POLITECNICO DI TORINO

Department of Electronics and Telecommunications

Master’s Degree Course in Mechatronic Engineering

Master Thesis

Mobile collaborative robot: impact of positional markers on accuracy

Supervisor:
Prof. Dario Antonelli

Candidate:
Mukhammadrizo Karimov

Academic Year 2023-2024

Table of Contents

Abstract

Industry 4.0 represents the fourth industrial revolution, characterized by a model of production and business management that integrates physical and digital devices. This model, particularly prevalent in manufacturing, is propelled by emerging technologies such as human-robot collaboration, the Industrial Internet of Things (IIoT), cyber-physical systems, augmented reality, machine learning, and artificial intelligence. These technologies are essential for analyzing, interpreting, and predicting the behavior of machinery and devices in manufacturing, including autonomous robots and automated guided vehicles (AGVs). A significant advantage of Industry 4.0's digital transformation is the real-time access to comprehensive information from devices involved in manufacturing processes. This data, gathered by embedded sensors and camera systems monitoring the shop floor, is transmitted through the internet for storage and further processing. Consequently, these data can be easily accessed and used to create a virtual environment mirroring the real one.

Mobile collaborative robots (cobots) have emerged as versatile tools in various industrial and service sectors, offering flexibility and efficiency in dynamic environments. However, ensuring repeatability, a crucial aspect in their operations, remains a significant challenge. This thesis investigates methodologies to enhance repeatability in mobile collaborative robots, focusing on both hardware and software aspects.

The study begins with an in-depth analysis of existing literature, identifying key factors affecting repeatability in mobile cobots. Through empirical experiments and simulations, various factors such as localization accuracy, motion control mechanisms, environmental disturbances, and collaborative interaction dynamics are systematically explored.

Furthermore, the thesis proposes novel solutions to address identified challenges. Hardware improvements include advancements in sensor technology, enhanced actuation mechanisms, and robust mechanical design to minimize structural deformations. Software enhancements involve sophisticated algorithms for localization and mapping, adaptive motion planning strategies, and real-time feedback control systems to mitigate external disturbances and ensure precise trajectory tracking.

Overall, this thesis contributes to the advancement of mobile collaborative robotics by providing a comprehensive understanding of repeatability challenges and offering practical solutions to enhance the performance and reliability of these systems in dynamic environments.

1. Introduction

1.1 Objective of the thesis

This thesis seeks to investigate the repeatability and accuracy of the MIR100 mobile industrial robot by utilizing data collected from two distinct sources: ROS (Robot Operating System) and OptiTrack motion capture system. The analysis will encompass scenarios both with and without the use of V marker tracking, aiming to provide a comprehensive understanding of how different tracking methods influence the robot's performance metrics. By examining a wide range of operational conditions and environments, this research aims to offer valuable insights into optimizing the MIR100's performance for industrial applications, thereby contributing to the advancement of mobile robotics in manufacturing and automation industries. The software and tools utilized include:

- Optitrack Motive
- Excel
- Matlab and Simulink
- ROS
- Ubuntu Virtual Machine including Rviz

The subsequent chapters will provide a detailed exploration of these methodologies, elucidating their features and consequences.

During the thesis Mobile Manipulator (MoMa) prototype was used, which was developed in the laboratory where this thesis was conducted, specifically in the Mid4Lab at Politecnico di Torino. It was created by integrating a mobile robot, MiR100, with a robotic manipulator from the UR3 CB3 series. Since it was already assembled, only the mobile component of this Mobile Manipulator, the MiR100, was utilized.

Figure 1: MiR100 which experiment carried out.

1.2 Mobile industrial robot (MiR100)

The MiR100 stands at the forefront of innovation in the realm of mobile robotics, epitomizing a fusion of cutting-edge technology and practical industrial utility. Crafted to cater to the evolving needs of modern industrial and logistics operations, this mobile robot represents a paradigm shift in automated material handling and transportation.

With its sleek and agile design, the MiR100 embodies versatility, seamlessly maneuvering through intricate and dynamic environments with unparalleled precision. Equipped with state-of-the-art sensors and navigation systems, including laser scanners and 3D cameras, it navigates autonomously, effortlessly adapting to changes in its surroundings while meticulously avoiding obstacles in its path.

Beyond its remarkable mobility, the MiR100 boasts an impressive payload capacity of up to 100 kg can reach a maximum linear speed of 1.5m/s, making it an invaluable asset in the transportation of a diverse array of materials, components, and finished products. Whether it's moving heavy machinery across a manufacturing floor or delivering goods within a warehouse, this mobile robot excels in optimizing logistical workflows with unparalleled efficiency and reliability.

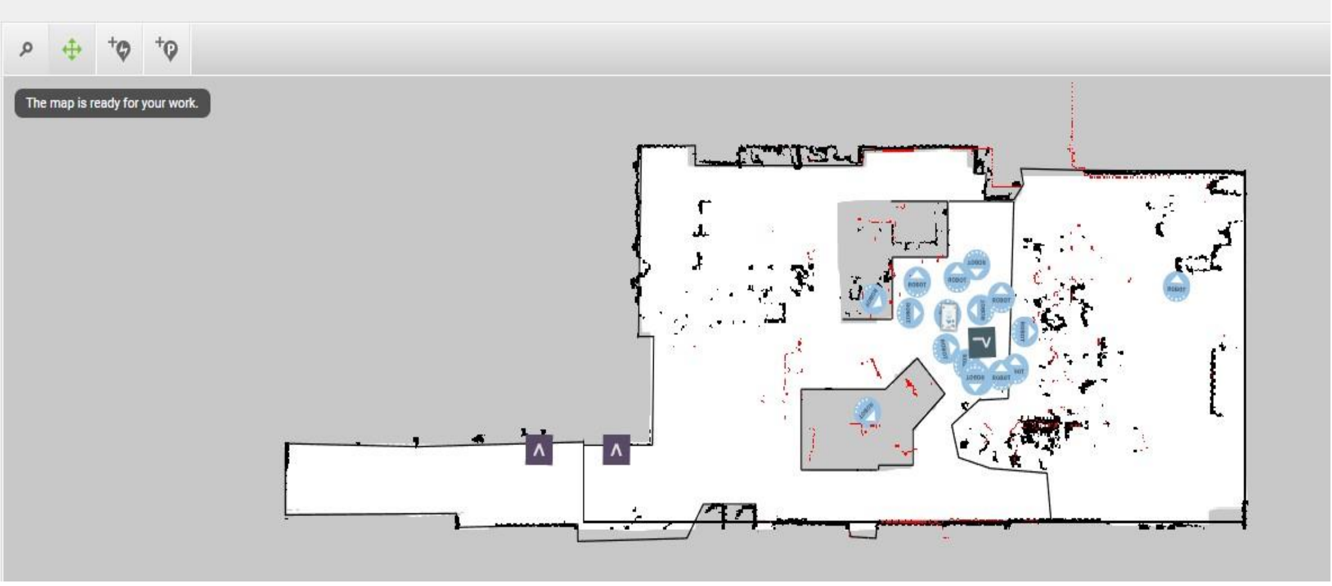
Furthermore, the MiR100 is not merely a standalone solution but a seamlessly integrable component within existing industrial ecosystems. Its intuitive and user-friendly interface allows for effortless programming and operation, empowering users of all skill levels to leverage its capabilities with ease. Moreover, its modular design facilitates the integration of custom payload interfaces, enabling tailored solutions to meet specific application requirements.

Durability and reliability are hallmarks of the MiR100, as it is engineered to withstand the rigors of industrial environments while maintaining peak performance over extended periods of operation. Robust construction, reinforced wheels, and ruggedized components ensure resilience in the face of challenging conditions, guaranteeing uninterrupted productivity and minimal downtime.

In essence, the MiR100 represents a transformative leap forward in the realm of mobile robotics, revolutionizing the way materials are handled, transported, and managed within industrial and logistics settings. With its unparalleled mobility, versatility, and reliability, it stands as a beacon of innovation, empowering businesses to achieve new heights of efficiency, productivity, and competitiveness in today's fast-paced and dynamic world.

Beyond its remarkable mobility, the MiR100 boasts an impressive payload capacity of up to 100 kg can reach a maximum linear speed of 1.5m/s, making it an invaluable asset in the transportation of a diverse array of materials, components, and finished products. Whether it's moving heavy machinery across a manufacturing floor or delivering goods within a warehouse, this mobile robot excels in optimizing logistical workflows with unparalleled efficiency and reliability.

Figure 2: Mobile Industrial Robot MiR100.



The map is ready for your work.

Figure 3: MiR100 live map from MiR's software

1.3 V-marker of the MiR100

In the realm of Mobile Industrial Robots (MiR), various types of markers are employed to facilitate navigation, localization, and efficient operation within industrial environments. Among these markers are the V-marker, VL-marker, L-marker, and Bar marker, each serving specific purposes in guiding the robot's movement and tasks.

- V-marker: The V marker, also known as the Virtual Marker, is a reference point used in the navigation systems of robots, including Mobile Industrial Robots (MiR). It represents the point in a system's impulse response where the response begins to deviate significantly from zero after being excited by an impulse input. In robotics, the V marker helps in characterizing the behavior of linear time-invariant (LTI) systems, aiding in system modeling, filter design, and system identification.

