



NIST Advanced Manufacturing Series NIST AMS 100-57

Continuous Mobile Manipulator Performance Measurement Data

Omar Aboul-Enein
Henry Medeiros¹
Mili Shah²
Ya-Shian Li-Baboud
Roger Bostelman
Ann Virts

This publication is available free of charge from:
<https://doi.org/10.6028/NIST.AMS.100-57>



¹Work performed under financial assistance award: 70NANBI8H259 from U.S. Department of Commerce, National Institute of Standards and Technology.

²Work performed under financial assistance award: 70NANB23H248 from U.S. Department of Commerce, National Institute of Standards and Technology.

NIST Advanced Manufacturing Series NIST AMS 100-57

Continuous Mobile Manipulator Performance Measurement Data

Omar Aboul-Enein
*Intelligent Systems Division
Engineering Laboratory*

Henry Medeiros
*University of Florida
Gainesville, FL*

Mili Shah
*The Cooper Union
New York, NY*

Ya-Shian Li-Baboud
*Software and Systems Division
Information Technology Laboratory*

Roger Bostelman
*Smart HLPR, LLC
Frederick, MD*

Ann Virts
*Intelligent Systems Division
Engineering Laboratory*

This publication is available free of charge from:
<https://doi.org/10.6028/NIST.AMS.100-57>

January 2024



U.S. Department of Commerce
Gina M. Raimondo, Secretary

National Institute of Standards and Technology
Laurie E. Locascio, NIST Director and Under Secretary of Commerce for Standards and Technology

Certain commercial entities, equipment, or materials may be identified in this document in order to describe an experimental procedure or concept adequately. Such identification is not intended to imply recommendation or endorsement by the National Institute of Standards and Technology, nor is it intended to imply that the entities, materials, or equipment are necessarily the best available for the purpose.

NIST Technical Series Policies

[Copyright, Fair Use, and Licensing Statements](#)

[NIST Technical Series Publication Identifier Syntax](#)

Publication History

Approved by the NIST Editorial Review Board on 2024-01-16

How to cite this NIST Technical Series Publication:

Omar Aboul-Enein, Henry Medeiros, Mili Shah, Ya-Shian Li-Baboud, Roger Bostelman, Ann Virts (2024) Continuous Mobile Manipulator Performance Measurement Data. (National Institute of Standards and Technology, Gaithersburg, MD), NIST AMS 100-57. <https://doi.org/10.6028/NIST.AMS.100-57>

NIST Author ORCID iDs

Omar Aboul-Enein: 0000-0002-5361-4589

Henry Medeiros: 0000-0002-7704-5587

Mili Shah: 0000-0003-1789-6985

Ya-Shian Li-Baboud: 0000-0003-3234-4345

Roger Bostelman: 0000-0002-8605-7758

Ann Virts: 0009-0003-3420-4888

Contact Information

omar.aboul-enein@nist.gov

Abstract

Mobile manipulators, which are robotic systems integrating an automatic or autonomous mobile base with a manipulator, can potentially enhance automation in many industrial and unstructured environments. Namely, large-scale manufacturing processes, typical in the aerospace, energy, transportation, and conformal additive manufacturing fields, encompass a notable subset of potential future mobile manipulator use-cases. Utilizing autonomous mobility for manipulator re-positioning could allow for continuous simultaneous arm and mobile base cooperation, which is referred to as continuous performance. Continuous mobile manipulator capabilities may hold particular benefit for large, curved, and complex workpieces. However, such flexibility can also introduce additional sources of performance uncertainty, preventing mobile manipulators from satisfying stringent pose repeatability and accuracy requirements. To identify and quantify this uncertainty, the Configurable Mobile Manipulator Apparatus was developed by the National Institute of Standards and Technology. Previous test implementations with the apparatus included non-continuous mobile manipulator performance, such as static and indexed performance, but continuous performance measurement had been demonstrated previously only in simulation and on proof-of-concept hardware. The work presented in this report details the transfer of the latest version of simulations and algorithms for continuous registration to an industrial mobile manipulator platform. Additionally, a 2^3 factorial designed experiment was conducted to produce an accompanied continuous mobile manipulator performance dataset. Analysis of the dataset focused on comparing the performance and robustness of two continuous localization methods: 1) a deterministic spiral search and 2) a stochastic Unscented Kalman Filter search across two selected mobile base speeds and sides of the apparatus.

