | | |
|-----------------------------------|--------|---------|------|---|
| OSEAM- TC- C/DT-03 | Passed | SNR-IMU | Test | <p>This test's goal is to evaluate the translational accuracy along the x-axis, measured primarily by the accelerometer in the MPU6050 IMU. The sensor is connected to an independent power source, and controlled linear motion is applied along the x-axis. The accelerometer data is recorded and processed to calculate displacement, and the accuracy is verified against predefined benchmarks. The test ensures that the x-axis measurements meet the required precision standards, with minimal deviation from the true values.</p> <p>The test is considered a pass if the measured translational displacement along the x-axis deviates by no more than ± 6 cm from the reference value. Any deviation exceeding this range will result in a fail for the x-axis test.</p> |
| OSEAM- TC- C/DT-04 | Passed | SNR-IMU | Test | <p>This test is designed to assess the translational accuracy along the y-axis, based on readings from the accelerometer in the MPU6050 IMU. The sensor is independently powered, and controlled linear motion is applied along the y-axis. Data from the accelerometer is analyzed to compute the relative displacement, and the results are compared with reference values. The test ensures that the y-axis measurements are accurate and reliable, adhering to the system's performance requirements.</p> <p>For the y-axis, the test is deemed a pass if the measured translational displacement remains within ± 6 cm of the expected reference value. Deviations greater than ± 6 cm will constitute a fail for the y-axis test.</p> |

| | | | | | |
|-------------------------|----------|---------|------|---|--|
| OSEAM-TC-C/DT-05 | Tailored | SNR-IMU | Test | <p>This test aims to evaluate the translational accuracy along the z-axis, focusing on vertical movements as detected by the MPU6050 IMU accelerometer. Using an independent power source, the sensor is subjected to controlled vertical motion, and its output is processed to determine displacement. The accuracy of the z-axis measurements is compared with standard reference values to ensure compliance with the required specifications. The test validates that the z-axis measurements are precise and suitable for system integration.</p> | <p>The z-axis test is considered successful (a pass) if the measured vertical displacement does not exceed a deviation of ± 6 cm from the reference value. A deviation beyond this threshold will result in a fail for the z-axis test.</p> |
| OSEAM-TC-C/DT-06 | Passed | SNR-IMU | Test | <p>This test's goal is to evaluate the rotational accuracy around the roll axis, measured using the gyroscope and magnetometer in the MPU6050 sensor. Controlled rotations are applied around the roll axis, and the sensor data is recorded to calculate the angular displacement. The test ensures that the calculated roll angles remain within the required precision limits.</p> | <p>The test is considered a pass if the measured roll angle deviates by no more than $\pm 4^\circ$ from the reference value. A deviation exceeding $\pm 4^\circ$ results in a fail.</p> |
| OSEAM-TC-C/DT-07 | Passed | SNR-IMU | Test | <p>The purpose of this test is to assess the rotational accuracy around the pitch axis using data from the gyroscope and accelerometer. Controlled rotations are applied around the pitch axis, and the sensor outputs are analyzed to determine angular displacement. The test ensures accurate pitch angle measurements within the specified tolerance.</p> | <p>The test is deemed a pass if the measured pitch angle deviation is within $\pm 4^\circ$ of the reference value. Any deviation greater than $\pm 4^\circ$ constitutes a fail.</p> |

| | | | | | |
|-----------------------------------|--------|---------|------|---|--|
| OSEAM- TC- C/DT-08 | Passed | SNR-IMU | Test | <p>This test aims to evaluate the rotational accuracy around the yaw axis by utilizing data from the MPU6050 and HMC5883L. Controlled rotations are performed around the yaw axis, and the resulting sensor data is processed to calculate angular displacement. The accuracy of the yaw measurements is compared against the reference to validate performance.</p> | <p>The test is considered a pass if the measured yaw angle deviation does not exceed $\pm 4^\circ$ from the reference value. Deviations beyond this range will result in a fail.</p> |
| OSEAM- TC- C/DT-09 | Passed | SNR-IMU | Test | <p>This test evaluates the system's ability to track combined yaw rotation and translational motion in the x-y plane using the MPU6050 IMU and HMC5883L magnetometer. Controlled simultaneous yaw rotation and linear motion along both the x-axis and y-axis are applied. The IMU captures accelerometer data for translational displacement, while the gyroscope and magnetometer data are used to calculate yaw angular displacement. The processed data is analyzed to ensure compliance with required accuracy standards for both rotational and translational measurements.</p> | <p>The test is considered a pass if the yaw angle deviation does not exceed $\pm 4^\circ$ and the combined x-y translational displacement deviation remains within ± 6 cm of the reference values. Deviations beyond these thresholds will result in a fail.</p> |

| | | | | | |
|-----------------------------------|--------|---------|---------------------|---|---|
| OSEAM- TC- C/DT-10 | Passed | SNR-CAM | Review Of Design | <p>The head pose estimation system, utilizing the camera for rotational and translational tracking, as well as the equivalent-level tests to validate the design, are reviewed on both subsystem and component levels to ensure compliance with all associated requirements. This review includes verifying the integration, calibration, and testing of the camera as required by the system's concept of operations (CONOPS), referencing engineering drawings, designs, and requirements. Critical parameters such as camera alignment, image quality, data processing accuracy, availability of signals, and compatibility with pose estimation algorithms are checked and validated.</p> | <p>The head pose estimation system, including the camera interfaces with the processing unit and associated software, undergoes validation throughout its entire development cycle (i.e., proof of concept, design, development, assembly, and testing). This validation ensures compliance with internal documentation and approval from the system owner.</p> |
| OSEAM- TC- C/DT-11 | Passed | SNR-CAM | Inspection | <p>The camera is subjected to a general visual inspection upon integration into the VR system to ensure the physical quality of the component and its suitability for head pose estimation.</p> | <p>The camera shows no signs of physical damage, deformation, corrosion, or any other visual defects. All labels and markings are intact and legible, ensuring the component is in optimal condition for use in the VR system.</p> |
| OSEAM- TC- C/DT-12 | Failed | SNR-CAM | Test | <p>This test evaluates the translational accuracy along the x-axis using the camera's ability to detect and track movement. Controlled linear motion is applied along the x-axis, and the camera captures frame-by-frame image data to calculate displacement. The test ensures that the</p> | <p>The test is considered a pass if the measured translational displacement along the x-axis deviates by no more than ± 6 cm from the reference value. Deviations exceeding ± 6 cm result in a fail.</p> |

| | | | | | |
|-------------------------|--------|---------|------|---|--|
| | | | | calculated x-axis translations are accurate and within the required tolerance. | |
| OSEAM-TC-C/DT-13 | Failed | SNR-CAM | Test | This test assesses the translational accuracy along the y-axis using the camera's motion tracking capabilities. Controlled linear motion is applied along the y-axis, and the camera processes the captured images to compute the displacement. The accuracy of the y-axis measurements is validated against predefined benchmarks. | The test is deemed a pass if the measured translational displacement along the y-axis deviates by no more than ± 6 cm from the reference value. Deviations beyond ± 6 cm constitute a fail. |
| OSEAM-TC-C/DT-14 | Failed | SNR-CAM | Test | This test evaluates the translational accuracy along the z-axis using the camera's depth perception or tracking methods. Controlled vertical motion is applied along the z-axis, and the camera captures and analyzes the displacement data to determine accuracy. The test ensures precise z-axis measurements within the specified limits. | The test is considered a pass if the measured translational displacement along the z-axis deviates by no more than ± 6 cm from the reference value. Deviations exceeding ± 6 cm will result in a fail. |
| OSEAM-TC-C/DT-15 | Passed | SNR-CAM | Test | This test evaluates the rotational accuracy around the roll axis using the camera's orientation tracking. Controlled rotations are applied around the roll axis, and the camera captures image data to calculate angular displacement through image processing algorithms. The test verifies that the calculated roll angles are within the acceptable tolerance. | The test is considered a pass if the measured roll angle deviates by no more than $\pm 4^\circ$ from the reference value. A deviation exceeding $\pm 4^\circ$ results in a fail. |

| | | | | | |
|-----------------------------------|----------|---------|------|--|---|
| OSEAM- TC- C/DT-16 | Tailored | SNR-CAM | Test | <p>This test assesses the rotational accuracy around the pitch axis using the camera's ability to detect angular changes. Controlled rotations are applied around the pitch axis, and the camera processes image data to estimate angular displacement. The test ensures that the pitch angle measurements adhere to the required precision standards.</p> | <p>The test is deemed a pass if the measured pitch angle deviation is within $\pm 4^\circ$ of the reference value. Any deviation greater than $\pm 4^\circ$ constitutes a fail.</p> |
| OSEAM- TC- C/DT-17 | Passed | SNR-CAM | Test | <p>This test evaluates the rotational accuracy around the yaw axis using the camera's orientation tracking capabilities. Controlled rotations are performed around the yaw axis, and the camera's image processing algorithms calculate angular displacement. The measured yaw angles are compared to the reference to ensure compliance with accuracy requirements.</p> | <p>The test is considered a pass if the measured yaw angle deviation does not exceed $\pm 4^\circ$ from the reference value. Deviations beyond this range will result in a fail.</p> |
| OSEAM- TC- C/DT-18 | Not Done | SNR-CAM | Test | <p>This test evaluates the system's ability to track combined yaw rotation and translational motion in the x-y plane using the camera. Controlled simultaneous yaw rotation and linear motion along both the x-axis and y-axis are applied. The camera captures image data, and the system processes it to compute angular displacement (yaw) and translational displacement (x and y). The results are analyzed to ensure compliance with required accuracy standards for both rotational and translational measurements.</p> | <p>The test is considered a pass if the yaw angle deviation does not exceed $\pm 4^\circ$ and the combined x-y translational displacement deviation remains within ± 6 cm of the reference values. Deviations exceeding these thresholds will result in a fail.</p> |

- IMU Translational Accuracy Tests

The translational accuracy tests are designed to evaluate the system's capability to measure and track linear displacements along the x, y, and z axes. These tests are essential to verify the performance of the MPU6050 IMU and associated subsystems in accurately estimating translational motion within specified tolerances. Controlled forward and backward movements, as well as initial static positions, are assessed to ensure the system's reliability and precision under various scenarios.

The tests along the **x-axis** begin with a static initial position to establish a baseline for horizontal measurements, ensuring the system can provide accurate reference values {1}. Following this, a controlled forward movement of 20 cm is applied to simulate common horizontal motion scenarios such as walking or object displacement {2}. A subsequent backward motion of 20 cm assesses the system's ability to track reverse movements while detecting potential drift or inaccuracies introduced during directional changes {3}. Finally, a longer forward motion of 40 cm is conducted to evaluate the system's precision over extended distances, ensuring reliable performance under increased displacement scenarios {4}.

The **y-axis** tests begin with a static initial position to verify baseline accuracy for lateral movements {5}. This is followed by a controlled forward lateral movement of 20 cm, simulating scenarios like sidestepping or lateral shifts {6}. A 20 cm backward motion tests the system's ability to track reverse lateral movements accurately, ensuring stability and minimal drift in the opposite direction {7}. To further evaluate precision, a 40 cm forward lateral motion is performed, challenging the system's reliability over greater distances while maintaining accuracy {8}.

The **z-axis** tests start with a static initial position to confirm the system's baseline accuracy for vertical measurements {9}. A controlled upward motion of 20 cm is then applied to simulate elevation changes, such as climbing or lifting scenarios {10}. This is followed by a downward motion of 20 cm to assess the system's ability to measure descent accurately, considering the influence of gravity and potential sensor drift {11}. Finally, a 40 cm upward motion is performed to test the system's accuracy over larger vertical displacements, ensuring it can handle extended elevation changes with precision {12}.

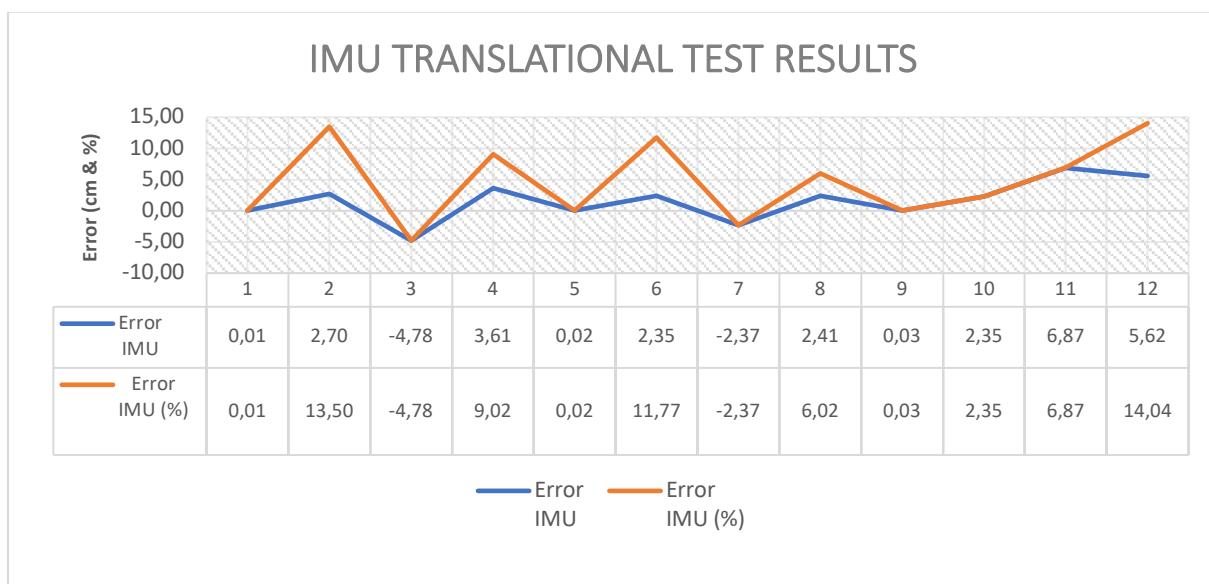


Figure 13. 9IMU Translational Test Results

- Rotational Accuracy Tests

The rotational accuracy tests are conducted to assess the system's ability to track angular displacements along the roll, pitch, and yaw axes under specific controlled conditions. Each test simulates distinct rotational movements, mimicking real-world scenarios, to evaluate how accurately the system captures and processes angular changes. The conditions for each test case are designed to challenge the IMU and camera integration, ensuring compliance with predefined tolerances and operational requirements.