- VL-marker: The VL marker, or Virtual Line Marker, is a virtual reference line used by robots for navigation and path planning. It is typically defined in the robot's software environment and represents a boundary or guideline for the robot to follow. VL markers are commonly used to create virtual paths or corridors for robots to navigate within industrial environments.
- L-marker: The L marker, or Localization Marker, is a physical or virtual marker used by robots to determine their position within an environment accurately. L markers can take various forms, such as QR codes, barcodes, unique visual patterns, or wireless beacons. By detecting and recognizing these markers, robots can localize themselves relative to their surroundings, enabling precise navigation and task execution.
- Bar-marker: The Bar marker is a type of physical marker used for navigation and localization purposes in robotics. It typically consists of a linear barcode or QR code affixed to a surface within the robot's operating environment. Bar markers are commonly employed in industrial settings to provide reference points for robots, allowing them to determine their position and orientation accurately. By detecting and decoding bar markers, robots can navigate autonomously, avoid obstacles, and perform tasks efficiently within their workspace.

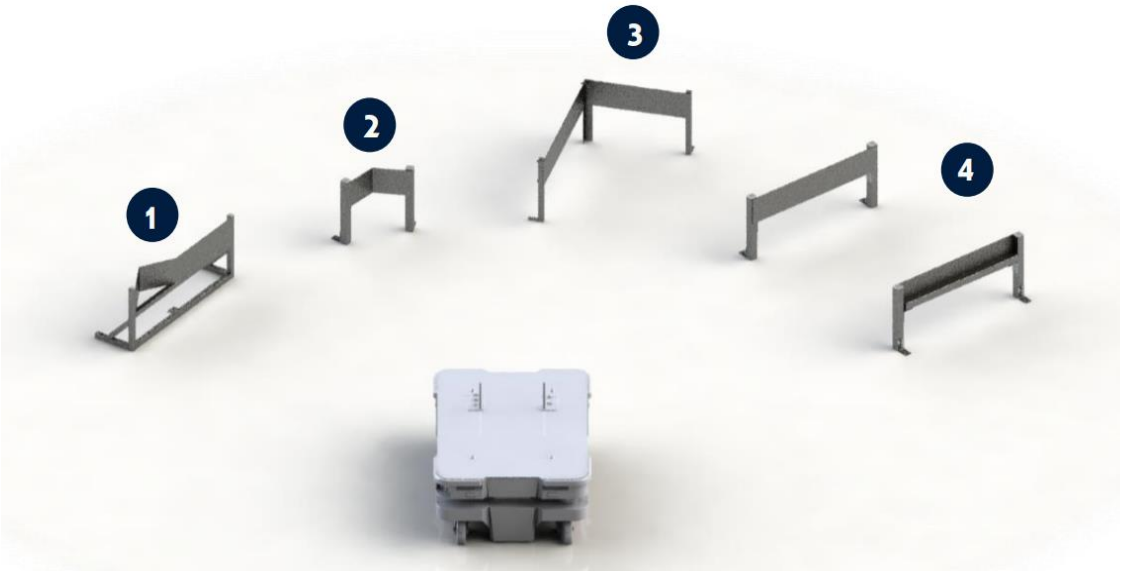


Figure 4: Position markers of MiR100

2. Data acquisition

2.1 Robot operating system (ROS)

ROS, short for Robot Operating System, is an open-source framework designed to facilitate the development of robotic software. Despite its name, ROS is not an operating system in the traditional sense; rather, it serves as a middleware that runs on top of existing operating systems like Linux.

Here's a more detailed explanation of ROS:

Middleware Architecture: ROS adopts a distributed computing architecture, allowing robotic components to communicate with each other seamlessly. It provides a set of communication protocols and libraries for passing messages between different modules of a robotic system. This middleware approach promotes modularity and reusability, as individual components can be developed independently and integrated into larger systems.

Tools and Libraries: ROS offers a comprehensive set of tools and libraries to support various aspects of robotics development. These include:

- ROS Core: The core of ROS, responsible for managing communication between nodes, launching processes, and managing the ROS file system.
- ROS Packages: Collections of related functionalities, such as drivers for specific sensors or actuators, perception algorithms, motion planning libraries, and simulation tools.
- ROS Messages: Defined data structures for passing information between nodes, enabling standardized communication between different components of a robotic system.
- ROS Services: Remote procedure calls that allow nodes to invoke functions on other nodes, enabling distributed computation and control.
- ROS Actions: Higher-level interfaces for managing long-running tasks, such as trajectory execution or navigation planning.
- Visualization Tools: ROS provides graphical tools for visualizing robot states, sensor data, and environment models, facilitating debugging and analysis.

Community and Ecosystem: One of the strengths of ROS is its large and active community of developers and researchers. The ROS community contributes to the development, maintenance, and documentation of the framework, as well as the creation of open-source packages and libraries. This collaborative ecosystem enables rapid innovation and knowledge sharing within the robotics community.

Cross-Platform Compatibility: ROS is designed to be platform-independent and can run on various operating systems, although it is primarily supported on Linux. This cross-platform compatibility allows developers to use ROS on a wide range of hardware platforms, from embedded systems to high-performance computing clusters.

Simulation and Testing: ROS provides tools for simulating robotic systems in virtual environments, allowing developers to test algorithms and behaviors without the need for physical hardware. Simulation tools like Gazebo and RViz are commonly used in conjunction with ROS to visualize robot models, sensor data, and simulation results.

Integration with Robotics Frameworks: ROS can integrate with other robotics frameworks and platforms, such as OpenCV for computer vision, PCL (Point Cloud Library) for 3D perception, and TensorFlow for machine learning. This interoperability enables developers to leverage existing libraries and algorithms within the ROS ecosystem.

Furthermore, users have access to an array of tools and software options to execute numerous tasks and operations related to interconnected devices. Within the Linux operating system, individuals can utilize 3D simulation environments such as Gazebo and RVIZ, which can be linked to ROS via dedicated bridge nodes.

Gazebo is a powerful open-source 3D simulation environment primarily used for robotics development, but it's also applicable in other fields like autonomous vehicles and drones. It allows users to simulate complex scenarios and environments, including physics-based interactions, sensor simulations, and robot behavior. Gazebo provides a realistic simulation platform for testing and validating algorithms, controllers, and systems before deployment in the real world. It offers a wide range of features, including support for various sensors, customizable environments, and integration with other robotics frameworks like ROS. Overall, Gazebo is a valuable tool for researchers, engineers, and hobbyists working in the robotics domain.

RVIZ, short for ROS Visualization, is a powerful 3D visualization tool primarily used in the context of the Robot Operating System (ROS). It provides users with the capability to visualize sensor data, robot models, and other information in a three-dimensional space. RViz is commonly utilized for debugging, testing, and monitoring robotic systems during development and operation.

Both Gazebo and RViz offer the functionality to execute robot models described in the Unified Robotics Description Format (URDF). This format allows users to define the properties and configurations of robots in a standardized XML-based language. Gazebo and RViz support URDF files, enabling users to simulate and visualize robot behavior accurately within their respective environments.

Overall, ROS serves as a powerful and versatile framework for developing, prototyping, and deploying robotic systems across a wide range of applications, including industrial automation, autonomous vehicles, drones, healthcare robotics, and more. Additionally, ROS is considered one of the best integration softwares for Digital twins.

Real-time data collected from robots can be stored using two primary methods. Initially, it can be directly archived into a bag file by executing the "rosbag" command on the ROS machine while specifying the relevant topics of interest. This bag file format efficiently captures the data stream, preserving the timing and content of each message for later analysis or playback.

Alternatively, MATLAB and Simulink offer convenient options for storing real-time data. In MATLAB, the "rosbag" command can be utilized to save data directly from within MATLAB scripts or functions. Similarly, in Simulink, settings can be configured to capture and store real-time data during simulations. These integrated approaches provide seamless compatibility with the ROS ecosystem, allowing for efficient data management and analysis within the MATLAB and Simulink environments.

2.1.1 Connecting ROS with Matlab and Simulink.

Connecting ROS (Robot Operating System) with MATLAB and Simulink enables seamless integration of robotics data and control into MATLAB's numerical computing environment and Simulink's simulation environment. Here's how you can connect ROS with MATLAB and Simulink:

- ROS Toolbox for MATLAB: MATLAB provides the ROS Toolbox, which allows you to communicate with ROS in MATLAB. You can use this toolbox to subscribe to ROS topics, publish messages, call ROS services, and interact with ROS nodes directly from MATLAB scripts or functions. ROS Toolbox also provides support for working with ROS bags, visualizing ROS data, and integrating ROS functionality into MATLAB algorithms and workflows.

- Simulink Support for ROS: Simulink offers built-in support for ROS through the Robotics System Toolbox. With this toolbox, you can design and simulate robotic systems that interact with ROS nodes and topics. Simulink blocks are available for subscribing to ROS topics, publishing messages, and interfacing with ROS services. You can use Simulink to model complex robot behaviors, control algorithms, and sensor fusion processes, and then deploy these models onto real robotic hardware running ROS.