Keywords

Continuous mobile manipulator performance; coordinate registration; exploratory data analysis; factorial experiment; ground truth; kalman filters; large-scale manufacturing; mobile manipulators; mobile robot; optical tracking system; configurable mobile manipulator artifact; robotics in manufacturing; systems integration.

Table of Contents

1. Introduction	1
2. Platform and Operating Environment	4
2.1. Simulation and Programming Environment	5
2.2. Cart Transporter	5
2.3. Manipulator-on-a-Cart	14
2.4. Type B CMMA	15
2.5. Optical Tracking System	17
2.6. Time Synchronization	18
3. Coordinate System Transformation and Registration Measurements	18
3.1. Cart Transporter Map to OTS Registration	19
3.1.1. Registration Problem Summary	19
3.1.2. Materials and Data Collection	22
3.1.3. Data Pre-processing	23
3.1.4. Registration Measurement Results	29
3.2. Robot Model Transformations	32
3.2.1. Measurement Problem Summary	32
3.2.2. Data Collection Procedure	34
3.2.3. Data Pre-processing and Results	36
3.2.4. Updated Robot Model	43
3.3. Fiducial Ground Truth Measurements	43
4. Experiment Design	48
4.1. Experimental Model	48
4.2. Summary of Collected Data	51
4.3. Procedure	55
5. Analysis and Results	59
5.1. Anomalies	59
5.2. Time Synchronization Offset	62
5.3. Model Adequacy Checks	63
5.3.1. Normality	64
5.3.2. Equal Variance	65

5.4. Exploratory Data Analysis	67
5.4.1. Alternate Variable Coding	67
5.4.2. Ordered Data Plots	67
5.4.3. DOE Scatter Plots	69
5.4.4. Interaction Effects Matrix Plots	72
5.4.5. Block Plots	74
5.4.6. DOE Youden Plots	77
5.4.7. Effects Plots	81
5.4.8. Half Normal Probability Plots	83
5.4.9. Summary of Outcomes and Discussion	83
6. Conclusion	88
References	91
Appendix A. List of Symbols, Abbreviations, and Acronyms	105
Appendix B. Analysis of Variance	107
B.1. Fiducial Interception Rate	107
B.2. Fiducial Search Time	108

List of Tables

Table 1. Commanded position and orientation of the goal points within the AMR-CT map coordinate system used to dock the AMR-CT next to the CMMA.	8
Table 2. List of initial positions for the retro-reflective fiducial targets on the CMMA as measured by the OTS.	46
Table 3. Final list of initial search positions for the retro-reflective fiducial targets. .	47
Table 4. Raw recorded response for each of the runs and replicates of the experiment (1 of 2).	60
Table 5. Raw recorded response for each of the runs and replicates of the experiment (2 of 2).	61
Table 6. Descriptive Statistics for Fiducial Interception Rate By Factor Combination	62
Table 7. Descriptive Statistics for Fiducial Search Time (s) By Factor Combination	62
Table 8. Time synchronization offset of clients prior to each replicate.	63
Table 9. Results for Equality of Variance Test (Levene's) on the samples of fiducial interception rates.	66
Table 10. Results for Equality of Variance Test (Levene's) on the samples of fiducial search times.	66
Table 11. Alternate Coding for Factor Combinations (with Interactions)	68
Table 12. Summary of EDA Step Outcomes for the Fiducial Interception Rate	85
Table 13. Summary of EDA Step Outcomes for the Average Fiducial Search Time .	87

Table 14. Analysis of Variance for comparing the fiducial interception rate.	108
Table 15. Analysis of Variance for comparing the fiducial search time.	109