In the **roll axis tests**, rotations include transitions such as from an initial position to -60° {1}, simulating a steep downward tilt. Subsequent rotations (-60° to 0°) simulate recovery to a neutral position {2}, followed by incremental upward tilts (e.g., 0° to 45° {3}, 45° to 60° {4}). These conditions test the IMU's ability to handle continuous angular transitions with minimal drift or noise.

For the **yaw axis tests**, rotations from an initial position to -150° {5} simulate extensive horizontal turns, representing scenarios like head tracking in VR systems or large directional changes in robotics. Additional conditions include smoother angular movements (-150° to -60° {6}, -60° to 0° {7}, and 0° to 90° {9}) to assess the system's response to varied angular velocities and directional changes.

The **pitch axis tests** introduce conditions like rotations from the initial position to -60° , simulating a downward nod, and reverse transitions back to the initial position (-60° to Initial) {10}. These are followed by upward tilts to 45° {11} and further to a referenced angle of 60° {12}. These movements test the system's ability to measure vertical angular changes under conditions where gravity significantly influences the sensors.

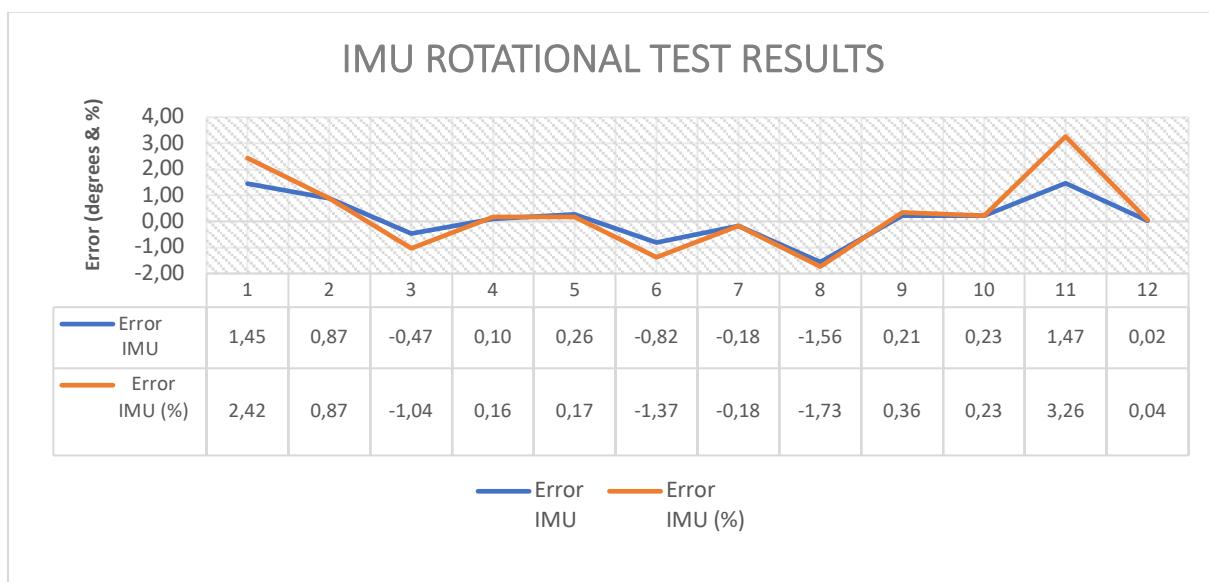


Figure 14.10 IMU Rotational Test Results

- Combined Accuracy Tests

The combined translational and rotational accuracy tests evaluate the system's capability to simultaneously track linear and angular displacements. These tests are critical for assessing the integration and synchronization of translational motion along the x- and y-axes with rotational motion around the yaw axis. By simulating realistic scenarios, such as coordinated movements involving directional shifts and angular rotations, these tests ensure the system's performance meets the accuracy standards required for complex motion applications. The conditions include a combination of static and dynamic motions, involving varying distances and angles, to validate the system's ability to maintain precision in multi-dimensional movements.

The combined tests for translational motion along the x-axis and rotational motion around the yaw axis begin with a static scenario (0 cm/0°) to establish a baseline accuracy for both translational and angular measurements {1}. Next, a simultaneous forward motion of 20 cm along the x-axis and a 90° yaw rotation is performed to evaluate the system's ability to accurately track complex combined motions {2}. A shorter forward movement of 10 cm paired with a 60° yaw rotation further challenges the system, testing its synchronization and precision in scenarios involving smaller but rapid motions {3}.

The combined tests for translational motion along the y-axis and rotational motion around the yaw axis also start with a static position (0 cm/0°) to confirm baseline performance {4}. A forward lateral movement of 20 cm along the y-axis combined with a 90° yaw rotation assesses the system's ability to manage simultaneous lateral and rotational displacements {5}. Finally, a shorter 10 cm lateral movement paired with a 60° yaw rotation provides a focused evaluation of the system's synchronization capabilities during combined motions of smaller magnitude {6}.

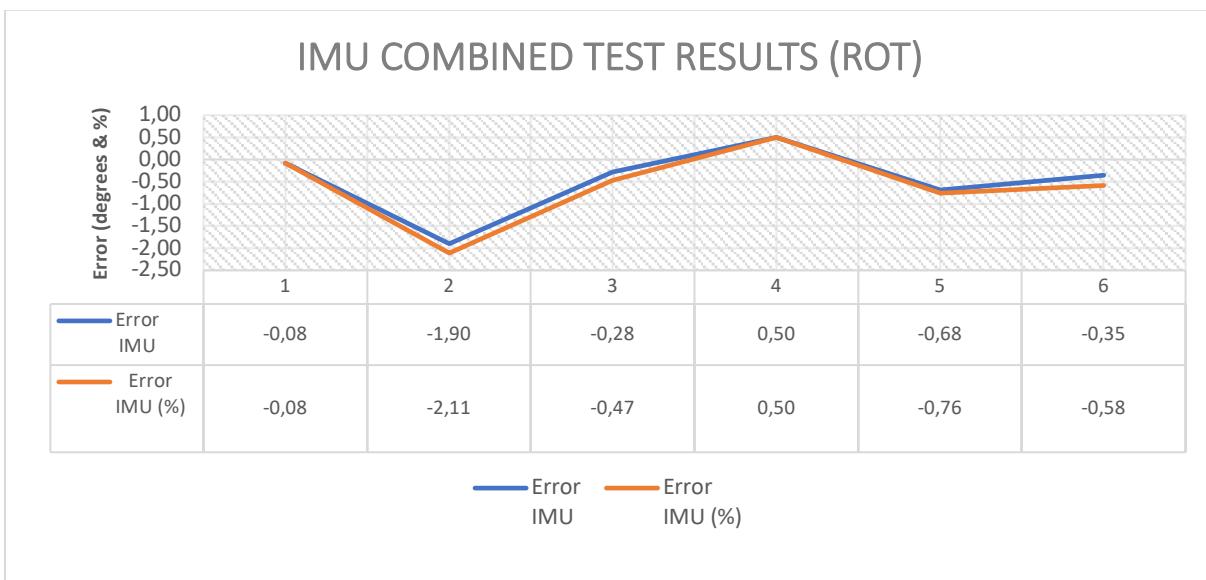


Figure 15. IMU Multiaxes Test Results (Rotation)

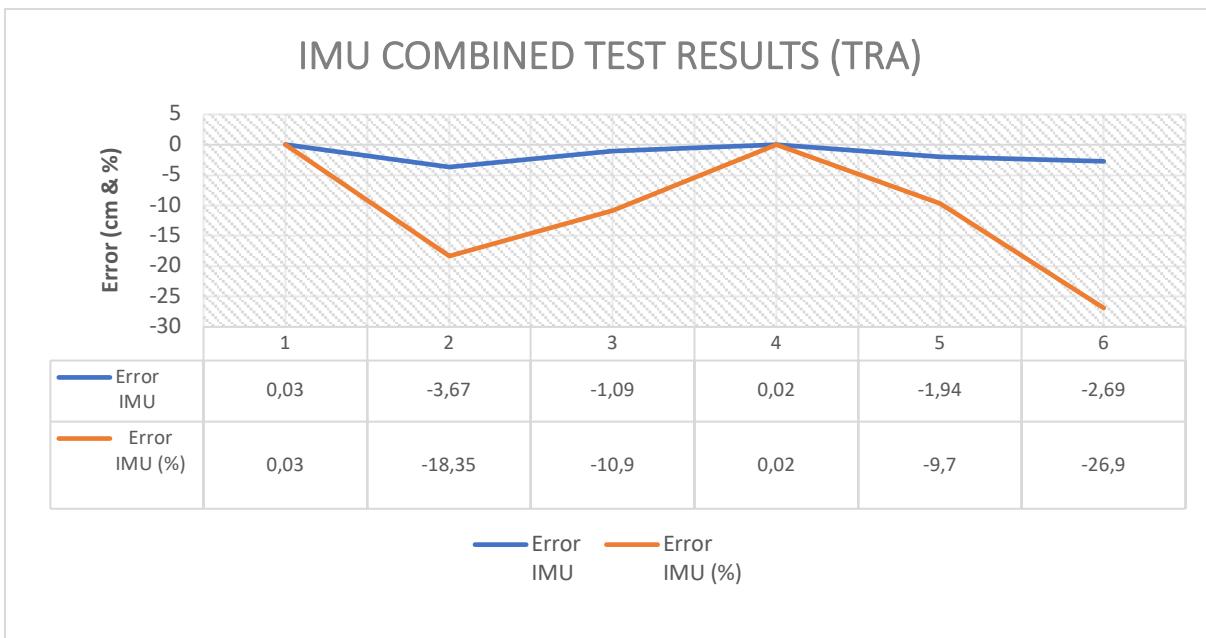


Figure 16. IMU Multiaxes Test Results (Translation)

- IMU Test Results Discussion and Evaluation

The performance analysis of the IMU system demonstrates overall compliance with specified accuracy thresholds in most tests, confirming the robustness of the MPU6050 IMU and HMC5883L magnetometer for head pose estimation. Translational accuracy along the x- and y-axes consistently met the ± 6 cm threshold, reflecting stable and reliable performance. However, the z-axis showed a notable deviation of 6.87 cm during 20 cm backward motion, exceeding the threshold.

Rotational accuracy across roll, pitch, and yaw axes consistently fell within the $\pm 4^\circ$ tolerance. The combined motion tests involving yaw and translational movement in the x-y plane were also successful, demonstrating reliable synchronization of angular and translational data. This indicates that the system can handle complex motion scenarios effectively.

The performance analysis underscores the limitations of using a single accelerometer for translational pose estimation, particularly in complex motion scenarios. Studies have shown that relying solely on accelerometer data can lead to significant pose estimation errors due to factors like sensor noise and drift. For instance, research on robot arm pose estimation indicates that using only one accelerometer per link results in higher estimation errors, while employing multiple accelerometers can improve accuracy (Wijayasinghe et al., 2018).

To enhance translational pose estimation accuracy, integrating additional sensors such as LiDAR is recommended. LiDAR sensors provide precise depth information, which, when fused with accelerometer data, can significantly improve pose estimation performance. This sensor fusion approach has been effectively applied in various applications, including human pose estimation (MDPI, 2024) and mobile robot navigation (MDPI, 2017).

Incorporating LiDAR into the system design can mitigate the limitations of single accelerometer-based pose estimation, leading to more accurate and reliable translational pose estimation in diverse operational environments.

- Camera Translational Accuracy Tests

The translational accuracy tests are conducted to evaluate the camera's capability to measure and track linear displacements along the x, y, and z axes. These tests are critical for verifying the performance of the camera in estimating translational motion within specified tolerances. By assessing controlled forward and backward movements, as well as static initial positions, the tests ensure the camera's reliability and precision under various conditions.

The tests along the x-axis begin with a static initial position to establish a baseline for horizontal measurements, ensuring the camera can provide accurate reference values (1). Following this, a controlled forward movement of 20 cm is applied to simulate typical horizontal motion scenarios such as object displacement or linear tracking (2). A subsequent backward motion of 20 cm assesses the camera's ability to track reverse movements and identify potential drifting or inaccuracies during directional changes (3). Finally, a longer forward motion of 40 cm evaluates the camera's precision over extended distances, ensuring reliability under increased displacement scenarios (4).