- ROS Bridge: ROS provides a ROS Bridge package that allows communication between ROS and non-ROS systems, including MATLAB and Simulink. The ROS Bridge converts ROS messages into a format that MATLAB and Simulink can understand, enabling seamless data exchange between ROS and MATLAB/Simulink environments. You can use the ROS Bridge to stream data between ROS nodes and MATLAB/Simulink models, enabling real-time monitoring, analysis, and control of robotic systems.

- Custom ROS Nodes: You can develop custom ROS nodes in MATLAB or Simulink using the Robotics System Toolbox or Simulink Coder. These custom nodes can interface with ROS topics, services, and parameters, allowing you to implement complex algorithms or control strategies directly within the ROS ecosystem.

By leveraging these tools and techniques, you can establish bidirectional communication between ROS and MATLAB/Simulink, enabling comprehensive analysis, simulation, and control of robotic systems within a unified environment. This integration facilitates rapid prototyping, testing, and deployment of robotics applications, accelerating the development cycle and enhancing the capabilities of robotic systems.

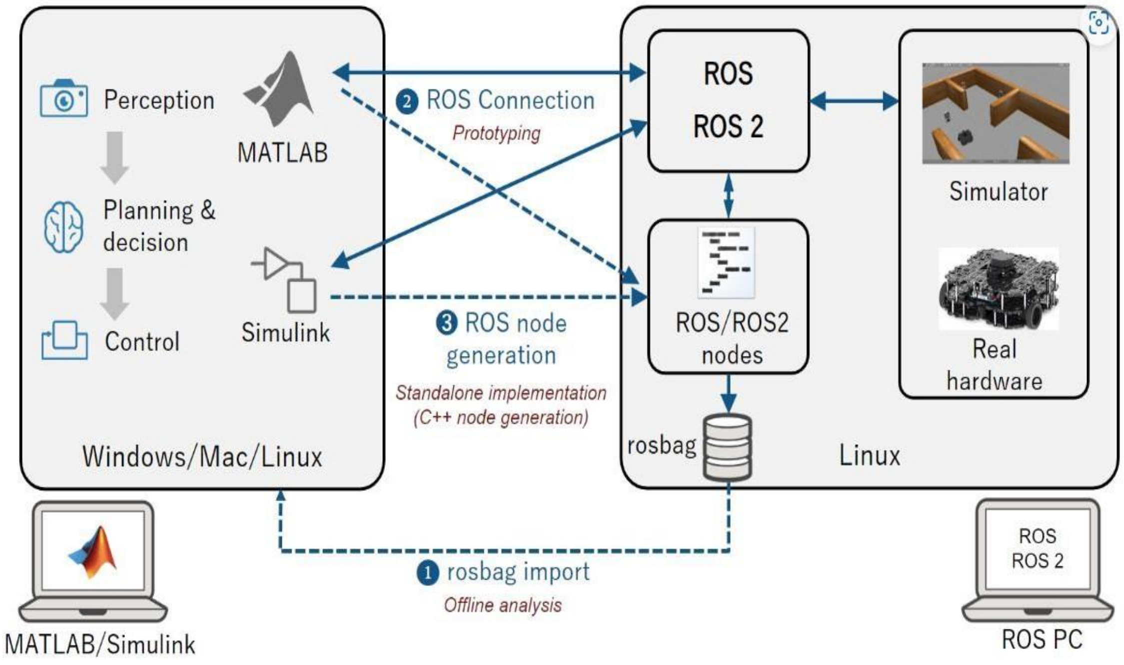


Figure 5: Diagram of communication between ROS devices and Matlab.

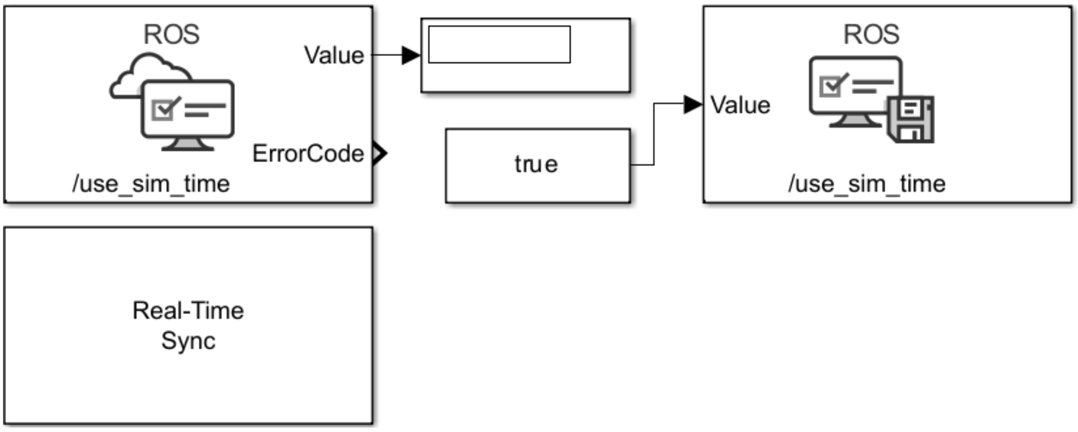


Figure 6: Time synchronization of Matlab and ROS in Simulink

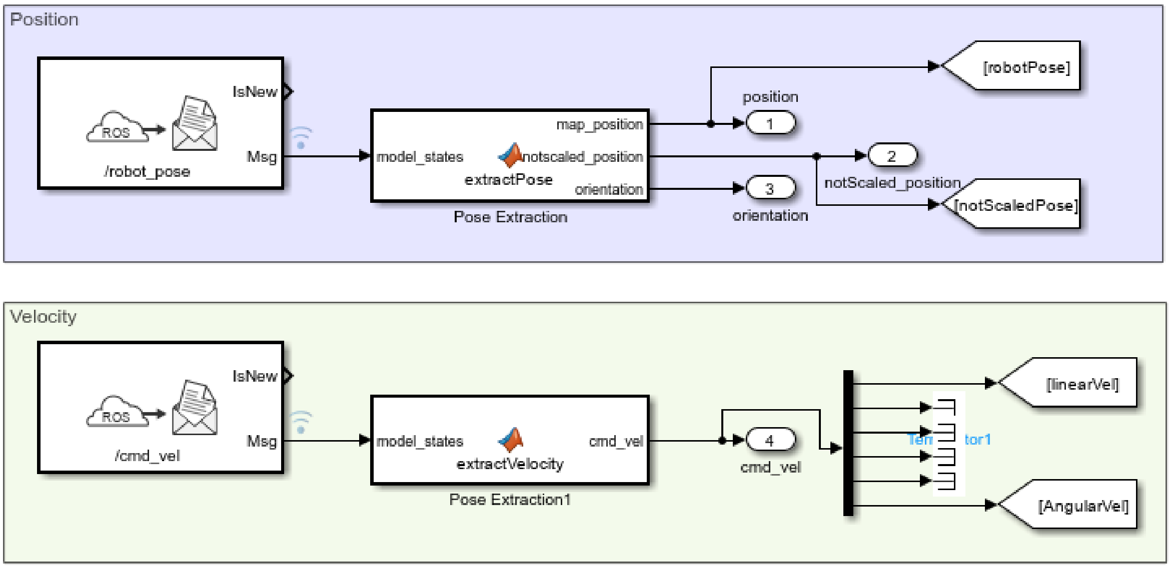


Figure 7: Simulink models to gain position and velocity of MiR100.

2.2 Optitrack

Optitrack Motive is a motion capture software developed by NaturalPoint, a leading provider of optical motion capture technology. It is designed to capture, track, and analyze the movements of objects or subjects in real-time with high precision and accuracy. Here's a breakdown of its key features and functionalities:

- Camera Calibration: Before capturing motion data, Optitrack Motive allows users to calibrate their camera setup. This calibration process ensures accurate tracking by establishing the relationship between the physical space and the camera sensors.
- Marker Tracking: Optitrack Motive primarily uses reflective markers placed on objects or subjects to track their movements. These markers reflect infrared light emitted by specialized cameras, allowing the software to precisely calculate their positions and orientations in 3D space.
- Real-time Data Capture: The software provides real-time feedback, allowing users to monitor and analyze motion data as it's being captured. This feature is particularly useful for live performances, biomechanics research, or virtual reality applications where immediate feedback is essential.
- Post-processing Tools: After capturing motion data, Optitrack Motive offers various post-processing tools for refining and enhancing the captured data. Users can filter noise, fill gaps, and smooth trajectories to ensure the accuracy and reliability of the recorded motion.
- Integration with Other Software: Optitrack Motive is designed to integrate seamlessly with other software applications commonly used in animation, biomechanics, robotics, virtual reality, and other fields. This allows users to import/export data, synchronize multiple data streams, and perform complex analyses using specialized software tools.
- Compatibility with Optitrack Hardware: While Optitrack Motive is a standalone software package, it is optimized for use with OptiTrack's hardware solutions, including high-speed cameras, motion capture suits, and accessories. This tight integration ensures maximum performance and reliability for motion capture applications.
- User-friendly Interface: Optitrack Motive features an intuitive user interface, making it accessible to both novice and experienced users. The software provides easy-to-use tools for setting up capture environments, calibrating cameras, placing markers, and analyzing motion data, reducing the learning curve associated with motion capture technology.