List of Figures

Fig. 1. The manipulator using the RLS to detect a retro-reflective fiducial [28].	2
Fig. 2. Overview of AMR-CT mobile manipulator hardware components reproduced from Ref. [34]. Includes the manipulator-on-a-cart (left), the AMR-CT docking with the manipulator-on-a-cart (center), and the AMR-CT (right). For this work, the linear actuators were not used and the digital levels were detached for the cart and used only for the calibration experiment presented in Sec. 3.1.	6
Fig. 3. Network and software diagram for the AMR-CT mobile manipulator operating environment.	7
Fig. 4. Annotated AMR-CT map of the lab space, which was configured with goal points, forbidden zones (to avoid collision with the CMMA), and an area to ignore tilted sensors. Note that the map coordinate system origin is also visible and annotated.	9
Fig. 5. ROS graph showing node and topics for ROS packages used with the Marquette University mobile manipulator platform reproduced from Ref. [10].	10
Fig. 6. Annotated ROS graph showing the nodes and topics that were removed in the process of integrating the code with the NIST AMR-CT mobile manipulator platform.	11
Fig. 7. The Type B Configurable Mobile Manipulator Apparatus (CMMA) configured with 12 retro-reflective fiducials (eight 6.4 mm diameter fiducials and four 3 mm diameter fiducials) and additional objects placed underneath to improve AMR-CT localization.	16
Fig. 8. Example screenshot showing how the OTS computer time synchronization was verified via command line.	19
Fig. 9. Example screenshot showing how the time synchronization of the control laptop Linux subsystem was verified.	20
Fig. 10. Diagram of homogeneous transformations in different coordinate systems needed to calibrate the unknown coordinate transformation between the AMR-CT map and the OTS (left). Simplified graph showing displaying the same information (right). Note that the double arrows imply the inverse of the respective labeled transformation, though not shown, can also be applied.	21
Fig. 11. Digital levels mounted to the base of the AMR-CT using adhesive strips (left). Example stop label indicating the stop number, as well as the approximate location and orientation of the AMR-CT at a level location in the lab (right).	23
Fig. 12. Quiver plot showing the location and heading of the AMR-CT measured using the OTS at all seven stops selected for measurement in the lab. The plot has been annotated to show the assigned stop numbers.	24

Fig. 13.	Marker plot of the VEHICLE_BASE rigid body created for obtaining OTS data needed to measure the unknown coordinate transformation between the AMR-CT map and the OTS (left). Corresponding picture of the physical marker placement on the AMR-CT (right).	25
Fig. 14.	Boxplots of the component-wise standard deviations for each marker of the VEHICLE_BASE rigid body and for each data capture obtained during the AMR-CT map to OTS coordinate transformation calibration experiment. .	26
Fig. 15.	Scatterplots showing the estimated height of the AMR-CT base centroid across all 21 data captures by run-order (top) and grouped by stop number (bottom).	28
Fig. 16.	Scatterplot of localization score vs stop number as determined by the AMR-CT controller for all 21 data captures.	29
Fig. 17.	Scatterplot of the front (top) and side (bottom) digital levels angle readings for all 21 data captures.	30
Fig. 18.	Scatterplot comparing the AMR-CT controller points after applying the 6DoF (top) and 3DoF (bottom) calibrated coordinate transformations with the position of the AMR-CT as measured by the OTS.	31
Fig. 19.	Box plots showing the error between the AMR-CT controller measured positions after applying the 6DoF (top) and 3DoF (bottom) calibrated coordinate transformations and the position of the AMR-CT as measured by the OTS. The far-left boxplot presents the error as the Euclidean distance (in mm) between corresponding points while the other three boxplots presents the component-wise error as the difference between the AMR-CT controller measured positions and the OTS measured positions.	33
Fig. 20.	Diagram of homogeneous transformations needed to determine the rigid transformation between the AMR-CT base and the manipulator base. Due to the cart payload structure obstructing the AMR-CT base while attached, the transformation was measured in two parts: The transformation between the AMR-CT base and the AMR-CT frame (left) and the transformation between the AMR-CT frame and the manipulator base (right).	35
Fig. 21.	Marker plot of the VEHICLE_BASE rigid body created for obtaining OTS data needed to calibrate the unknown rigid transformation between the AMR-CT base and the manipulator base (left). Corresponding picture of the physical marker placement on the AMR-CT (right). Note the slight difference in marker labeling in comparison to Fig. 13.	36
Fig. 22.	Marker plot of the VEHICLE_FRAME rigid body created for obtaining OTS data needed to calibrate the unknown rigid transformation between the AMR-CT base and the manipulator base (left). Corresponding picture of the physical marker placement on the AMR-CT (right).	37
Fig. 23.	Marker plot of the CART_BASE rigid body created for obtaining OTS data needed to calibrate the unknown rigid transformation between the AMR-CT base and the manipulator base (left). Corresponding picture of the physical marker placement on the cart (right).	38