The y-axis tests also begin with a static initial position to verify baseline accuracy for lateral movements (5). A controlled lateral forward movement of 20 cm is then conducted, simulating sidestepping or lateral shifts (6). A 20 cm backward lateral motion tests the camera's ability to track reverse movements accurately, ensuring minimal drift in the opposite direction (7). To further assess precision, a 40 cm forward lateral motion is performed, challenging the system's accuracy over greater distances while maintaining reliability (8).

The z-axis tests start with a static initial position to confirm the system's baseline accuracy for vertical measurements (9). A controlled upward motion of 20 cm is performed to simulate elevation changes, such as climbing or lifting scenarios (10). This is followed by a downward motion of 20 cm to evaluate the camera's ability to measure descent accurately while considering the influence of gravity and potential drift (11). Finally, a 40 cm upward motion is conducted to test the system's performance over larger vertical displacements, ensuring precision in extended elevation scenarios (12).

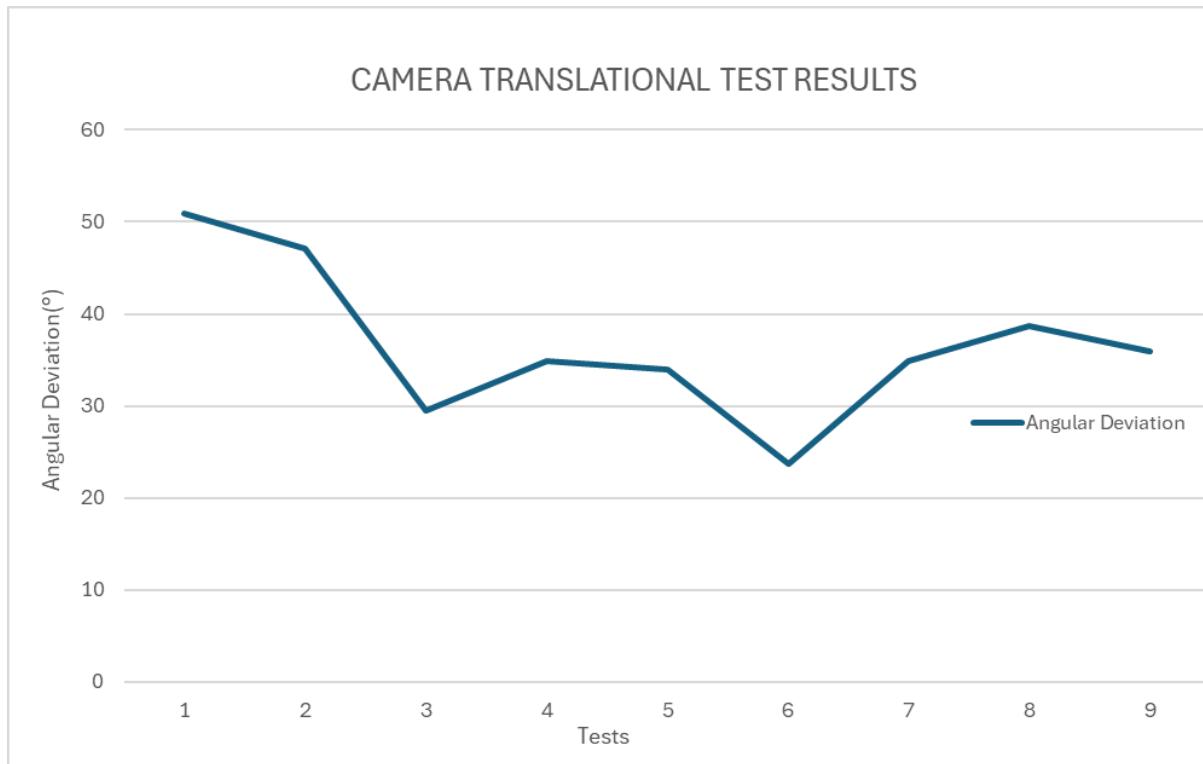


Figure 17.12 Camera Translational Test Results

In this test, we expect our deflection angle to be close to zero because the lower the deflection angle we get, the less margin of error there is in terms of distance. However, we got high deflection angles. For example, when moving along the X-axis, we expect our matrix to output values like [1, 0, 0], but if it returns values such as [0.63, -0.68, -0.35], this indicates a significant deviation. The angular deviation can be calculated as $\arccos(0.63)$. Hence, the camera's translational test failed.

- Camera Rotational Accuracy Tests

The rotational accuracy tests for the camera are conducted to evaluate its ability to track angular displacements along the roll, pitch, and yaw axes under controlled conditions. Each test simulates specific rotational movements, representing real-world scenarios, to assess the accuracy of the camera in capturing and processing angular changes. These conditions are designed to challenge the camera's performance, ensuring it meets predefined tolerances and operational requirements.

In the roll axis tests, the camera undergoes rotations starting from an initial position to -30° (1), simulating a steep downward tilt. This is followed by a recovery from -30° to 0° (2), representing a return to a neutral position. Subsequent rotations include incremental upward tilts, such as 0° to 45° (3) and 45° to 60° (4). These movements test the camera's ability to handle continuous angular transitions with minimal tracking errors or distortions.

For the yaw axis tests, the camera is rotated from an initial position to 45° (5), simulating extensive horizontal turns, as in scenarios like head tracking or large directional changes. Additional rotations include smoother transitions, such as -45° to -60° (6), -60° to 0° (7), and 0° to 90° (8), assessing the camera's ability to track angular movements at varied velocities and maintain directional accuracy.

The pitch axis tests involve rotations from an initial position to -30° (9), simulating a downward nod, and then returning from -30° back to the initial position (10). Further movements include upward tilts to 45° (11) and 60° (12), testing the camera's ability to measure vertical angular changes under conditions influenced significantly by gravity.

- Camera Test Results Discussion and Evaluation

For the camera subsystem, rotational tracking tests yielded accurate results, meeting the required $\pm 4^\circ$ error margin. However, translational tests along all axes revealed deviations exceeding the ± 6 cm tolerance in certain conditions, particularly in featureless environments. These results highlight the need for improved visual odometry algorithms and potential hardware upgrades, such as integrating an RGB-D camera for enhanced depth sensing. The camera subsystem successfully complied with frame rate and resolution requirements, aligning with system-level goals and industry standards for visual tracking systems.

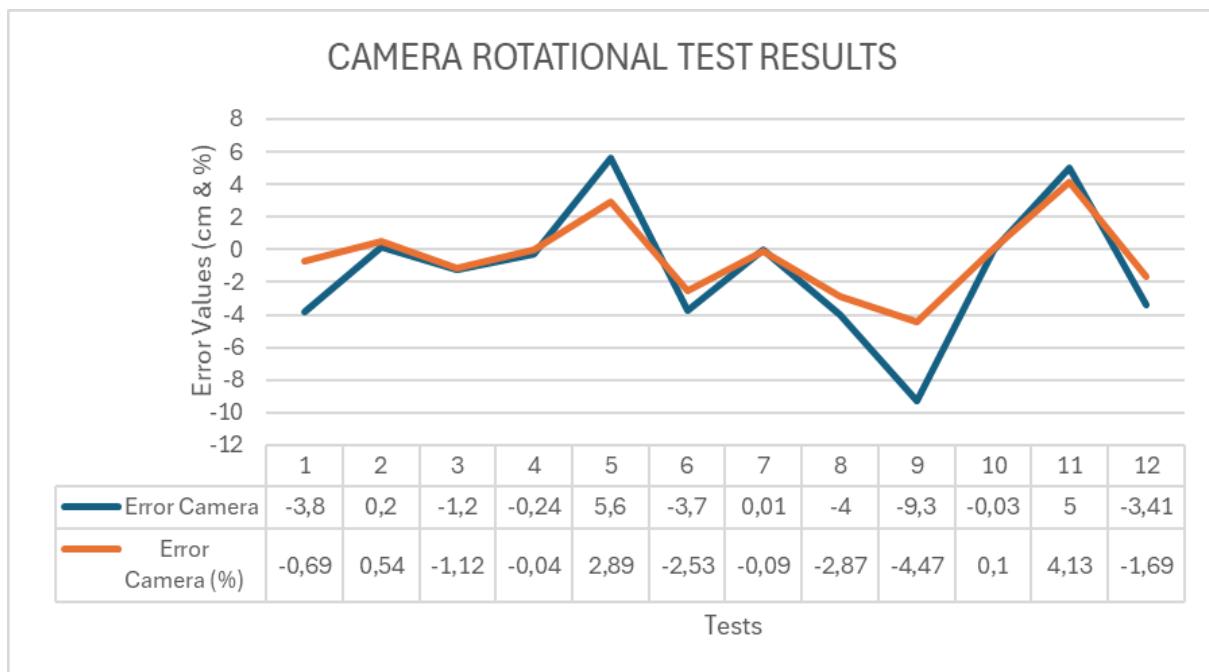


Figure 18. Camera Rotational Test Results

Planning

- Schedule

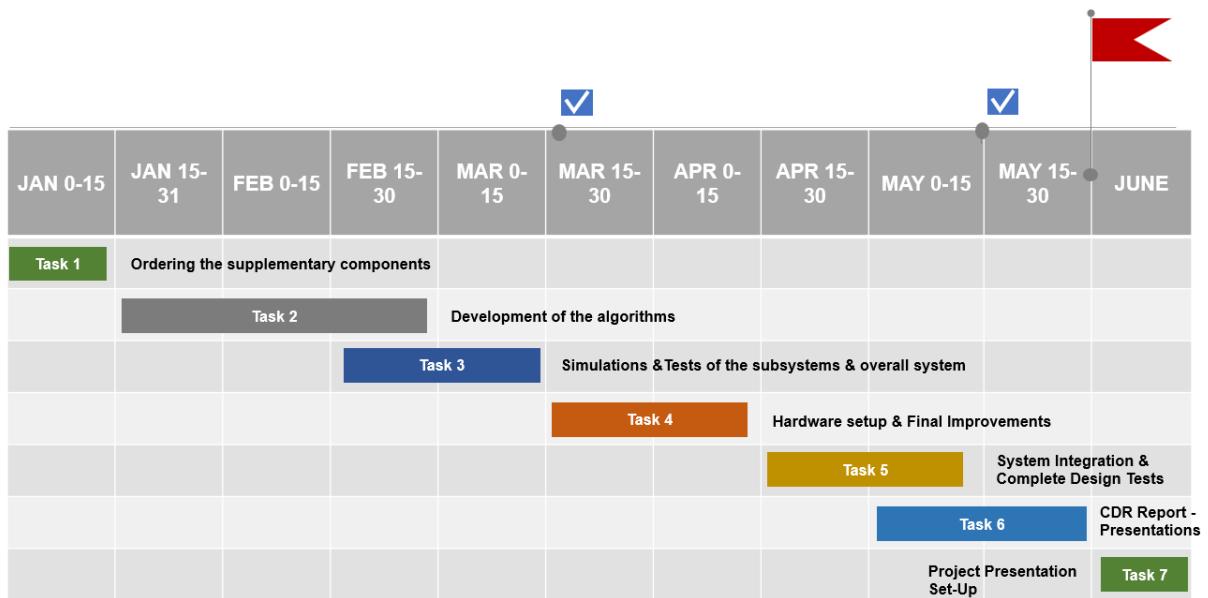


Figure 19. Gantt Chart

- **Ordering the Supplementary Components**

Ordering all necessary hardware components for the system will begin in early January and conclude by mid-January. This step ensures the timely acquisition of required materials. This task will be done by all group members.

- **Development of the Algorithms**

The development of algorithms, including the head pose estimation algorithm, will start in mid-January and continue through February. This task focuses on implementing and refining the 6-DOF head pose estimation using sensor fusion techniques. This task will be done by Enescan Çelebi, Hacer Ayça Yılmaz and Murathan Kutaniş.

- **Simulations and Tests of the Subsystems & Overall System**

Simulations and subsystem tests will commence in mid-February and proceed until early April. This phase includes evaluating the algorithms and system behavior in a controlled environment to ensure functionality and robustness. This task will be done by Hacer Ayça Yılmaz, Alperen Şahin and Öykü Özyurt.

- **Hardware Setup & Final Improvements**

Hardware setup, assembly, and any necessary final improvements will take place from early March to late April. This step ensures the system is fully functional and ready for integration testing. This task will be done by Sümeysa Arıcan, Alperen Şahin and Öykü Özyurt.

- **System Integration & Complete Design Tests**

Integration of the system's components and conducting comprehensive design tests will occur from late April to mid-May. This ensures that all features and functionalities work together seamlessly. This task will be done by Murathan Kutaniş, Enescan Çelebi and Sümeysa Arıcan.

- **CDR Report - Presentations**

The Critical Design Review (CDR) report and presentations will be completed by early June. This final stage involves documenting the system's development, preparing for evaluations, and concluding the project. This task will be done by all group members.

- **Project Presentation Setup**

Preparations for the project presentation will take place in the second half of May. This includes assembling the final demonstration set-up and ensuring all materials are ready for review. This task will be done by all group members.

- **Risk Analysis**

Assessing failure modes and potential mitigation measures for the Inside-Out Tracking Sensor Suite requires Failure Modes, Effects, and Criticality Analysis (FMECA). This approach not only evaluates the immediate ways the system could fail but also examines the interfaces and complex interactions between the various components, such as the IMU, camera, and processing unit. Building on the rationale for selecting the system architecture and components outlined in earlier sections, this section identifies potential failure modes within the proposed architecture and explains how the implemented mitigation measures ensure that the system's risk is maintained at an As Low as Reasonably Practicable (ALARP) level.