Overall, Optitrack Motive is a powerful and versatile motion capture software solution suitable for a wide range of applications, including animation production, biomechanics research, sports performance analysis, and virtual reality content creation. Its combination of real-time tracking, post-processing capabilities, and seamless integration with Optitrack hardware makes it a preferred choice for professionals in various industries.

- Positional accuracies are typically within ±0.2 millimeters or less.
- Rotational accuracies are typically within ±0.1 degrees or less.
- Latency is less than 10 milliseconds.

The provided values represent the ideal specifications according to the official website. However, the actual accuracy of the system depends on its calibration, which is influenced by factors like camera conditions and environmental lighting. To calibrate the system, a dedicated wand with three calibration markers is slowly moved until all cameras accurately detect their positions. For the tasks in this thesis, a calibration with a mean error (ME) of 0.531mm is used, which currently represents the minimum achievable accuracy within the laboratory setting where the simulations and analyses take place. The accuracy level of the measurements in this thesis is determined by considering various observations and calculations:

- Uniform distribution
- Type B Evaluation of Standard Uncertainty (u)
- Expanded uncertainty (U) estimated by assuming a coverage factor k=2 at a confidence levelof 95%, in according to ISO GUM

$$\gamma = \mu \pm U \text{ – Measurement model of Optitrack}$$

$$U = k \cdot u$$

μ is the average with equal weighting for each value.

γ is the unknown value of the measurand, considering the disturbances.

$$\gamma = \mu \pm U = \mu \pm ku = \mu \pm k\frac{ME}{\sqrt{3}} = \mu \pm 2\,\frac{0.531}{\sqrt{3}} = (\mu \pm 0.613)\ mm$$

The expanded uncertainty, indicated by $U = 0.613$ millimeters, has been determined. Given the large volume of data gathered during the acquisition process, a 95% confidence interval is opted for to ensure that a considerable proportion of measurements align with the defined uncertainty thresholds. This selection is made to enhance the reliability of the measurements within the specified uncertainty bounds.

Based on this result, Optitrack is regarded as the most precise data acquisition system, faithfully capturing real-world phenomena. Hence, it serves as the reference model in this thesis for comparing data obtained from other acquisition systems.

2.2.1 Connection between Optitrack and Matlab

Motion capture systems like OptiTrack are valuable tools for capturing and analyzing movement data in various fields, including biomechanics, animation, and virtual reality. The integration of Optitrack with Matlab is a widely used numerical computing environment, for real-time data analysis. The integration leverages the NatNet SDK provided by NaturalPoint to establish a connection between Optitrack cameras and Matlab. Through this integration, motion capture data can be streamed into Matlab for processing and analysis, offering researchers and developers a powerful platform for understanding complex motion behaviors. The abstract outlines the steps involved in setting up the integration, including downloading and installing the NatNet SDK, configuring Optitrack cameras, and implementing data processing algorithms in Matlab. Additionally, it discusses potential applications of this integration, such as biomechanical analysis, virtual reality interaction, and animation production. Overall, the integration of Optitrack with Matlab provides researchers and developers with a versatile and efficient solution for real-time motion capture data analysis.

In order to connect Optitrack with Matlab, you would typically use the NatNet SDK (Software Development Kit) provided by NaturalPoint, the company behind Optitrack. Here's a general guide on how to establish this connection:

- Download NatNet SDK: Visit the NaturalPoint website and download the NatNet SDK. This SDK contains libraries and examples to help you integrate Optitrack data with your applications.
- Install SDK and Dependencies: Follow the instructions provided with the SDK to install it on your system. Ensure that you have any necessary dependencies installed as well.
- Set Up Optitrack System: Make sure your Optitrack system is properly set up and calibrated. This includes configuring your cameras and placing markers on objects or subjects you want to track.

- Use Matlab to Interface with NatNet SDK: Matlab can interface with external libraries using the loadlibrary and calllib functions. Use these functions to load the NatNet SDK library into Matlab and call its functions to receive streaming data from Optitrack.

- Implement Data Processing: Once you've established a connection and are receiving data in Matlab, you can implement any necessary data processing or analysis. This might include filtering, transforming coordinate systems, or extracting relevant information from the motion capture data.

- Visualize Results (Optional): If desired, you can visualize the motion capture data within Matlab using plotting functions or create custom visualizations. Alternatively, you can export the data to be visualized in other software tools.

- Error Handling and Debugging: Throughout the process, ensure that you handle errors appropriately and debug any issues that arise. This might involve checking for connection errors, ensuring data integrity, and troubleshooting any discrepancies between the expected and received data.

- Testing and Optimization: Once you have a basic implementation working, test it thoroughly to ensure its reliability and accuracy. You may also optimize your code for efficiency, especially if dealing with large datasets or real-time processing requirements.

By following these steps and utilizing the NatNet SDK provided by NaturalPoint, you can connect Optitrack with Matlab and integrate motion capture data into your Matlab-based workflows or applications.

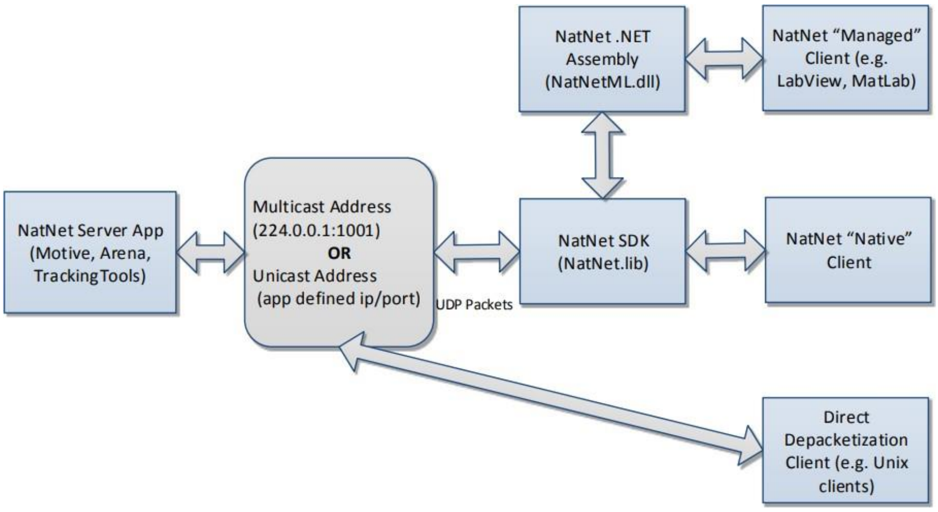


Figure 8: Block diagram of NatNet SDK.

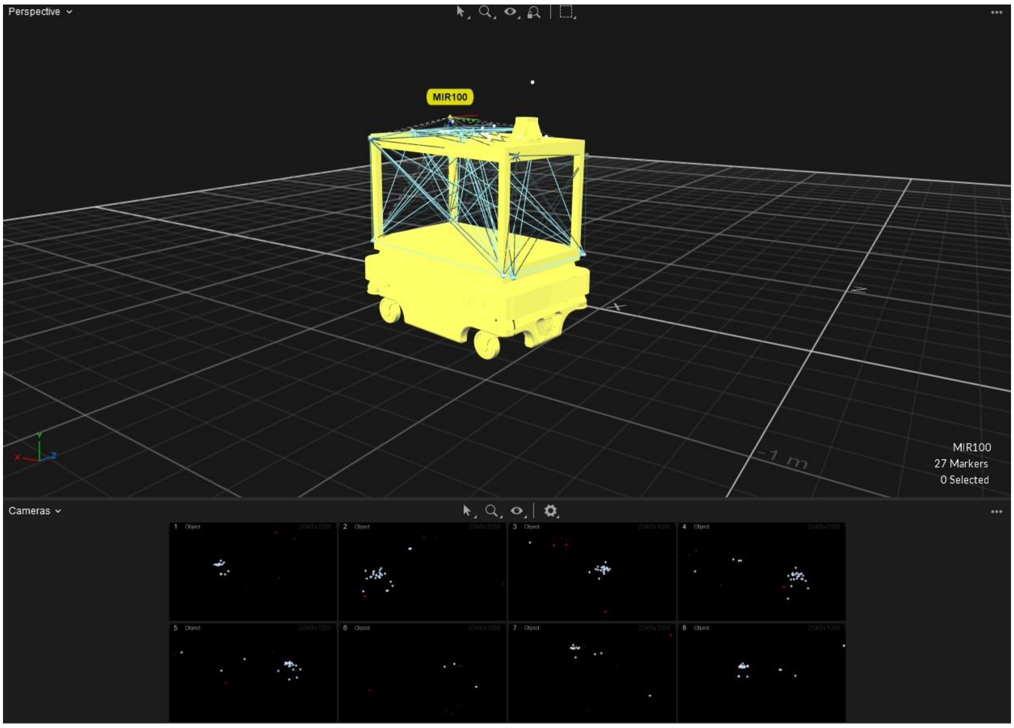


Figure 9: MiR100 in Optitrack Motive system.