Fig. 24. Boxplots of the component-wise standard deviations for each marker of the VEHICLE_BASE rigid body and for each data capture obtained during the rigid transformation measurement.	39
Fig. 25. Boxplots of the component-wise standard deviations for each marker of the VEHICLE_FRAME rigid body and for each data capture obtained during Set 1 of the rigid transformation measurement.	40
Fig. 26. Boxplots of the component-wise standard deviations for each marker of the VEHICLE_FRAME rigid body and for each data capture obtained during Set 2 of the rigid transformation measurement.	41
Fig. 27. Boxplots of the component-wise standard deviations for each marker of the CART_BASE rigid body and for each data capture obtained during Set 2 of the rigid transformation measurement.	42
Fig. 28. Output of the ROS <code>view_frames</code> transform debugging tool for updated robot model.	44
Fig. 29. Scatterplot showing the measured ground truth initial positions of the retro-reflective fiducials, as measured by the OTS. Note that markers M1 and M11 were included to introduce rigid body asymmetry to improve tracking quality.	46
Fig. 30. Bar chart showing the percentage of occlusions for each of five, 10 s OTS captures of the ground truth fiducial positions on the CMMA.	48
Fig. 31. Boxplots showing the component-wise standard deviation per OTS marker on the CMMA.	49
Fig. 32. Screenshot of OTS rigid bodies tracked during the experiment for ground truth comparison.	55
Fig. 33. The EOAT with six 19 mm diameter OTS markers attached to form a rigid body.	56
Fig. 34. The CMMA with 13 19 mm diameter OTS markers attached to form a rigid body.	57
Fig. 35. Cart base rigid body configuration. Note that the configuration is identical to Fig. 23 except for the marker labeling order.	58
Fig. 36. Bar chart showing the time synchronization offset between the clients synchronized to the NIST time servers over NTP. The offsets were measured prior to each replicate.	64
Fig. 37. Q-Q Plot for fiducial interception rate.	65
Fig. 38. Q-Q Plot for fiducial search times.	66
Fig. 39. Ordered data bar graph for the fiducial interception rate comparing the raw response against the factor combination used for each run.	70
Fig. 40. Ordered data bar graph comparing the raw response (i.e., the average fiducial search time) against the factor combination used for each run. . .	71
Fig. 41. DOE scatter plot comparing the raw fiducial interception rate (top) and average fiducial search time (bottom) across the three main factor levels. The center line plots the grand mean for the respective response variable. . .	73

Fig. 42.	Interaction effects matrix plot for the average response (i.e., the fiducial interception rate). Note that the on-diagonal subplots show the change in average response across factor levels for each of the three main factors while the off-diagonal plots show the change in average response for 2-factor interactions. The horizontal line of each plot shows the grand mean of the fiducial interception rates.	75
Fig. 43.	Interaction effects matrix plot for the average response (i.e., the average search time). Note that the on-diagonal subplots show the change in average response across factor levels for each of the three main factor while the off-diagonal plots show the change in average response for 2-factor interactions. The horizontal line of each plot shows the grand mean of the average search times.	76
Fig. 44.	Block plot for the fiducial interception rate. Note that each block is annotated with its respective length, as well as the corresponding high and low settings of the factor of interest.	78
Fig. 45.	Block plot for the average fiducial search time. Note that each block is annotated with its respective length, as well as the corresponding high and low settings of the factor of interest.	79
Fig. 46.	The DOE Youden plot for the fiducial interception rate (top) and average fiducial search time (bottom).	80
Fig. 47.	Effects plots for the fiducial interception rate (top) and average fiducial search times (bottom).	82
Fig. 48.	Half normal probability plot for the fiducial interception rate (top) and average fiducial search time (bottom).	84
Fig. 49.	Plot of the AMR-CT base and EOAT position during replicate 3 (runs 3-5). The intercepted fiducial positions are also plotted for reference, with the marks at the origin representing undetected fiducials.	89
Fig. 50.	Zoomed plot of the AMR-CT base and EOAT position during replicate 3 (runs 3-5). The intercepted fiducial positions are also plotted for reference, with the marks at the origin representing undetected fiducials. The zoomed in view shows Side 1 of the CMMA between fiducials 3 and 4.	90

Acknowledgments

The authors thank the following individuals for their contributions to this project. Dr. Soochaeol Yoon, Intelligent Systems Division at NIST, for his assistance with the data collection. Samuel Reed-Weidner, Omron Robotics and Safety Technologies, for his technical consultation on the AMR-CT hardware and software, which included explanation of the AMR-CT pose broadcasting rate. Dr. Jeremy Marvel, Shelly Bagchi, of the Intelligent Systems Division at NIST, Dr. Vinh Nguyen, Department of Mechanical Engineering-Engineering Mechanics at Michigan Tech, and Dr. Jennifer Case, Fermi National Accelerator Laboratory managed by Fermi Research Alliance, LLC, for their general consultation. Dr. Dennis Leber, Statistical Engineering Division at NIST for his consultation on statistical analysis methods, which included the EDA.