- **FMECA Methodology**

The FMECA method is utilized to identify hazardous conditions within the Inside-Out Tracking Sensor Suite and assign semi-quantitative risk levels to each failure mode associated with these hazards. It offers a structured approach to listing various potential

system failures, evaluating their associated risks, and identifying mitigation measures to reduce the likelihood of these failures. For this project, the FMECA method will support the ALARP justification for the tracking system as a critical component in ensuring safety and performance. Detailed explanations of the method are not the focus of this section; instead, the overall approach is summarized as follows:

1. Identify risks related to the system's design, architecture, functionality, and operation.
2. List possible failure modes and their underlying causes.
3. Assign semi-quantitative likelihood, severity, and risk levels to each failure mode, based on evaluations from subject matter experts or system designers. These values and their contribution to overall system criticality will follow pre-defined criteria similar to Table 7-1, Table 7-2, and Table 7-3.
4. Document consequences and propose mitigation measures for each failure mode.
5. Reassess the likelihood, severity, and risk levels after implementing mitigation measures to ensure that the system's overall risk is ALARP.

The semi-quantitative risk level is calculated as the product of likelihood and severity, ranging from 1 to 25. Risks are categorized as intolerable if the risk level is 15 or above, tolerable if between 5 and 12, and low if below 5. Mitigation measures must be implemented and justified for intolerable risks, while tolerable and low risks may not require extensive measures.

Table 7-1 6 Likelihood matrix showing likelihood levels

| LIKELIHOOD OF OCCURRENCE | | |
|-----------------------------|--|----------|
| Definition | Meaning | Value |
| Frequent | Likely to occur many times | 5 |
| Occasional | Likely to occur sometimes | 4 |
| Remote | Unlikely to occur but possible | 3 |
| Improbable | Very unlikely to occur | 2 |
| Extremely Improbable | Almost inconceivable that the event will occur | 1 |

Table 7-2 Severity matrix showing severity levels

| SEVERITY OF CONSEQUENCES | | |
|--------------------------|---|-------|
| Definition | Meaning | Value |
| Catastrophic | Results in a major accident, death or equipment destroyed | 5 |
| Hazardous | Serious injury or major equipment damage | 4 |
| Major | Serious incident or injury | 3 |
| Minor | Results in a minor incident | 2 |
| Negligible | Nuisance of little consequence | 1 |

Table 7-3 Risk matrix showing risk levels (red – intolerable, orange – tolerable, green- broadly acceptable)

| RISK LIKELIHOOD | RISK SEVERITY | | | | |
|------------------------|---------------|---------|---------|-------------|----------------|
| | NEGLIGIBLE 1 | MINOR 2 | MAJOR 3 | HAZARDOUS 4 | CATASTROPHIC 5 |
| FREQUENT 5 | | | | | |
| OCCASIONAL 4 | | | | | |
| REMOTE 3 | | | | | |
| IMPROBABLE 2 | | | | | |
| EXTREMELY IMPROBABLE 1 | | | | | |

The overall goal of the FMECA method in the context of the Inside-Out Tracking Sensor Suite is two-fold. First, it facilitates the identification of critical failure modes and risks associated with the system, enabling iterative improvements to the system design and the implementation of mitigation measures that enhance its functionality: to provide accurate, real-time head tracking and ensure seamless user interaction. Second, it serves as a robust foundation for building ALARP risk justification arguments that support the chosen design architecture of the tracking system.

1. Sensor Failures

1.1 Sensor Calibration Failures

Sensor Calibration Failure occurs when the IMU, camera, or both are not accurately calibrated, leading to misalignment between the sensor data and the real-world orientation or position. This failure can result in incorrect head pose estimations, severely affecting the functionality of the Inside-Out Tracking Sensor Suite.

Causes

- Improper Initial Setup: The calibration process during the first use is not user-friendly or fails to account for all possible variances.
- Physical Misalignment: Sensors are not properly positioned or securely mounted, causing drift or offset.
- Environmental Factors: Vibrations, thermal expansion, or mechanical shocks impact the sensor's baseline.
- Aging Components: Long-term use results in gradual sensor degradation, affecting calibration accuracy.

Consequences

- Tracking Errors: Misaligned sensor data leads to inaccurate 6-DOF head pose estimations.
- Reduced Immersion: Users experience a mismatch between physical movements and the VR environment, breaking immersion.
- System Instability: Cascading errors in sensor fusion algorithms due to inaccurate data input.
- Operational Downtime: Increased time spent recalibrating the system instead of seamless use.
- User Frustration: Negative user experiences from unreliable tracking.

Mitigation Measures

To address Sensor Calibration Failure, the following measures can be implemented:

- Periodic Recalibration:
 - Prompt users for recalibration at regular intervals or when the system detects significant drift.
 - Incorporate ambient condition sensors (e.g., temperature, vibration detectors) to trigger recalibration when necessary.

Table 8. Pre-and-Post Mitigation Likelihood, Severity and Criticality

| | Likelihood | Severity | Risk Level |
|-----------------|------------|----------|------------|
| Pre- Mitigation | 3 | 4 | 12 |
| Post-Mitigation | 2 | 3 | 6 |

1.2 Sensor Data Noise and Drift

Sensor Data Noise and Drift occur when the IMU (gyroscope and accelerometer) produces inaccurate or inconsistent readings due to noise or gradual deviation (drift) over time. This can compromise the accuracy of head pose estimations and affect the performance of the Inside-Out Tracking Sensor Suite.

Causes

- Environmental Vibrations: Mechanical vibrations from external sources introduce high-frequency noise into sensor readings.
- Thermal Effects: Changes in temperature affect the sensitivity and accuracy of IMU components.
- Aging Sensors: Long-term use or material degradation leads to gradual inaccuracies in sensor output.
- Algorithmic Limitations: Poorly tuned sensor fusion algorithms fail to filter out noise or correct drift effectively.
- Power Supply Fluctuations: Inconsistent power delivery impacts sensor stability.

Consequences

- Unstable Tracking: The system struggles to provide smooth, real-time head pose estimations due to noisy or drifting data.
- Loss of Immersion: Users perceive unnatural or jittery movements in the virtual environment.
- Cascading Errors: Drift in IMU readings propagates through the Kalman filter, reducing the accuracy of the fused data.
- Increased Latency: Algorithms may spend additional processing time trying to compensate for noise and drift.
- Maintenance Overhead: Frequent recalibration and adjustments are required to maintain system accuracy.

Mitigation Measures

- High-Quality Sensors:
 - Select IMUs with low noise and high stability, such as those with built-in drift compensation or advanced error correction features.
 - Ensure sensors meet the required sensitivity thresholds (e.g., $\pm 0.01^\circ/\text{s}$ gyroscope, $\pm 0.01 \text{ m/s}^2$ accelerometer).

Table 99. Pre-and-Post Mitigation Likelihood, Severity and Criticality

| | Likelihood | Severity | Risk Level |
|-----------------|------------|----------|------------|
| Pre- Mitigation | 4 | 4 | 16 |
| Post-Mitigation | 2 | 3 | 6 |

1.3 Data Fusion Latency

Data Fusion Latency occurs when the Kalman filter or other fusion algorithms fail to process sensor data (IMU and camera inputs) within the required time frame. This results in delayed or laggy head pose estimations, which degrade the performance of the Inside-Out Tracking Sensor Suite.

Causes

- Inefficient Algorithms: Sensor fusion algorithms are not optimized, leading to longer processing times.
- Hardware Limitations: The processing unit lacks sufficient computational power to handle high-frequency data input.
- Asynchronous Sensor Data: Misaligned timestamps between IMU and camera data increase processing complexity.
- Large Data Payloads: High-resolution camera streams and frequent IMU updates overload the processing pipeline.
- Real-Time Communication Delays: Delays in transmitting data between sensors and the processing unit exacerbate latency.

Consequences

- Reduced Responsiveness: Users experience a noticeable lag between their physical movements and corresponding changes in the virtual environment.
- Loss of Immersion: Delays in rendering head movements break the continuity of the VR experience.
- Tracking Errors: Slow data fusion increases the likelihood of outdated or mismatched sensor data being used, reducing tracking accuracy.
- System Instability: High latency may cause cascading failures in real-time operations, such as rendering and collision detection.
- User Dissatisfaction: Laggy performance negatively affects user satisfaction and system usability.

Mitigation Measures

- **Reduced Sampling Rates:**
 - Optimize sensor sampling rates to balance data frequency and processing load without compromising accuracy.
 - Perform sensitivity analysis to determine the minimum sampling rate required for effective tracking.

Table 10.10 Pre-and-Post Mitigation Likelihood, Severity and Criticality

| | Likelihood | Severity | Risk Level |
|-----------------|------------|----------|------------|
| Pre- Mitigation | 3 | 5 | 15 |
| Post-Mitigation | 2 | 4 | 8 |

1.4 Wireless Communication Failure

Wireless Communication Failure occurs when the connection between the VR headset and the external computer is disrupted, resulting in data loss, latency spikes, or reduced communication quality. This failure can significantly affect real-time rendering and tracking in the Inside-Out Tracking Sensor Suite.

Causes

- Signal Interference: Nearby electronic devices or crowded wireless networks cause interruptions in data transmission.
- Hardware Limitations: Inadequate wireless modules in the headset or external computer result in unreliable connections.
- Range Issues: Operating beyond the effective range of the wireless network weakens signal strength.

Table 11.11 Pre-and-Post Mitigation Likelihood, Severity and Criticality

| | Likelihood | Severity | Risk Level |
|-----------------|------------|----------|------------|
| Pre- Mitigation | 3 | 3 | 9 |
| Post-Mitigation | 2 | 2 | 4 |

Mitigation Measures

- Data Compression and Prioritization:
 - Compress sensor data before transmission to reduce payload size and alleviate bandwidth limitations.
 - Prioritize critical data packets (e.g., head pose updates) over less time-sensitive information.

1.5 System Overload

System Overload occurs when the processing unit of the Inside-Out Tracking Sensor Suite is unable to handle the computational demands of sensor data acquisition, fusion, and rendering in real-time. This leads to system slowdowns, crashes, or degraded performance.

Causes

- Insufficient Hardware Resources: The processing unit lacks adequate CPU, GPU, or memory capacity to manage high-frequency data processing tasks.
- Unoptimized Software: Inefficient algorithms or poorly structured code increase processing times and resource consumption.
- Excessive Data Throughput: High-resolution camera streams and frequent IMU updates overwhelm the system's processing capacity.
- Background Processes: Non-essential tasks running on the processing unit consume resources needed for critical functions.
- Improper Load Distribution: Lack of parallel processing or hardware acceleration leads to unevenly distributed computational loads.

Consequences

- Tracking Delays: The system cannot process sensor data in real-time, leading to inaccurate or laggy head pose estimations.
- System Crashes: Overloaded hardware may fail entirely, requiring a restart and interrupting operation.
- Reduced Immersion: Performance issues negatively impact the user's experience in the virtual environment.
- Increased Latency: Overload conditions exacerbate delays in data fusion and rendering.
- Shortened Hardware Lifespan: Prolonged overuse of hardware components causes overheating and wear, leading to premature failure.

Mitigation Measures

- Software Optimization:
 - Optimize algorithms for efficiency, reducing their computational demands while maintaining accuracy.
 - Implement multi-threading or parallel processing to distribute workloads across multiple cores.
 - Use memory-efficient data structures to minimize memory usage.

Table 12.12 Pre-and-Post Mitigation Likelihood, Severity and Criticality

| | Likelihood | Severity | Risk Level |
|-----------------|------------|----------|------------|
| Pre- Mitigation | 4 | 5 | 20 |
| Post-Mitigation | 2 | 4 | 8 |

1.6 Environment Adaptability

Environment Adaptability Failure occurs when the Inside-Out Tracking Sensor Suite struggles to maintain consistent tracking performance in variable environments, such as low-light conditions, dynamic lighting changes, or featureless surroundings. This can degrade tracking accuracy and usability.

Causes

- Low-Light Conditions: Insufficient ambient lighting reduces the camera's ability to detect features for tracking.
- Dynamic Lighting Changes: Sudden or frequent changes in lighting conditions disrupt the consistency of feature detection algorithms.
- Featureless Environments: Sparse visual features in the environment (e.g., blank walls or open spaces) limit the camera's ability to perform effective tracking.
- Reflective or Transparent Surfaces: Reflective or transparent objects confuse the feature detection algorithms, leading to inaccurate tracking.
- Suboptimal Camera Hardware: Cameras with low sensitivity or poor resolution struggle in challenging environments.

Consequences

- Tracking Inaccuracy: The system fails to maintain precise head pose estimations, leading to errors in virtual representation.
- User Dissatisfaction: Poor tracking performance frustrates users, reducing the system's usability and reliability.
- Loss of Immersion: Users experience discontinuities in the VR environment, breaking immersion.
- Operational Downtime: Frequent recalibration or adjustments may be needed to maintain functionality in challenging environments.
- Limited Application Scenarios: The system becomes less suitable for environments like dimly lit rooms or featureless spaces.

Mitigation Measures

- Hardware Improvement:
 - Equip the system with stereo cameras or RGB-D cameras to improve depth perception and robustness in featureless settings.
 - Include cameras with wider fields of view to capture more environmental features.