2.2.2 Changing coordinate system of Optitrack

Optitrack gives real time positional data on its coordinate system and in millimeters. We tried to change it in order to be the same coordinate system with data given by ROS. The offsets are found by moving the robot to the origin of Optitrack's coordinates system (Figure 10).

There is the way which we used:

X_opt_new= -Z_opt + X_offset

Y_opt_new= -X_opt + Y_offset

X_offset=31.055 m

Y_offset=19.281 m

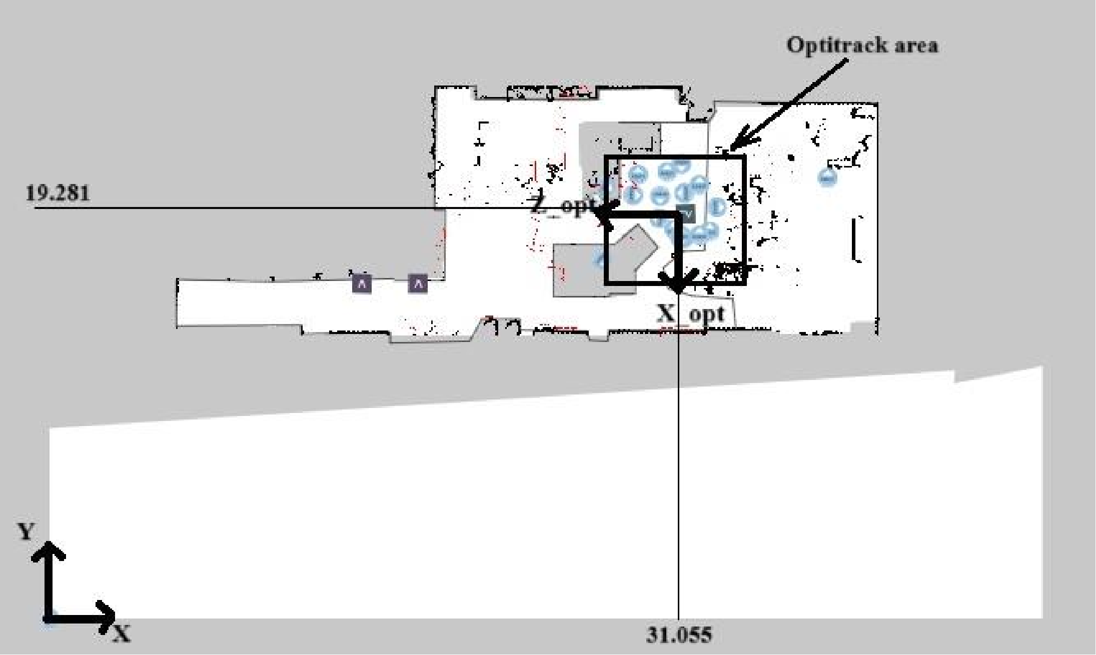


Figure 10: Reference frames of MiR100 and Optitrack

3. Case study

The case study in this thesis focuses on evaluating repeatability and accuracy in mobile robot navigation with and without V Markers. Specifically, the MiR100 is programmed to reach a location called “Destra”, proceed to the next station called “Arrival”, going the next station “D’avanti”, returning again “Arrival”, going the station “Sinistra”, returning the station “Arrival” and return the station “Destra” again.

We built this path in two cases. Firstly, the station “Arrival” is just the point where the robot should reach. In the second case, the station “Arrival” is not point but the marker that the robot should dock it.

In both cases, we should take real-time robot position data from both Optitrack and ROS. Then, we will compare it to check repeatability and accuracy.

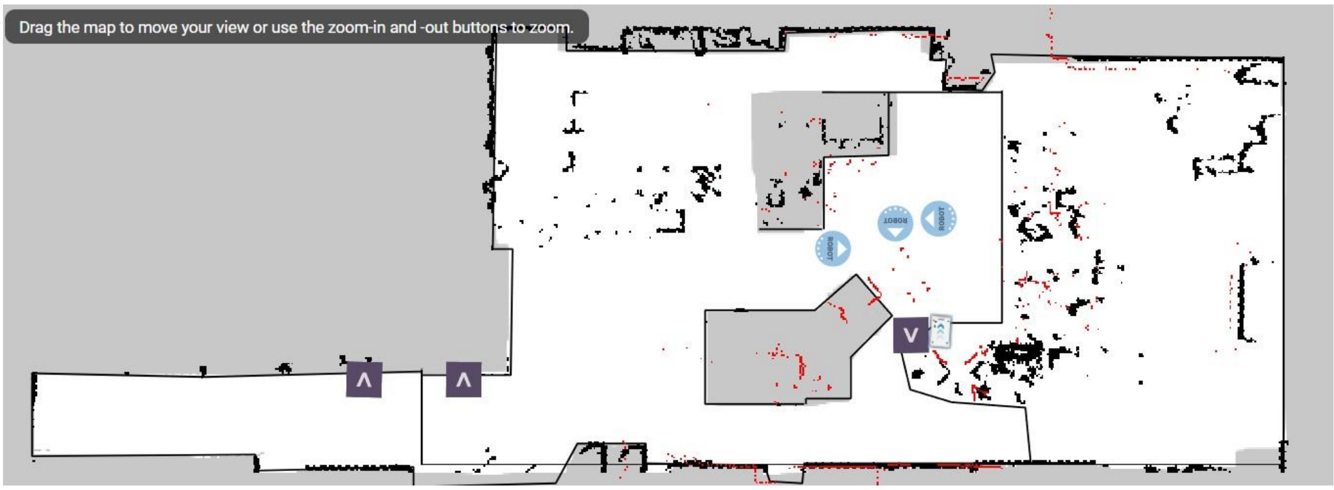
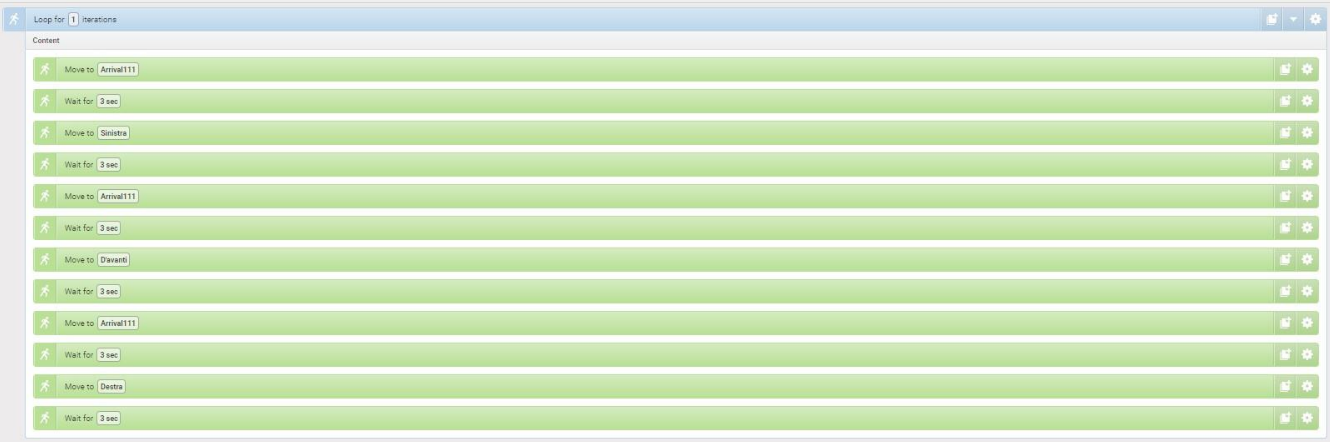
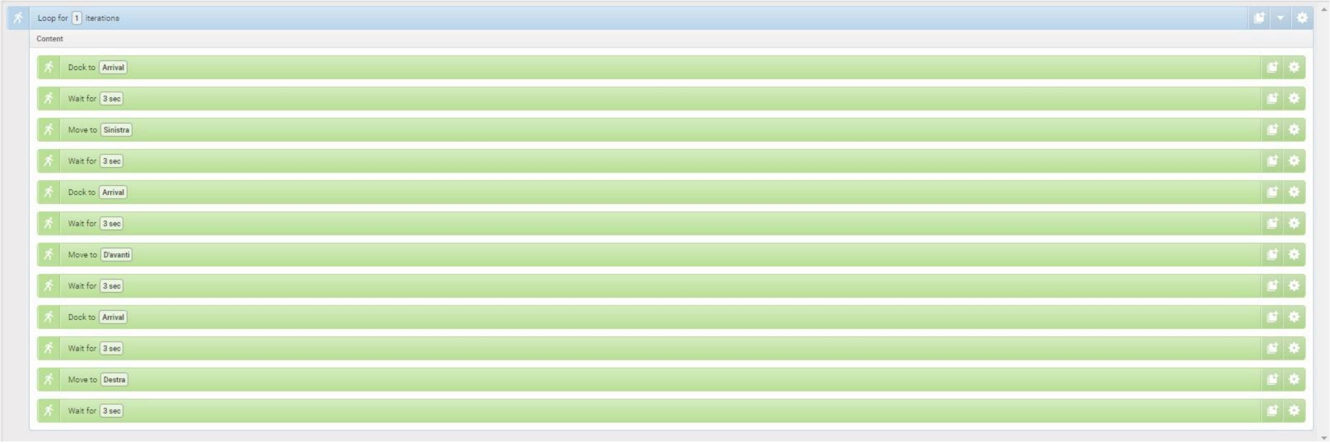


Figure 11: MiR100 map from MiR’s software with stations.