1. Introduction

Mobile manipulators (i.e., robotic systems that feature integration of a manipulator with an automatic or Autonomous Mobile Robot (AMR)) hold the potential to benefit a wide variety of industrial use-cases with enhanced manufacturing autonomy and flexibility. In the past, interest has been shown in utilizing mobile manipulators for flexible manufacturing and material handling, such as conveyor and Computer Numerical Control (CNC) tending [1–3]. However, large-scale manufacturing tasks also represent a potentially notable subset of use cases for mobile manipulators. For example, in the aerospace, energy, nautical, and ground transportation sectors, components such as aircraft wings, towers, ship bows, and high speed train bodies feature curved, complex workpieces with expansive build volumes [1, 4–7]. Such parts often require drilling, boring, polishing, sanding, deburring, and paint application or removal [1, 5, 7, 8]. Furthermore, these large-scale, curved, and complex workpieces have stringent manufacturing accuracy requirements [4–6].

Despite continuous mobile manipulation having been noted as particularly challenging in the past [9–13], specific large-scale manufacturing applications that require continuous motion capabilities could benefit from mobile manipulator automation (i.e., where the mobile base and manipulator move simultaneously while physically attached). Note this is in contrast to non-continuous capabilities (i.e., the mobile base and manipulator of a mobile manipulator do not move simultaneously while physically connected) [14, 15]. Examples of such applications may include, but are not limited to non-contact uses, such as spray-painting or visual inspection, or contact uses, such as cleaning, grinding, or troweling plaster [16]. Welding of large-scale structures has also been noted to have improved uniformity when utilizing continuous motion capabilities [9, 17]. Another potentially impactful use-case of the expanded autonomous build volumes offered through mobile manipulation is large-scale conformal Additive Manufacturing (AM). This has been suggested and implemented with non-continuous capabilities, but also, crucially, continuous capabilities for print-in-motion [18–23]. However, insufficient mobile base localization and low pose accuracy /repeatability in unstructured environments are challenges that hinder mobile manipulator adoption for this use-case [2, 18–20, 22–24]. Further note that the works presented in Refs. [16, 22, 23] provide recent examples where Model-Predictive Control (MPC) was applied towards addressing continuous mobile manipulator capabilities.

Mobile manipulator manufacturers, academics, and end-users could all benefit from a standardized test method to identify and quantify sources of mobile manipulator performance uncertainty. Manufacturers could use such standardized test methods in reporting and determining mobile manipulator capability specifications. Furthermore, manufacturers and academics could use such test methods in the development, refinement, and performance comparison of mobile manipulator hardware, software, control algorithms, planning algorithms, localization methods, etc. Finally, end-users could rely on such test methods to verify that a mobile manipulator platform can perform suitably for their specific manufacturing task and that the platform is robust to the environmental conditions of their specific

plant.

To fulfill this need, the National Institute of Standards and Technology (NIST) has developed and conducted performance evaluation activities using a cost-efficient, versatile test artifact, dubbed the Configurable Mobile Manipulator Apparatus (CMMA) (previously referred to as the Re-configurable Mobile Manipulator Artifact (RMMA)). The CMMA provides a mock-workpiece, with machined holes, to evaluate the performance of mobile manipulator peg-in-hole insertion type manufacturing tasks [15, 25–27]. As noted in Ref. [15] and Sec. 2.4, performance evaluation consists of intercepting retro-reflective targets with a Retro-reflective Laser Sensor and Emitter (RLS) mounted to the End-of-Arm Tool (EOAT) of the manipulator, as depicted in Fig. 1. Alignment of the RLS with the retro-reflective targets serves as analog to peg-in-hole insertion type manufacturing tasks, and the target diameter sizes can be selected to reflect desired manufacturing performance tolerance.