Table 13. 13Pre-and-Post Mitigation Likelihood, Severity and Criticality

| | Likelihood | Severity | Risk Level |
|-----------------|------------|----------|------------|
| Pre- Mitigation | 3 | 4 | 12 |
| Post-Mitigation | 2 | 3 | 6 |

1.7 Component Costs

Cost Overrun due to Component Costs occurs when the high-quality cameras, IMUs, and computing units required for the Inside-Out Tracking Sensor Suite exceed the allocated budget of \$300, particularly if additional hardware upgrades are needed to meet performance requirements.

Causes

- High-End Components: Advanced sensors and processing units required for optimal performance may have higher market prices.
- Unexpected Price Increases: Market fluctuations or supply chain issues lead to higher component costs than anticipated.
- Underestimated Budgeting: Initial cost estimates did not account for all components or overlooked hidden costs (e.g., shipping, taxes).
- Additional Hardware Needs: Unplanned upgrades or replacements are required to meet performance or compatibility demands.
- Limited Suppliers: Restricted supplier options for specialized components result in limited negotiation power for pricing.

Consequences

- Budget Overrun: Exceeding the \$300 budget impacts project financial feasibility.
- Delayed Development: Additional time is required to secure funding or source cost-effective components.
- Compromised Performance: Substituting lower-quality components to stay within budget reduces system effectiveness.
- Increased Risk of Failures: Cost-cutting measures lead to reliance on components with lower reliability or shorter lifespans.

Mitigation Measures

- Bulk Purchases and Supplier Negotiations:
 - Purchase components in bulk where feasible to benefit from volume discounts.
 - Negotiate with suppliers for academic or research-based pricing discounts.

Table 14.14 Pre-and-Post Mitigation Likelihood, Severity and Criticality

| | Likelihood | Severity | Risk Level |
|-----------------|------------|----------|------------|
| Pre- Mitigation | 4 | 4 | 16 |
| Post-Mitigation | 2 | 3 | 10 |

1.8 Regulatory Standards

Non-Compliance with Regulatory Standards occurs when the Inside-Out Tracking Sensor Suite fails to meet ergonomic and electronic device standards required for wearable systems. This can result in additional development time, cost increases, or legal and operational restrictions.

Causes

- Unfamiliarity with Standards: Lack of detailed knowledge about specific ergonomic or electronic compliance standards for wearable systems.
- Iterative Design Adjustments: Multiple iterations are required to meet standards, delaying project completion.
- Hardware Limitations: Components used in the system do not inherently meet compliance standards, requiring replacements or redesigns.

Consequences

- Delays in Development: Additional time is required for redesign, testing, and certification to achieve compliance.
- Increased Costs: Iterative adjustments, re-certifications, and component replacements inflate project expenses.
- Restricted Use: Non-compliance may prevent the system from being legally deployed in certain environments or markets.
- Reputational Damage: Failure to meet regulatory standards undermines stakeholder confidence in the project.
- Reduced Usability: Non-compliance with ergonomic standards impacts user comfort and system adoption.

Mitigation Measures

- Early Standard Identification:
 - Identify applicable standards early in the project, including those related to ergonomics (e.g., ISO 9241) and electronic devices (e.g., FCC, CE certifications).
 - Consult with regulatory experts to ensure all relevant requirements are accounted for.

Table 15.15 Pre-and-Post Mitigation Likelihood, Severity and Criticality

| | Likelihood | Severity | Risk Level |
|-----------------|------------|----------|------------|
| Pre- Mitigation | 3 | 4 | 12 |
| Post-Mitigation | 2 | 3 | 6 |

1.9 Timeline Risk

Timeline Risks occur when the project fails to meet its development deadlines due to unforeseen delays in design, procurement, testing, or deployment. This can impact the delivery of the Inside-Out Tracking Sensor Suite within the planned timeframe.

Causes

- Underestimated Complexity: Certain tasks, such as sensor fusion implementation or calibration processes, take longer than initially anticipated.
- Procurement Delays: Late delivery of components (e.g., cameras, IMUs, processors) due to supply chain issues.

- Technical Challenges: Unanticipated technical problems, such as software bugs or hardware incompatibility, require additional time to resolve.
- Resource Constraints: Limited team availability or expertise causes bottlenecks in critical development phases.
- Inadequate Planning: Poorly defined milestones or unrealistic timelines result in missed deadlines.

Consequences

- Project Delays: The overall project timeline is extended, potentially missing key delivery or deployment deadlines.
- Increased Costs: Additional time translates into higher resource utilization and potential budget overruns.
- Incomplete Features: Features may need to be deprioritized or omitted to meet the final delivery date.

Mitigation Measures

- Parallel Development
 - Identify independent tasks that can be worked on simultaneously to accelerate overall progress.
 - Assign team members to parallel tracks to ensure no critical path bottlenecks occur.

1.10 User Factor

User Factor Risks occur when the system design does not adequately account for user needs, behaviors, or limitations. This can lead to usability issues, discomfort, or improper use of the Inside-Out Tracking Sensor Suite, reducing its effectiveness and adoption.

Table 16. 16Pre-and-Post Mitigation Likelihood, Severity and Criticality

| | Likelihood | Severity | Risk Level |
|-----------------|------------|----------|------------|
| Pre- Mitigation | 4 | 3 | 12 |
| Post-Mitigation | 2 | 3 | 6 |

Causes

- Complex User Interface (UI): The system requires excessive user input or involves unintuitive setup and operation.
- Ergonomic Oversights: The system is uncomfortable or impractical for extended use, such as being too heavy, poorly balanced, or restrictive.

- Lack of User Training: Users do not receive adequate instructions or guidance, leading to misuse or suboptimal performance.
- Diverse User Profiles: Variability in user physical characteristics (e.g., head size, fit preferences) and skill levels is not adequately considered.
- Unclear Feedback Mechanisms: Users are not provided with clear feedback about the system's status, errors, or calibration.

Consequences

- Reduced Usability: Users struggle to operate the system effectively, diminishing its perceived value and functionality.
- Discomfort and Fatigue: Extended use becomes physically taxing, discouraging adoption and repeated use.
- Improper Use: Incorrect operation due to insufficient training or unclear instructions leads to performance issues or tracking errors.
- Negative User Experience: Frustration or dissatisfaction reduces user trust and confidence in the system.
- Limited Adoption: Poor user experience limits the system's appeal and application in broader markets.

Mitigation Measures

- Comprehensive User Training:
 - Provide a detailed user manual and quick start guide with visual aids for easy understanding.
 - Include video tutorials and interactive demos to help users learn the system efficiently.

Table 17.17 Pre-and-Post Mitigation Likelihood, Severity and Criticality

| | Likelihood | Severity | Risk Level |
|-----------------|------------|----------|------------|
| Pre- Mitigation | 4 | 4 | 16 |
| Post-Mitigation | 2 | 3 | 6 |

A risk assessment of the entire Inside-Out Tracking Sensor Suite was performed for a high-level consideration of the system's design and functionality, with a detailed approach to how specific components or systems of components could fail. The risks, failure modes, and mitigation measures outlined previously are used to inform the development of the Verification and Validation (V&V) plan for the tracking system, as well as the assembly, integration, and testing (AIT) process and eventual deployment.

These considerations are aligned with the operational concept of the system to ensure that the identified risks are maintained at ALARP levels throughout its lifecycle, including design, testing, and usage. Summarizing the entire FMECA process for the Inside-Out Tracking Sensor Suite, the tally of identified risks, metrics, and mitigation measures is detailed in Table [18].

Table 18.18 Summary of total risks, failure modes and risk scores prior to and post-mitigation measures

| | Total Risk | Total Failure Modes | High Risk Pre-Mitigation | High Risk Post-Mitigation | Medium Risk Pre-Mitigation | Medium Risk Post-Mitigation | Low Risk Post-Mitigation | Low Risk Post-Mitigation |
|---------------|------------|---------------------|--------------------------|---------------------------|----------------------------|-----------------------------|--------------------------|--------------------------|
| Entire System | 5 | 10 | 6 | 0 | 5 | 9 | 0 | 1 |

The upcoming plans for our team members for the remained parts are determined as below:

| | |
|-------------------|---|
| Sümeyra Arıcan | Communication channel establishment between the Raspberry Pi and computer |
| Alperen Şahin | Testing of the camera algorithm / Hardware setup |
| Enescan Çelebi | Sensor Fusion / Application of the camera algorithm |
| Öykü Özyurt | Establishing Power Unit / Hardware Connections |
| Murathan Kutaniş | IMU – LIDAR Integration / System Harness |
| Hacer Ayça Yılmaz | Head model rendering / Improvement of the camera algorithm |

- Sensor Fuse Algorithms

In order to obtain final head pose estimation which will be used in the VR applications, estimations found by inertial and visual sensor suites will be fused. The primary approach for the sensor fusion is to use Kalman filter template with measurements updates for differentiated for each sensor suite. This approach will rely on the suite's determined covariance values and offer optimal solution for the estimation. But this approach will not be applicable for most cases since the sensor suites rely on each other to produce robust estimations.

Visual sensor suite lacks the scale of the translational motion, while inertial sensors are prone to translational drift. So, in order to generate robust estimation, the inertial sensor's stationarity condition will be supplied by the camera and translational scale will be determined by the inertial sensors. Visual translational direction estimation will be fused with inertial translation's normalized form.

In rotational motion both sensors can produce estimations on their themselves so their estimations might be fused with Kalman filter. But since the current inertial rotation algorithms do not calculate the covariance, constant predefined tuned covariance will be used instead.

- Power Systems

This subsystem is responsible for both generating power and distributing it to others.

subsystems. It includes a battery and power cables. The power distribution diagram of the project is illustrated in Figure 20.

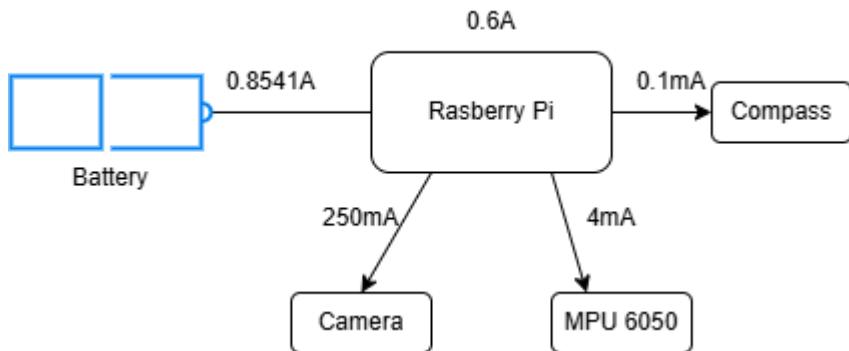


Figure 20. The Power Distribution Diagram

Battery

The battery powering the whole system is planned to be 20W 10.000 mAh power supply. We have chosen a 30.000 mAh power supply to have continuous operation with the system. The capacity of the battery is chosen to be 10.000 mAh meaning that it can hold up to 1 hour if 10A current is drawn continuously from the battery. The whole system will get the power from the same source. We choose power supply as a power bank to be compact with our design.

$$\frac{10000mAh}{1000} = 10Ah$$

$$\frac{10Ah}{0.8541A} = 11.7h$$

By the calculation above with having the full charge on our battery the system can be run almost 12 hours which is much greater than what we need to be. The typical current rating of individual elements is in the table. The total current will be supplied to the Raspberry pi module.

Table 19. Current Ratings

| Component | Component Typical Current |
|-----------------|---------------------------|
| Raspberry pi 5 | 0.6A |
| Camera Module 3 | 0.25A |
| MPU 6050 | 0.004A |
| GY 271 | 0.0001A |
| Total: 0.8541A | |

By default, the raspberry pi current is 600mA to ensure that we have sufficient margin to support these workloads. However, if the Raspberry Pi 5 firmware detects this supply, it increases the USB current limit to 1.6A, providing 5W of extra power for downstream USB devices and 5W of extra on-board power budget.

Cable

Since the raspberry pi will be our main power distributor among the sensors and camera, cabling is between power supply and the Raspberry pi with Type-C USB cable. The internal power distribution will happen with the jumpers.

- Communication

The communication system in this VR project is designed to seamlessly integrate wired communication between the Raspberry Pi 5 and the connected sensors, while using wireless communication to interface with the external computer. This dual-layer communication ensures efficient, real-time data transfer and processing, crucial for the system's performance.

Wired Communication

The Raspberry Pi 5 is equipped with various sensors that communicate through wired interfaces. Each sensor utilizes a specific protocol to ensure reliable data transfer.

The Camera connects via the Camera Serial Interface (CSI) using the MIPI CSI-2 protocol. This interface is chosen for its high-bandwidth capabilities, which allow it to handle video data efficiently. CSI ensures minimal latency, making it ideal for real-time applications.

The MPU6050 (IMU) and GY-271 (Magnetometer) use the I2C protocol for communication. The I2C bus is a simple and effective method for integrating multiple sensors with the Raspberry Pi, as it allows data from both devices to be transmitted over shared clock and data lines. These sensors provide essential motion and orientation data, contributing to the system's accuracy.

If integrated, the LiDAR sensor will connect via UART or I2C, depending on the specific model. LiDAR provides in-depth information for the environment, enhancing the system's ability to detect and interpret spatial data. The choice of interface will prioritize reliability and real-time performance.