Loop for 1 iterations
Content
Move to Arrival111
Wait for 3 sec
Move to Sinistra
Wait for 3 sec
Move to Arrival111
Wait for 3 sec
Move to D'avanti
Wait for 3 sec
Move to Arrival111
Wait for 3 sec
Move to Destra
Wait for 3 sec

Figure 12: MiR100 program without V-marker



Loop for 1 iterations

Content

- Dock to Arrival
- Wait for 3 sec
- Move to Sinistra
- Wait for 3 sec
- Dock to Arrival
- Wait for 3 sec
- Move to D'avanti
- Wait for 3 sec
- Dock to Arrival
- Wait for 3 sec
- Move to Destra
- Wait for 3 sec

Figure 13: MiR100 program with V-marker

4. Case study results

The main objective of the thesis is to assess the accuracy and repeatability of data collection systems, specifically focusing on ROS (Robot Operating System) and Optitrack. This involves conducting a comprehensive evaluation of these systems to determine how reliably they capture and reproduce data in various scenarios.
The assessment process may include:

- Setup and Configuration: Understanding the setup requirements for both ROS and Optitrack systems. This involves configuring sensors, cameras, and other hardware components to ensure proper functionality.
- Calibration: Ensuring that both systems are properly calibrated to minimize errors in data collection. Calibration procedures may vary between ROS and Optitrack and should be thoroughly documented.
- Data Collection: Performing data collection experiments using both systems in controlled environments. These experiments may involve different types of sensors (such as cameras, LiDAR, IMUs) and objects of interest.
- Analysis of Collected Data: Analyzing the data collected by both systems to evaluate their accuracy and repeatability. This may involve comparing the captured data against ground truth measurements or known values.
- Repeatability Testing: Conducting multiple trials of the same experiment to assess the repeatability of data collected by each system. This helps determine the consistency of results over multiple runs.
- Error Analysis: Identifying and analyzing sources of error in data collection, such as sensor noise, calibration inaccuracies, or environmental factors.
- Comparison and Conclusion: Comparing the performance of ROS and Optitrack systems based on the assessment metrics. Drawing conclusions regarding the strengths and weaknesses of each system in terms of accuracy and repeatability.
- Recommendations: Providing recommendations for improving the accuracy and repeatability of both systems based on the findings of the assessment.

Overall, the objective is to provide valuable insights into the capabilities of ROS and Optitrack as data collection systems, helping researchers and practitioners make informed decisions when choosing between them for their specific applications.

The proposed experiment entails conducting twelve successive repetitions of an identical trajectory to thoroughly assess the system's performance. Specifically, it involves examining the MiR100's position upon reaching the "Arrival" point in two cases, as previously indicated, in order to evaluate consistency and accuracy across multiple trials.

4.1 Repetitions without V-marker

The analysis is conducted by first evaluating the accuracy of the systems used for data acquisition and then evaluating their reliability in terms of uncertainty.

4.1.1 Accuracy

The positional differences from reference point are represented in *Figure 13*. At the station “Destra”, the ROS system gives some data inside the black circle, which represents positional accuracy of MiR given in the mir datasheet. Looking at the station “Sinistra”, there is more than half of the data given by ROS system are in 0.05 range. As for the most important station “Arrival”, there is almost all of the positional data points are inside the black circle.

Regarding to the Optitrack system, at the stations “Destra” and “Sinistra”, we can see that almost all of the positional data points are is in range 0.1 m. However, at the station “Arrival”, noticeable amount of points are inside the black circle.

STATIONS	X [m]	Y [m]
Destra	31.496	21.077
Arrival	30.843	18.675
Sinistra	28.552	20.239

Table 1: Reference positions of stations from MiR100’s map.

DISTANCE FROM REFERENCE POINT [m]		
STATIONS	**ROS**	**Optitrack**
Destra	0.046	0.067
Arrival	0.045	0.063
Sinistra	0.054	0.075

Table 2: Average distances from reference points.

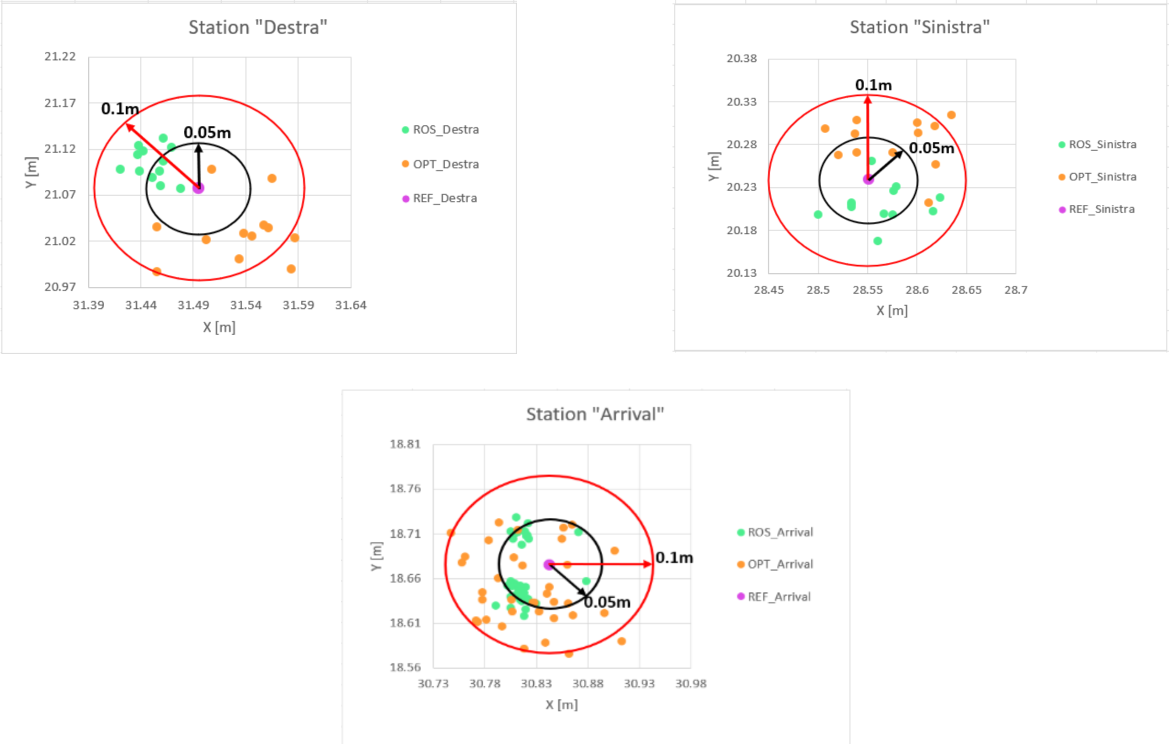


Figure 14: Charts of repetition experiment for the accuracy evaluation

4.1.2 Uncertainty

The uncertainty analysis for the two data acquisition systems involves assessing their extended uncertainties (U) through rigorous statistical examination. Utilizing the T-Student distribution is essential in this process, especially considering the limited availability of data samples. By applying this statistical method, we can thoroughly evaluate the variability and reliability of measurements obtained from both systems, providing a comprehensive understanding of their performance characteristics. This meticulous analysis enables us to gain valuable insights into the robustness and accuracy of the data collected, enhancing the overall reliability of the study's findings.

The degrees of freedom (DoF) for both acquisition systems are calculated based on the number of available samples, denoted as n-1. With a confidence interval set at 95%, the confidence factor (k) is computed using the inverse of the T-Student distribution in Excel, yielding a value of 2.03, in line with the GUM table standards. The standard uncertainty (u) for each system is determined through covariance, which involves a linear combination of two factors: the instrument's standard uncertainty and the standard deviation (σ) relative to the mean of the acquired data.

To delve deeper into this process, let's elaborate on how covariance plays a pivotal role in assessing the standard uncertainty. It not only accounts for the inherent uncertainty associated with the instrument itself but also considers the variability observed in the acquired data relative to its mean value. This comprehensive approach ensures a more robust estimation of uncertainty, vital for maintaining accuracy and reliability in measurement systems.

The formula used is provided below:

$$u^2 = \sqrt{\sigma^2 + {u_{sys}}^2}$$

Regarding the MiR system, a confidence factor (k) of 2 has been selected, following the guidelines of the PUMA (Procedure for Uncertainty of Measurement Management) method. This choice is informed by the consideration of degrees of freedom (dof) set at 100 and a confidence interval of 95%. The decision to opt for this confident factor stems from a deep trust in the positioning accuracy value provided in the MiR datasheet, indicating a high level of assurance in the manufacturer's specifications. This meticulous approach underscores the significance of ensuring accuracy and reliability in the assessment of measurement uncertainties. By adhering to established methodologies and leveraging the manufacturer's data, the analysis aims to provide a robust framework for evaluating the system's performance, thereby fostering greater confidence in the obtained results.

The extended uncertainty is computed by using of the following formula:

$$U = \pm ku = \pm k\frac{ME}{\sqrt{3}}$$

When analyzing the ROS systems, the standard deviation of the MiR system serves as a key metric. Conversely, in the case of Optitrack, the previously calculated value, which stands at 0.307mm, is utilized for assessment. This distinction highlights the tailored approach taken in evaluating the uncertainties associated with each system. Expanding upon this, it's crucial to recognize that different acquisition systems may have distinct sources of uncertainty. For the MiR system within the ROS framework, the variability inherent in its standard deviation plays a significant role in determining overall uncertainty. This encompasses factors such as sensor accuracy, environmental conditions, and potential sources of noise within the system. On the other hand, Optitrack's uncertainty assessment relies on the previously calculated value, emphasizing a meticulous consideration of its specific characteristics and performance metrics. This could include factors such as calibration accuracy, marker placement precision, and system calibration stability over time.

Finally, for this analysis, the mean values of the x and y coordinates from the twelve repetitions are considered separately.

As mentioned, we did twelve repetitions and one path makes robot to go the point "Arrival" three times. So, we collect data 36 times in 12 repetitions. Consequently, the degree of freedom will be 35 for the experiment.

The results obtained are presented in the table 3.

Information of MiR100:

Confident factor (k): 2

Degrees of freedom: 100

Positioning accuracy (ME): 0.05 m

Standard uncertainty (u): 0.029 m

Extended uncertainty (U): 0.058 m

Arrival point without V-marker				
	X coordinate		**Y coordinate**	
	ROS	**Optitrack**	**ROS**	**Optitrack**
Degrees of freedom	35	35	35	35
Confident factor (k)	2.03	2.03	2.03	2.03
Standard deviation (σ) [m]	0.016	0.042	0.035	0.043
Standard uncertainty (u) [m]	0.033	0.042	0.045	0.043
Extended uncertainty (U) [m]	0.067	0.085	0.091	0.087
Mean values of samples [m]	30.817	30.823	18.668	18.648

Table 3: Uncertainties at the station "Arrival" without V-marker.

The charts provided offer a means to evaluate the compliance of acquisition systems, particularly concerning the MiR system. They present the average values and their corresponding extended uncertainties for each system and station. If a value, together with its uncertainty range, falls entirely within the highlighted red limits (representing the extended uncertainty of the reference station coordinate from the MiR system), the system is considered compliant with MiR specifications. On the flip side, when a notable portion of the uncertainty range of a system aligns with the MiR system, it suggests that the employed system is probably compliant, though without absolute assurance at a 95% level of confidence. In all other instances, it is probable that the system is not compliant. This analysis aids in ensuring adherence to MiR specifications and promotes confidence in system performance.

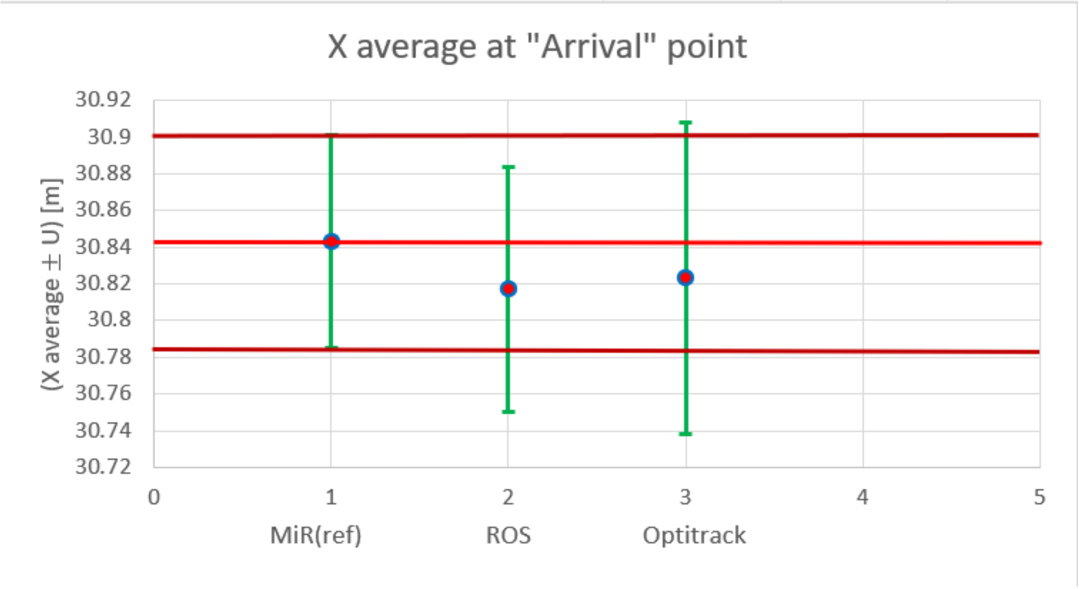


Figure 15: X average values with extended uncertainties at the station "Arrival".

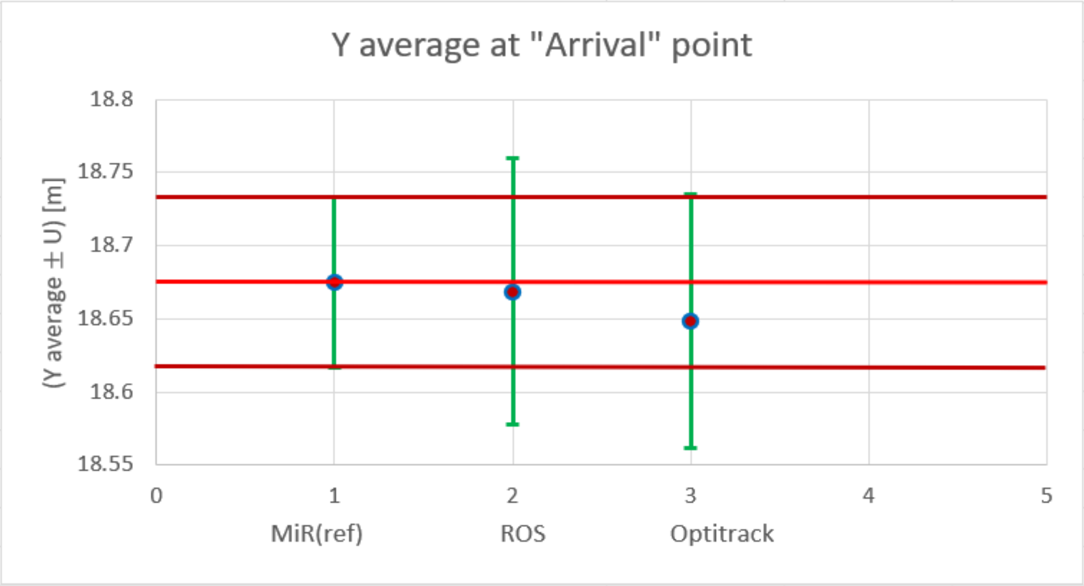


Figure 16: Y average values with extended uncertainties at the station "Arrival".

4.2 Repetitions with V-marker

The same experiment was carried out to test repeatability with V-marker of MiR100 at the station of “Arrival”. To be precise, 12 repetitions were analysed again in the same path among the stations “Destra”, “Sinistra”, “D’avanti” and “Arrival”. In this case, station “Arrival” is not just point that robot move but it is V-marker that robot can dock it.

4.2.1 Accuracy

The same experiment was repeated 12 times again but with V-marker at the station of “Arrival”. So, there is no noticeable changes at the stations of “Destra” and “Sinistra”. Both of the data provided by ROS system and Optitrack are almost similar to the graphs given in section 4.1.1. It is because, in this experiment we used V-marker only at the station of “Arrival”.

As for the station of “Arrival”, we can observe that there is noticeable change in positional data points provided by both ROS and Optitrack. Almost all data points are inside 0.05 m range and more consistent compared to data points in section 4.1.1.

DISTANCE FROM REFERENCE POINT [m]		
STATIONS	**ROS**	**Optitrack**
Destra	0.051	0.078
Arrival	0.030	0.042
Sinistra	0.037	0.069

Table 4: Average distances from reference points.