Wireless Communication

Wireless communication is achieved using the Raspberry Pi's built-in 5 GHz Wi-Fi. This connection bridges the Raspberry Pi with an external computer, facilitating the transfer of pose estimation data and other system outputs. The 5 GHz frequency band is selected for its higher bandwidth and reduced interference compared to the 2.4 GHz band, which is often congested.

The data transmission relies on standard network protocols. TCP/IP ensures reliable data delivery, crucial for control commands and system-critical updates. For scenarios where speed takes precedence over reliability, such as streaming data, UDP may be utilized to minimize latency. This flexible protocol choice allows the system to balance accuracy and performance based on operational requirements.

The wireless connection is protected using WPA3 encryption, providing a robust defense against unauthorized access. Additionally, secure socket layers (SSL) are implemented for TCP/IP communication, safeguarding data during transmission.

- Adding A Lidar Sensor

Until this point, our design successfully met the MPU6050's rotational performance requirements with high scores. However, the translational accuracy of IMU still requires improvement as it currently exceeds the 10% error threshold that we established. To address this problem, particularly the accelerometer's performance in fast-paced motion detection scenarios, we plan to integrate LIDAR (Laser Imaging Detection and Ranging) in

the second term of the year. LIDAR will provide a 3D map of the environment and solve the accumulated error issue. Unlike accelerometers, LiDAR can directly measure distances to surrounding objects with high accuracy and repeatability, making it particularly suitable for translation calculations. It provides reliable and real-time spatial data, enabling precise tracking of movement without being affected by cumulative drift or noise. Furthermore, it will mitigate any biasing issue occurring in the IMU measurements, such as signaling IMU to reset accelerometer biases when LIDAR detects no movement. Overall, it will provide one more reliable sensor to the sensor fusion which is desirable for us as the number of reliable sensors will be increased and the camera motion estimation will rely on more positional data.

Collision Avoidance feature is one of the optional applications we want to integrate in our project. The implementation of LIDAR will be used in this part. As it measures the distance the laser light is transmitted and reflected to the source, we will have a map of objects around the VR user. Additionally, the wide view advantage of LIDAR will provide information from many directions. In case of any object trajectory directed towards the user, LIDAR will detect it in advance.

- **Camera**

Camera subsystem requires improvements due to the lack of accuracy in the camera pose estimation. It is concluded that although the optical flow algorithm and feature detection works well, the extraction of the rotation and translation matrices do not satisfy the requirements yet alone provide tricky estimations. As the issue is considered to stem from the recoverpose(.) function of OpenCV we use in the overall code [Appendix 1]. SFM Module will be implemented to solve/improve the camera pose estimation.

- **OpenCV SFM Module**

The module is compatible with our algorithm as it processes the previous and current frames of a streaming video. In addition to computing rotation and translation matrices, SFM module also creates 3D point clouds from multiple views. Moreover, the module itself uses an iterative optimization technique to find/refine camera intrinsics matrix. This will enable us to reach more precise estimations as we manually derived the intrinsic matrix K in our code and may cause inaccuracies in the rotation and translation matrices.

SFM module also provides its own visualization tool with viz module. Compared to our visualization method, which consists of a cube and reflects the yaw, pitch and roll movements according to the rotation matric, the module can draw a line to indicate the translational movement and insert a camera perspective to show the direction of the camera.

Another improvement of the module is that it optimizes the entire set of frames globally. By simultaneously processing the 3D positions and contributing constraints of the consecutive frames to the spatial localization, the error can be minimized as well as the noise coming from oscillatory motions can be mostly eliminated. To successfully implement the module, feature points should be well-tracked, and the scaling should be done as the 3D points are defined in projective space nor in real world metric space. The problems introduced by this module will be solved in the following ways:

- 1- RANSAC
- 2- IMU and LIDAR data

Our current code uses RANSAC and eliminates almost all the loosely estimated feature points; in other words, badly detected points are not entered in the computation of the



camera pose. Furthermore, scaling will be implemented to the normalized 3D points and translation matrix by using the data provided by IMU. In fact, LIDAR will enhance the scaling ratio accuracy as it will transfer the depth information of the place.

- **Harness**

The planned harness design aims to ensure a reliable, modular, and efficient wiring system for the integration of the MPU6050 IMU, camera, and HMC5883L magnetometer, LIDAR. Currently, harnesses for these components have been developed, focusing on robust communication and power delivery while adhering to high engineering standards. Each harness is designed to provide secure connections, reduce noise, and allow flexibility for future system upgrades.

Although the product is not intended for military use, the harnesses are being developed in compliance to military-grade standards, such as MIL-STD-461 for electromagnetic compatibility and MIL-STD-810 for environmental robustness. This ensures that the system can perform reliably under a wide range of operational conditions, including those involving vibration, thermal stress, and potential electromagnetic interference. By adopting these standards, the design achieves a higher level of durability and reliability, aligning with industry best practices for critical systems.

The harnesses for the MPU6050 and HMC5883L use shielded I2C lines to minimize interference, while the camera harness includes differential signaling for video transmission, ensuring high-quality image data. Grounding strategies and secure connector housings are employed to maintain signal integrity and simplify maintenance. The current focus on modularity and adherence to high standards positions the system for potential use in demanding environments while remaining cost-effective for non-military applications.

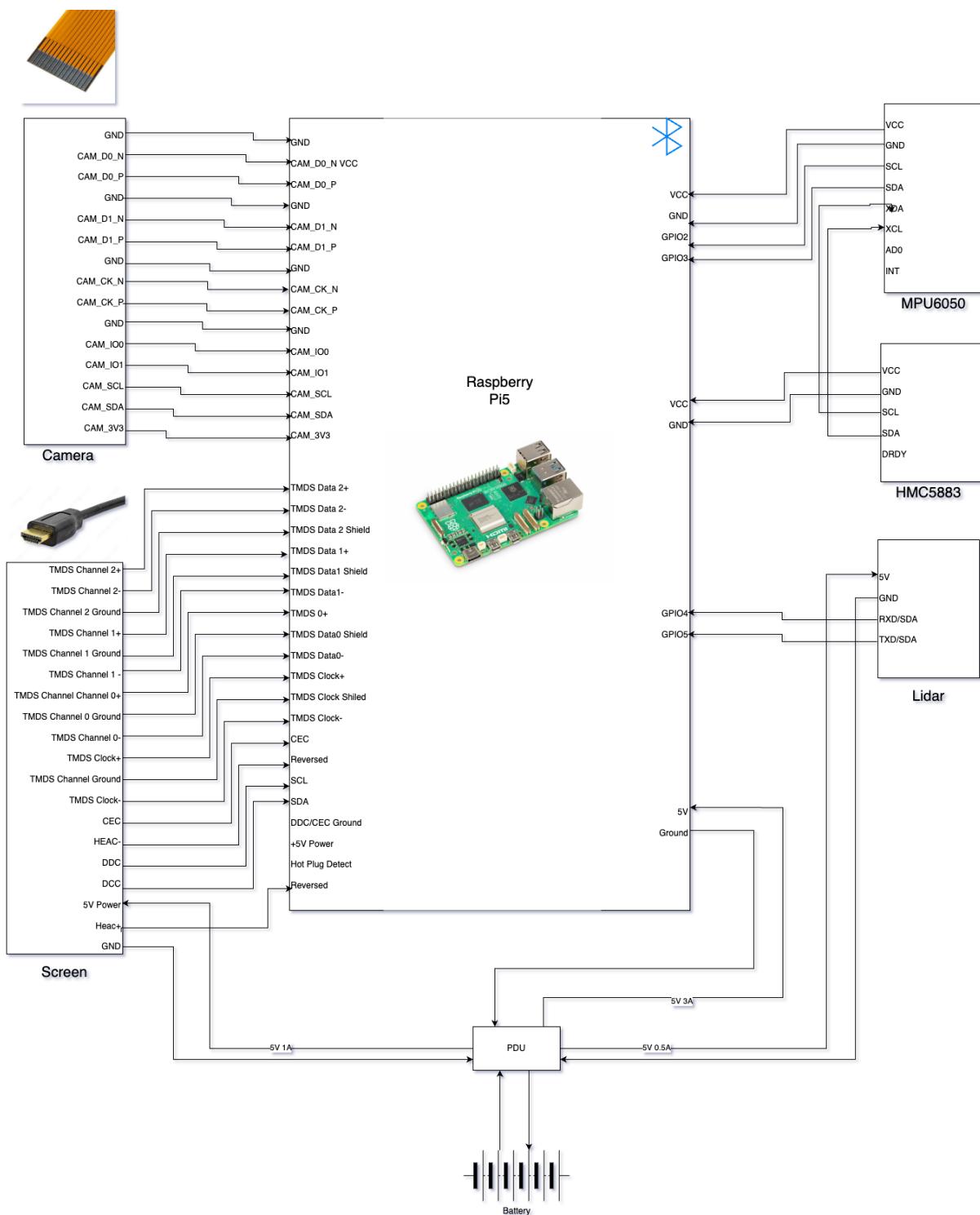


Figure 21.13 Harness Plan

- Deliverables and Cost Analysis

Table 20. Deliverables

| Material | Order Date | Delivery Date | Count | Unit Price |
|--------------------------|------------|---------------|-------|-------------|
| Raspberry Pi 5 8gb | 20.11.2024 | 22.11.2024 | 1 | 3443 TL |
| IMU | 20.11.2024 | 22.11.2024 | 2 | 98 TL |
| Raspberry Camera | 20.11.2024 | 22.11.2024 | 1 | 1521 TL |
| 128 GB SD Card | Murathan | 26.11.2024 | 1 | 199 TL |
| Cable Mouse | Murathan | 26.11.2024 | 1 | Outside BOM |
| Cable Keyboard | Ayça | 27.11.2024 | 1 | Outside BOM |
| Micro HDMI to HDMI Cable | Murathan | 26.11.2024 | 1 | Outside BOM |
| Mini Fan | Ayça | 27.11.2024 | 1 | 98 TL |
| GY-271 Magnetometer | 15.12.2024 | 18.12.2024 | 1 | 27 TL |

- **Total Cost:**

- Raspberry Pi: 3443 TL
- IMU: $98 \text{ TL} \times 2 = 196 \text{ TL}$
- Raspberry Camera: 1521 TL
- 128 GB SD Card: 199 TL
- Mini Fan: 98 TL
- GY-271 Magnetometer: 27 TL
- **LiDAR Sensor:** 1500 TL

- **Total (excluding "Outside BOM"):** $3443 + 196 + 1521 + 199 + 98 + 27 + 1500 = \mathbf{6984 \text{ TL}}$

The inclusion of the LiDAR sensor increases the total cost of the system to **6984 TL**, staying well within the \$300 budget when converted (based on the current TL/USD exchange rate, approximately).

- Ethical Consideration in System Design

The development of the Inside-Out Tracking Sensor Suite emphasizes adherence to ethical engineering practices to ensure user safety, accessibility, and environmental sustainability. The system is designed to comply with ergonomic and electronic standards, such as ISO 9241 and FCC/CE certifications, ensuring user comfort and minimizing health risks during prolonged use. Additionally, the project prioritizes responsible sourcing of components, aiming to reduce environmental impact through efficient material use and waste management. By fostering inclusivity, the design accommodates diverse user profiles, ensuring accessibility for individuals with varying physical characteristics and skill levels. This commitment reflects the team's dedication to ethical responsibility in delivering a user-focused and environmentally conscious product.

Test Plans

As the IMU has high performance out of the tests overall, test procedures will not be done to IMU and rather to LIDAR, since the error of IMU is decided to be complemented using LIDAR.

- Component Tests

1- Camera

➤ Translational Accuracy Test

- Objective: Verify camera tracking for linear movements.
- Steps:
 - Place markers at 0cm, 20cm, and 40cm along x, y, and z axes.
 - Move the camera through these points and log the positions detected by the updated algorithm.
 - Ensure the camera is aligned with the markers during movement.
- Performance Metric:
 - **Error in position tracking:** <±3cm deviation from actual distances.

➤ Rotational Accuracy Test

- Objective: Evaluate camera tracking for angular movements.
- Steps:
 - Mount the camera on a rotation fixture.
 - Rotate to -30°, 0°, 45°, and 30° for yaw, pitch, and roll axes, pausing for 5 seconds at each angle.

- Log and compare detected angles with actual values.
- Performance Metric:
 - **Error in angle detection:** $<\pm 3^\circ$ deviation.
- **Combined Translational and Rotational Test**
 - Objective: Assess the camera's tracking performance under combined movements.
 - Steps:
 - Create a path with predefined translational and rotational critical points.
 - Record detected position and orientation at each point.
 - Performance Metric:
 - **Combined position and angle error:** $<\pm 3\text{cm}$ for position and $<\pm 3^\circ$ for angles.

2- LIDAR

- **Range Accuracy Test**
 - Objective: Verify LIDAR's ability to measure distances accurately.
 - Steps:
 - Place objects at 0.3m, 0.5m, 1m, and 2m.
 - Record LIDAR-measured distances for each object.
 - Performance Metric:
 - **Error in distance measurement:** $<\pm 1\text{cm}$.
- **Field of View Test**
 - Objective: Ensure LIDAR covers the specified detection area.
 - Steps:
 - Rotate the LIDAR to scan a 170° area.
 - Check detection consistency within the specified range and resolution.
 - Performance Metric:
 - **Detection range:** 170° coverage with $<2\%$ missed points.
- **Obstacle Detection Test**

- Objective: Evaluate LiDAR's detection capabilities for static and dynamic obstacles.
- Steps:
 - Place multiple static and moving objects at random positions.
 - Log detected distances and relative positions.
- Performance Metric:
 - **Error in obstacle location:** <±2cm.