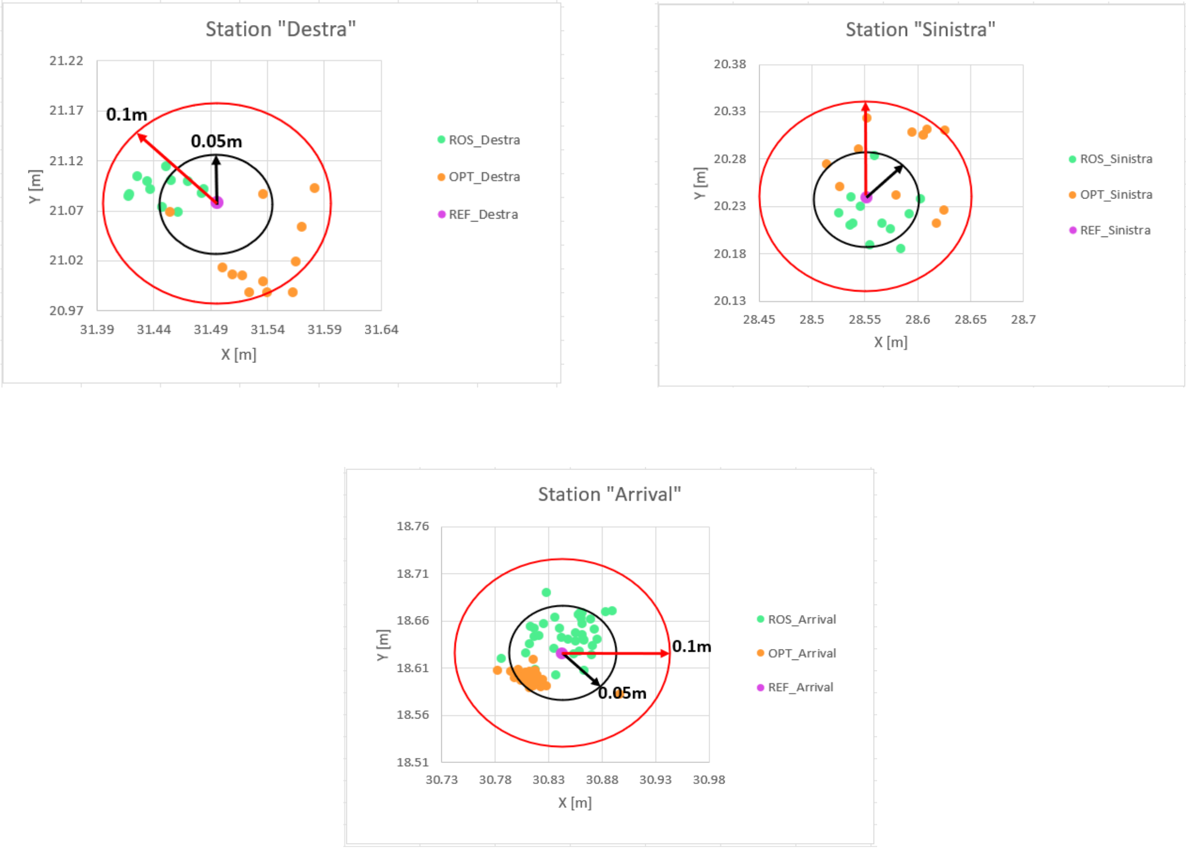


Figure 17: Charts of repetition experiment for the accuracy evaluation with V-marker

4.2.2 Uncertainty

Using the same formulas and calculation methods, the values below were calculated. Notice that the robot moves to dock the station “Arrival” 3 times in each repetition and experiment was carried out 12 times, degrees of freedom (dof) is 35 for this case also.

Arrival point with V-marker				
	X coordinate		**Y coordinate**	
	ROS	**Optitrack**	**ROS**	**Optitrack**
Degrees of freedom	35	35	35	35
Confident factor (k)	2.03	2.03	2.03	2.03
Standard deviation (σ) [m]	0.024	0.016	0.019	0.007
Standard uncertainty (u) [m]	0.037	0.016	0.034	0.007
Extended uncertainty (U) [m]	0.075	0.032	0.069	0.014
Mean values of samples [m]	30.846	30.813	18.645	18.599

Table 5: Uncertainties at the station “Arrival” with V-marker.

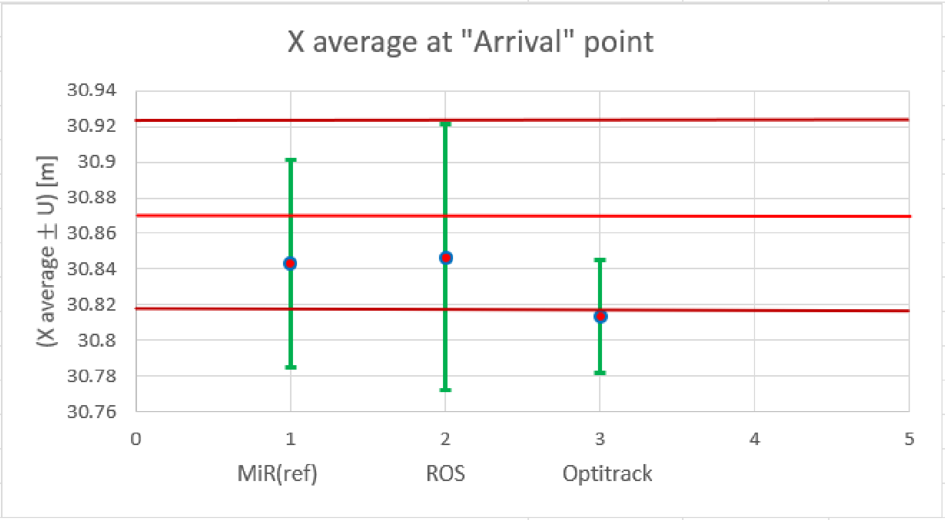


Figure 18: X average values with extended uncertainties at the station "Arrival" with V-marker.

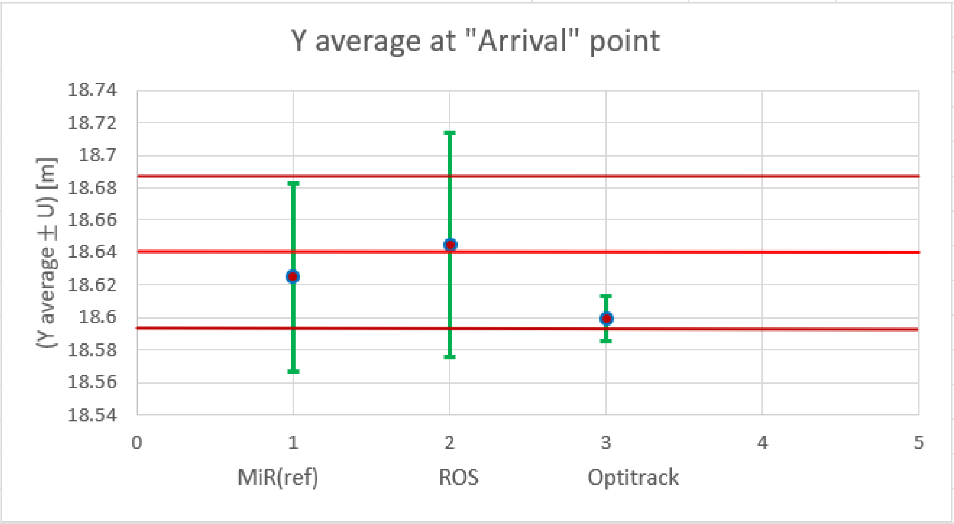


Figure 19: Y average values with extended uncertainties at the station "Arrival" with V-marker.

5. Conclusion

The experiment conducted to evaluate the repeatability of the MIR100 robot with and without the utilization of the V marker yielded significant insights into its performance within an industrial environment. The findings of this study underscore the pivotal role played by the V marker in enhancing the accuracy and precision of the robot's navigation and positioning capabilities.

When comparing the data collected under two conditions—utilizing the V marker and operating without it—the results unmistakably indicate a notable improvement in accuracy when the V marker is employed. This improvement can be attributed to several factors inherent in the functionality of the V marker within the robot's navigation system.

First and foremost, the V marker serves as a reliable reference point that aids the robot in precisely determining its position within the environment. By leveraging the V marker's unique characteristics and its integration into the navigation algorithms of the MIR100, the robot can effectively localize itself with heightened accuracy. Consequently, this facilitates more consistent and repeatable performance across various tasks and operating conditions.

Furthermore, the utilization of the V marker contributes to reducing uncertainties associated with environmental factors, such as variations in lighting conditions or the presence of obstacles. By providing a stable and consistent reference frame, the V marker enables the robot to navigate with greater confidence and reliability, thereby minimizing deviations in its trajectory and improving overall repeatability.

The findings of this study have significant implications for the practical deployment of the MIR100 robot in industrial settings. By leveraging the benefits afforded by the V marker, manufacturers and operators can enhance the efficiency and productivity of their operations while ensuring the consistent and reliable performance of the robot.

In conclusion, the experiment demonstrates that the incorporation of the V marker significantly enhances the repeatability of the MIR100 robot, resulting in more accurate and consistent navigation and positioning. Moving forward, further research and development efforts can focus on optimizing the integration of the V marker into the robot's navigation system to unlock its full potential in industrial applications.

6. Bibliography

- Mobile industrial robot's website, mobile-industrial-robot.com
- Datasheet of MiR100, datasheet
- Datasheet of positional markers, Alfatest datasheet
- Optitrack official website, optitrack.com
- Optitrack support, naturalpoint.com
- ISO GUM: Guide to the experession of uncertainty
- NatNet User's Guide, NatNet User's Guide
- ROS driver for MiR100, GitHub - DFKI-NI/mir_robot
- Matlab ROS Toolbox documentation, ROS Toolbox
- Thesis of Giuseppe La Spisa, Design and evaluation of different digital control technologies of a mobile manipulator
- Thesis of Marco Soriano, Design and fidelity evaluation of a Digital Twin of a Mobile Manipulator
- ROS website, ros.org