- Subsystem Tests

1- Functional System Test

- Objective: Verify that all system components work together as expected.
- Steps:
 1. Initialize and power on all components (IMU, camera, LiDAR, Raspberry Pi).
 2. Test basic functionality:
 - IMU: Verify accelerometer and gyroscope data is correctly received.
 - Camera: Ensure video stream is captured.
 - LiDAR: Check point cloud data.
 3. Run the sensor fusion algorithm and verify that combined data produces accurate pose estimation.
 4. Test communication between Raspberry Pi and computer for real-time head model rendering.

2- Sensor Fusion

➤ Data Synchronization Test

- Objective: Verify accurate timestamping and data synchronization between sensors.
- Steps:
 - Simultaneously collect data from all sensors during predefined movements.
 - Check if timestamps align and data overlap accurately.
- Performance Metric:

- **Synchronization delay:** <5ms.

➤ **Fusion Accuracy Test**

- Objective: Assess accuracy of fused position and orientation data.
- Steps:
 - Perform a combined translational and rotational test using all sensors.
 - Compare fused output with ground truth.
- Performance Metric:
 - **Error in fused data:** < \pm 1.5cm for position and < \pm 1° for angles.

➤ **Robustness Test**

- Objective: Test sensor fusion under challenging conditions (e.g., vibrations, occlusions).
- Steps:
 - Introducing noise and dynamic movements.
 - Record fusion system stability and output consistency.
- Performance Metric:
 - **Deviation under noise:** < \pm 10% from ideal values.

3. Real-Time Performance Test

- Objective: Ensure the system performs in real-time under expected conditions.
- Steps:
 - Measure the real-time processing speed of the Raspberry Pi when capturing data from all sensors.
 - Record latency in communication between Raspberry Pi and the computer during real-time rendering.
 - Evaluate the stability of the sensor fusion algorithm in continuous operation.
 - Monitor the power consumption of the Raspberry Pi and ensure the power unit provides enough energy.

4- Communication Protocol

➤ Latency Test

- Objective: Ensure low-latency communication between Raspberry Pi and computer.
- Steps:
 - Send test packets from Raspberry Pi to computer and measure response time.
- Performance Metric:
 - **Round-trip latency:** <50ms.

➤ Data Integrity Test

- Objective: Validate accurate transmission of pose data.
- Steps:
 - Send a set of predefined pose data and verify received data on the computer.
- Performance Metric:
 - **Error in transmitted data:** <0.1%

➤ Real-Time Rendering Test

- Objective: Verify real-time pose updates of the head model on the computer.
- Steps:
 - Rotate and translate the Raspberry Pi setup.
 - Observe rendering of the head model.
- Performance Metric:
 - **Update frequency:** >30Hz.

5-Environmental Stress Test

- Objective: Test the system's performance under various environmental conditions (e.g., temperature, lighting).

- Steps:
 - Perform tests in different lighting conditions to evaluate the camera's pose estimation.
 - Test the system in varying temperature environments to assess the stability of the IMU, camera, and LiDAR.
 - Analyze how environmental noise (e.g., vibrations) affects sensor reading and fusion accuracy.

Conclusion

The "Inside-Out Tracking Sensor Suite" demonstrates significant potential as a high-performance, cost-effective solution for real-time head pose estimation in virtual reality (VR) applications. Through the integration of inertial measurement units (IMUs) and camera-based visual odometry, the proposed system effectively addresses the challenges of accurate, low-latency, and six-degree-of-freedom (6-DOF) tracking. Compared to alternatives such as external tracking systems and standalone inertial tracking, the proposed solution offers superior portability, robustness, and user convenience by eliminating external dependencies and reducing drift errors through advanced sensor fusion.

The analysis of alternative solutions highlights key trade-offs, with external tracking systems providing high accuracy but requiring fixed setups, and standalone IMUs being more portable but prone to drift and environmental limitations. The integration of IMU and camera subsystems, coupled with algorithms like Extended Kalman Filters and Madgwick Filters, strikes a balance between accuracy, flexibility, and cost-efficiency, making the proposed solution ideal for diverse VR environments.

The report provides a comprehensive overview of the project's objectives, system design, testing methodologies, and potential challenges. It emphasizes the technical innovations in the main solution and the steps taken to refine its performance. Concluding remarks underscore the system's readiness for further development and testing to address remaining challenges, ensuring that the product achieves its target specifications and aligns with market demands. This project lays the foundation for delivering an innovative, user-friendly, and accessible VR tracking solution, paving the way for future advancements in immersive technologies.

REFERENCES

1. 3Blue1Brown. (2018, February 10). *Quaternions and 3D rotation, explained visually* [Video]. YouTube. <https://youtu.be/zjMuIxRvygQ?si=ahSenbcOqZpvQe2V>
2. 3Blue1Brown. (2018, September 6). *Visualizing quaternions (4d numbers) with stereographic projection* [Video]. YouTube. <https://www.youtube.com/watch?v=d4EgbgTm0Bq&t=502s>
3. Chen, W., Shang, G., Hu, K., Zhou, C., Wang, X., Fang, G., & Ji, A. (2022). A Monocular-visual SLAM system with semantic and optical-flow fusion for indoor dynamic environments. *Micromachines*, 13(11), 2006.
4. Fischler, M. A., & Bolles, R. C. (1981). *Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography*. *Communications of the ACM*, 24, 381–395. <https://doi.org/10.1145/358669.358692>
5. Free3D. (n.d.). *Rigged male human* [3D model]. Retrieved from <https://free3d.com/3d-model/rigged-male-human-442626.html>
6. Freepik. (n.d.). *Pouch 001* [3D model]. Retrieved from https://www.freepik.com/3d-model/pouch-001_3146.htm
7. Hartley, R., & Zisserman, A. (2003). *Multiple view geometry in computer vision*. New York, NY, USA: Cambridge University Press. ISBN: 0521540518
8. Honeywell. (2010). *HMC5883L 3-axis digital compass IC* [Data sheet]. Adafruit.
9. InvenSense. (2013). *MPU-6000 and MPU-6050 product specification revision 3.4* [Data sheet].
10. Jianbo Shi, & Tomasi. (1994). *Good features to track*. 1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, pp. 593–600. <https://doi.org/10.1109/CVPR.1994.323794>
11. Li Y, Jack Wang J. A Pedestrian Navigation System Based on Low Cost IMU. *Journal of Navigation*. 2014;67(6):929–949. doi:10.1017/S0373463314000344
12. Lucas, B. D., & Kanade, T. (1981). *An iterative image registration technique with an application to stereo vision*. Proceedings of Imaging Understanding Workshop, pp. 121–130.
13. MDPI. (2017). *Sensors | Free Full-Text | Title of the article*. *Sensors*, 17(10), 2164. <https://www.mdpi.com/1424-8220/17/10/2164>
14. MDPI. (2024). *Sensors | Free Full-Text | Title of the article*. *Sensors*, 24(11), 3427. <https://www.mdpi.com/1424-8220/24/11/3427>
15. Nistér, D. (2004). *An efficient solution to the five-point relative pose problem*. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(6), 756–770. <https://doi.org/10.1109/TPAMI.2004.17>
16. Tome, D., Alldieck, T., Peluse, P., Pons-Moll, G., Agapito, L., Badino, H., & De la Torre, F. (2020). Selfpose: 3d egocentric pose estimation from a headset mounted camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(6), 6794–6806.
17. Wijayasinghe, I. B., Saadatzi, M. N., Abubakar, S., & Popa, D. O. (2018). *A study on optimal placement of accelerometers for pose estimation of a robot arm*. 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia, pp. 1444–1451. <https://doi.org/10.1109/ICRA.2018.8460501>

APPENDIX

Appendix 1: Sensor Suit Design

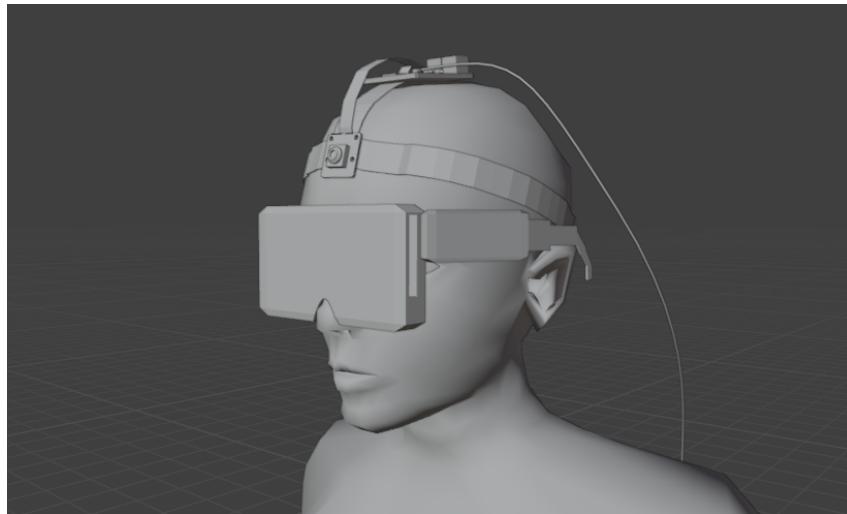


Figure 22. Sensor Suit Design



Figure 23. Sensor Suit Design from different perspective



Figure 24. Full Sensor Suit Design

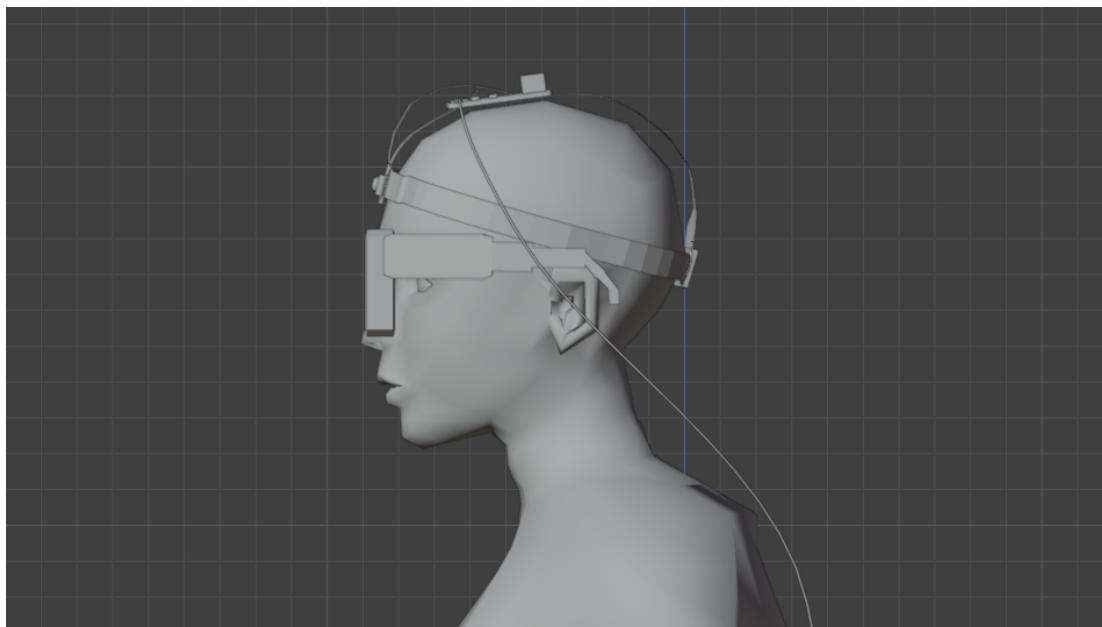


Figure 25. Sensor Suit Design Headset

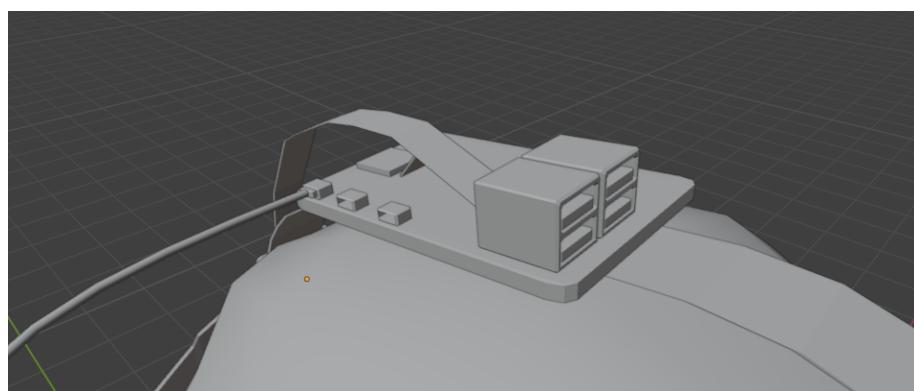


Figure 26. Sensor Suit Design Raspberry

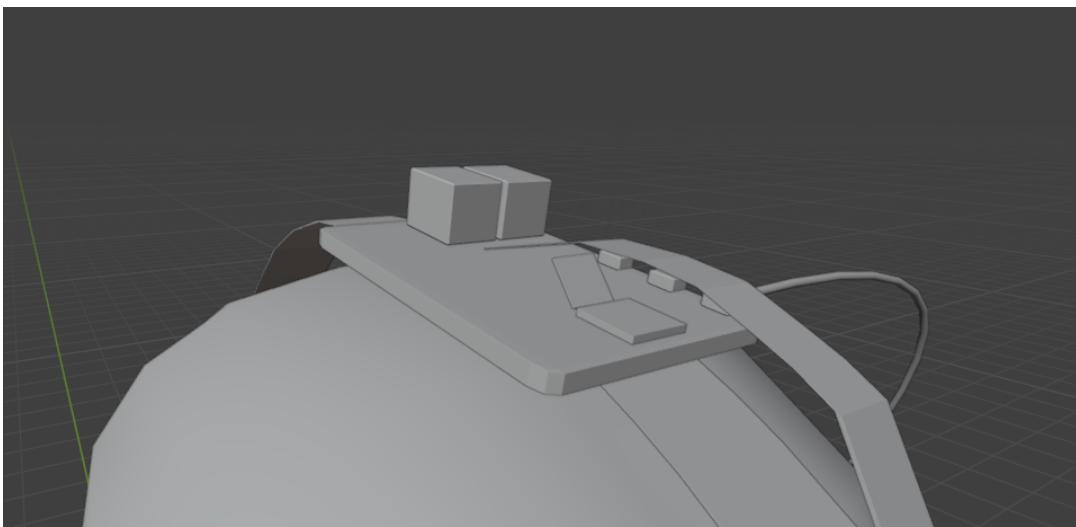


Figure 27. Sensor Suit Design IMU

| Parameter | Actual Performance IMU | | | | Actual Performance Camera | | | | Expected Performance | Expected Max Error | Error IMU | Error Camera | Avg Error | Passed | Test Done By | |
|--|------------------------|-------|-------|-------|---------------------------|--------------------------|--------------------------|--------------------------|------------------------|--------------------|-----------|--------------|-----------|--------|--------------|--------|
| | 1. | 2. | 3. | Avg | 1. | 2. | 3. | Avg | | | | | | | | |
| Translational Accuracy (x) Initial | 0,00 | 0,01 | 0,01 | 0,01 | - | - | - | - | $\leq \pm 6\text{cm}$ | $\pm 10\%$ | 0,01 | 0,01 | NA | NA | SA ÖÖ | - |
| Translational Accuracy (x) 20cm Forward | 22,04 | 24,18 | 21,89 | 22,70 | [0,63 -0,70 -0,35] | [0,64 -0,67 -0,36] | [0,63 -0,68 -0,35] | [0,63 -0,68 -0,35] | $\leq \pm 6\text{cm}$ | $\pm 10\%$ | 2,70 | 13,50 | 50,94° | NA | SA ÖÖ | EÇ HAY |
| Translational Accuracy (x) 20cm Backward | -4,16 | -4,16 | -4,52 | -4,28 | [-0,67 0,60 0,39] | [-0,70 0,60 0,50] | [-0,70 0,50 0,43] | [-0,68 0,57 0,44] | $\leq \pm 6\text{cm}$ | $\pm 10\%$ | -4,78 | -4,78 | 47,15° | NA | SA ÖÖ | EÇ HAY |
| Translational Accuracy (x) 40cm Forward | 45,11 | 42,53 | 43,19 | 43,61 | [0,93 0,08 -0,33] | [0,94 -0,03 -0,32] | [0,75 -0,5 -0,4] | [0,87 -0,15 -0,35] | $\leq \pm 6\text{cm}$ | $\pm 10\%$ | 3,61 | 9,02 | 29,54° | NA | SA ÖÖ | EÇ HAY |
| Translational Accuracy (y) Initial | 0,04 | 0,02 | 0,01 | 0,02 | - | - | - | - | $\leq \pm 6\text{cm}$ | $\pm 10\%$ | 0,02 | 0,02 | NA | NA | SA ÖÖ | - |
| Translational Accuracy (y) 20cm Forward | 22,10 | 21,47 | 23,49 | 22,35 | [-0,60 -0,73 0,27] | [0,15 -0,98 -0,07] | [-0,36 -0,76 0,53] | [-0,28 -0,82 0,24] | $\leq \pm 6\text{cm}$ | $\pm 10\%$ | 2,35 | 11,77 | 34,91° | NA | SA ÖÖ | EÇ HAY |
| Translational Accuracy (y) 20cm Backward | -2,18 | -3,11 | -1,83 | -2,37 | [-0,02 0,90 -0,03] | [-0,10 0,90 0,06] | [-0,63 0,68 0,35] | [-0,25 0,83 0,13] | $\leq \pm 6\text{cm}$ | $\pm 10\%$ | -2,37 | -2,37 | 33,9° | NA | SA ÖÖ | EÇ HAY |
| Translational Accuracy (y) 40cm Forward | 44,84 | 43,56 | 38,82 | 42,41 | [0,45 0,84 -0,23] | [0,30 0,93 -0,17] | [0,08 0,99 -0,13] | [0,28 0,92 -0,18] | $\leq \pm 6\text{cm}$ | $\pm 10\%$ | 2,41 | 6,02 | 23,07° | NA | SA ÖÖ | EÇ HAY |
| Translational Accuracy (z) Initial | 0,01 | 0,05 | 0,03 | 0,03 | - | - | - | - | $\leq \pm 6\text{cm}$ | $\pm 10\%$ | 0,03 | 0,03 | NA | NA | SA ÖÖ | - |
| Translational Accuracy (z) 20cm Forward | 23,92 | 18,37 | 24,75 | 22,35 | [-0,20 0,26 0,90] | [0,64 0,35 0,67] | [0,44 -0,05 0,89] | [0,29 0,19 0,82] | $\leq \pm 10\text{cm}$ | $\pm 10\%$ | 2,35 | 2,35 | 34,91° | NA | SA ÖÖ | EÇ HAY |

| | | | | | | | | | | | | | | | | |
|--|---------|---------|---------|---------|--------------------------|--------------------------|--------------------------|--------------------------|-----------------------------|------------|-------|-------|--------|-------|----------|-----------|
| Translational Accuracy (z) 20 cm Backward | 7,23 | 4,45 | 8,92 | 6,87 | [0,11 -0,38 -0,90] | [-0,30 0,43 -0,82] | [0,58 -0,54 -0,62] | [0,13 -0,16 -0,78] | \leq $\pm 10\text{cm}$ | $\pm 10\%$ | 6,87 | 6,87 | 38,73° | NA | SA ÖÖ | EÇ HAY |
| Translational Accuracy (z) 40cm Forward | 51,27 | 47,77 | 37,81 | 45,62 | [-0,42 -0,23 0,87] | [0,25 0,39 0,88] | [0,73 0,08 0,67] | [0,19 0,08 0,81] | \leq $\pm 10\text{cm}$ | $\pm 10\%$ | 5,62 | 14,04 | 35,9° | NA | SA ÖÖ | EÇ HAY |
| Rotational Accuracy (Roll) Initial to -60° (Camera: -30°) | -59,43 | -58,22 | -57,99 | -58,55 | -32,00 | -34,00 | -35,50 | -33,80 | $\leq \pm 4^\circ$ | $\pm 10\%$ | 1,45 | 2,42 | -3,80 | -0,69 | SA ÖÖ | EÇ HAY |
| Rotational Accuracy (Roll) -60 to 0° | 1,20 | 0,90 | 0,52 | 0,87 | 0,01 | 0,50 | 0,07 | 0,20 | $\leq \pm 4^\circ$ | $\pm 10\%$ | 0,87 | 0,87 | 0,20 | 0,54 | SA ÖÖ | EÇ HAY |
| Rotational Accuracy (Roll) 0 to 45° | 42,33 | 45,05 | 46,22 | 44,53 | 42,00 | 46,00 | 43,40 | 43,80 | $\leq \pm 4^\circ$ | $\pm 10\%$ | -0,47 | -1,04 | -1,20 | -1,12 | SA ÖÖ | EÇ HAY |
| Rotational Accuracy (Roll) 45 to 60° | 60,04 | 59,19 | 61,06 | 60,10 | 30,00 | 29,00 | 29,10 | 29,36 | $\leq \pm 4^\circ$ | $\pm 10\%$ | 0,10 | 0,16 | -0,24 | -0,04 | SA ÖÖ | EÇ HAY |
| Rotational Accuracy (Yaw) Initial to -150° (Camera: 45°) | -147,55 | -151,39 | -150,28 | -149,74 | 48,50 | 58,00 | 49,40 | 51,60 | $\leq \pm 4^\circ$ | $\pm 10\%$ | 0,26 | 0,17 | 5,60 | 2,89 | SA ÖÖ | EÇ HAY |
| Rotational Accuracy (Yaw) -150 to -60° | -58,73 | -61,29 | -62,44 | -60,82 | -65,50 | -64,70 | -61,60 | -63,70 | $\leq \pm 4^\circ$ | $\pm 10\%$ | -0,82 | -1,37 | -3,70 | -2,53 | SA ÖÖ | EÇ HAY |
| Rotational Accuracy (Yaw) -60 to 0° | 1,10 | -2,40 | 0,75 | -0,18 | -0,10 | 0,02 | 0,02 | 0,01 | $\leq \pm 4^\circ$ | $\pm 10\%$ | -0,18 | -0,18 | 0,01 | -0,09 | SA ÖÖ | EÇ HAY |
| Rotational Accuracy (Yaw) 0 to 90° | 88,25 | 89,26 | 87,81 | 88,44 | -70,00 | -92,00 | -120,00 | -94,00 | $\leq \pm 4^\circ$ | $\pm 10\%$ | -1,56 | -1,73 | -4,00 | -2,87 | SA ÖÖ | EÇ HAY |
| Rotational Accuracy (Pitch) Initial to -60° (Camera: -30°) | -60,23 | -59,62 | -59,51 | -59,79 | -46,00 | -33,00 | -39,00 | -39,30 | $\leq \pm 4^\circ$ | $\pm 10\%$ | 0,21 | 0,36 | -9,30 | -4,47 | MK | EÇ HAY |

| | | | | | | | | | | | | | | | | |
|--|---------------|---------------|---------------|---------------|-------|-------|-------|-------|---|------------|-------|-------|-------|-------|----|--------|
| Rotational Accuracy (Pitch) -60 to Initial | 0,11 | 0,41 | 0,17 | 0,23 | 0,01 | -0,30 | 0,20 | -0,03 | $\leq \pm 4^\circ$ | $\pm 10\%$ | 0,23 | 0,23 | -0,03 | 0,10 | MK | EÇ HAY |
| Rotational Accuracy (Pitch) Initial to 45° | 46,16 | 46,71 | 46,53 | 46,47 | 48,00 | 47,00 | 55,00 | 50,00 | $\leq \pm 4^\circ$ | $\pm 10\%$ | 1,47 | 3,26 | 5,00 | 4,13 | MK | EÇ HAY |
| Rotational Accuracy (Pitch) (60°) (Camera: 30°) | 60,92 | 59,61 | 59,54 | 60,02 | 30,00 | 31,00 | 19,70 | 26,59 | $\leq \pm 4^\circ$ | $\pm 10\%$ | 0,02 | 0,04 | -3,41 | -1,69 | MK | EÇ HAY |
| Combined T(x) + R(Yaw) (0cm/0°) | 0,02 / -1,05 | 0,05 / 0,42 | 0,01 / 0,38 | 0,03 / -0,08 | NA | NA | NA | NA | $\leq \pm 6\text{cm}$, $\leq \pm 3^\circ$ | $\pm 10\%$ | -0,08 | -0,08 | NA | NA | MK | - |
| Combined T(x) + R(Yaw) (20cm/90°) | 22,14 / 90,53 | 24,72 / 86,39 | 24,15 / 87,38 | 23,67 / 88,10 | NA | NA | NA | NA | $\leq \pm 6\text{cm}$, $\leq \pm 3^\circ$ | $\pm 10\%$ | -1,90 | -2,11 | NA | NA | MK | - |
| Combined T(x) + R(Yaw) (10cm/60°) | 10,52 / 60,53 | 11,64 / 59,31 | 11,10 / 59,33 | 11,09 / 59,72 | NA | NA | NA | NA | $\leq \pm 6\text{cm}$, $\leq \pm 3^\circ$ | $\pm 10\%$ | -0,28 | -0,47 | NA | NA | MK | - |
| Combined T(y) + R(Yaw) (0cm/0°) | 0,01 / 0,50 | 0,02 / 0,80 | 0,02 / 0,30 | 0,02 / 0,50 | NA | NA | NA | NA | $\leq \pm 6\text{cm}$, $\leq \pm 3^\circ$ | $\pm 10\%$ | 0,50 | 0,50 | NA | NA | MK | - |
| Combined T(y) + R(Yaw) (20cm/90°) | 21,45 / 88,37 | 19,88 / 90,05 | 24,51 / 89,55 | 21,94 / 89,32 | NA | NA | NA | NA | $\leq \pm 6\text{cm}$, $\leq \pm 3^\circ$ | $\pm 10\%$ | -0,68 | -0,76 | NA | NA | MK | - |
| Combined T(y) + R(Yaw) (10cm/60°) | 13,11 / 59,49 | 10,45 / 59,30 | 14,50 / 60,15 | 12,69 / 59,65 | NA | NA | NA | NA | $\leq \pm 6\text{cm}$, $\leq \pm 3^\circ$ | $\pm 10\%$ | -0,35 | -0,58 | NA | NA | MK | - |