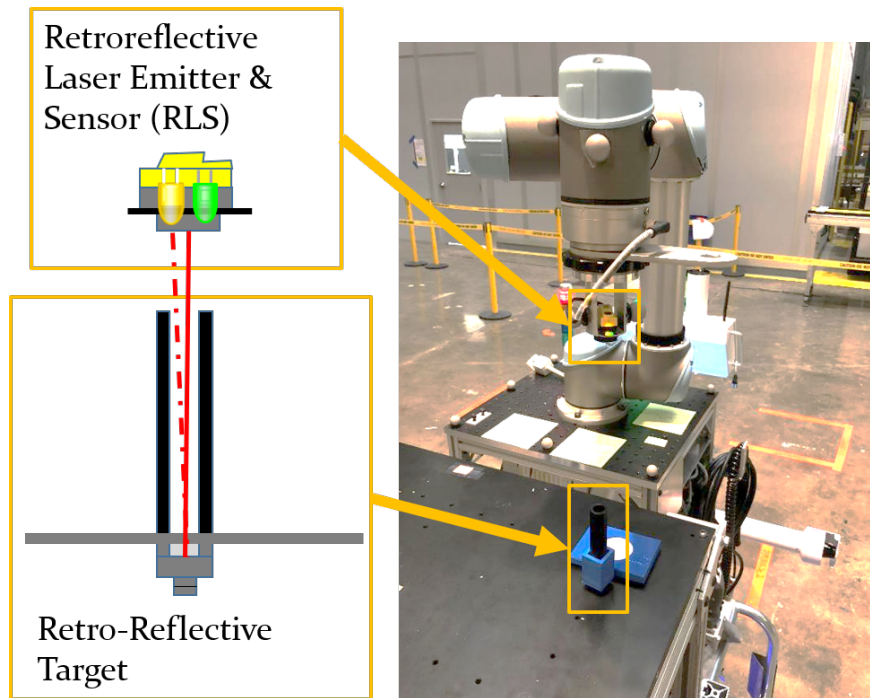


Fig. 1. The manipulator using the RLS to detect a retro-reflective fiducial [28].

Reference [15] contains detailed information pertaining to the design of two CMMA variants: The Type A CMMA³ for representing smaller, non-curved parts and the Type B CMMA⁴ for representing larger parts, with curved, complex edge profiles. Reference [15] also summarizes past evaluation activities, with corresponding full citations to the respective works, conducted using both CMMA variants. This included evaluation of the aforementioned static and index level of mobile manipulator coordination using the Type

³Previously referred to in Ref. [15] as the “Static” RMMA.

⁴Previously referred to as the “Dynamic” RMMA.

A CMMA and tested on a mobile manipulator utilizing 1) an Automatic Guided Vehicle (AGV) base limited to following pre-programmed paths [14, 29–32], 2) an Autonomous Mobile Robot (AMR) base capable of path-planning to avoid obstacles [33], and 3) an Autonomous Mobile Robot Cart Transporter (AMR-CT) base capable of separating from the manipulator to tend to other payloads for increased task concurrency [34]. Furthermore, use of the Type A CMMA to compare the performance of various laser-based and visual marker-based coordinate registration methods and algorithms has been demonstrated [31, 34].

In the past, measurement of continuous mobile manipulator performance (also previously referred to as dynamic performance) was conducted in simulation and on a proof-of-concept mobile manipulator platform using physical and virtual Type B CMMAs [9]. In this work, a Kalman filter-based stochastic search algorithm, proposed for localizing retro-reflective fiducials mounted to the Type B CMMA was compared against a deterministic spiral search method using the CMMA. On the proof-of-concept platform, it was found that the stochastic approach intercepted more fiducials on average: 4.10 fiducials, when compared to the deterministic search, which intercepted 3.86 fiducials on average. The corresponding average search times were 4.18 s and 5.06 s, respectively.

The work in Ref. [9] was later extended to simulation evaluations with a non-linear Unscented Kalman Filter (UKF)-based search method [10]. In simulation, comparisons of the average fiducial interception rate, search time, and AMR position prediction error were conducted across different search methods (i.e., the linear Kalman filter search from Ref. [9], the UKF search, and the deterministic spiral search), three different fiducial arrangements, and increasing AMR base speeds (i.e., from 1 cm/s up to 3 cm/s at increasing 0.5 cm/s intervals). It was found that the UKF-based stochastic search strategy had a higher average fiducial interception rate when compared to the linear Kalman filter average interception rate (e.g., the former had an interception rate of 89.11% as opposed to 71.55% on fiducial arrangement 1 and 91.33% as opposed to 71.11% on arrangement 2). Furthermore, evaluation of these algorithms with the performance tests revealed that the UKF could outperform the spiral search in terms of interception rate and search time given fine-tuned hyper-parameters. However, the optimal set of hyper-parameters varied depending on the fiducial arrangement. For example, the Kalman filter performance was adversely impacted by distance between fiducials, with this degradation possibly attributed to 1) greater motion drift related to longer distances, 2) updating the UKF even when a fiducial was not intercepted, and 3) drift introduced by the manipulator kinematic solvers. It was additionally noted that the fiducial interception rate for the linear Kalman filter exhibited a positive trend with increasing base velocity (due to minimized motion drift), but the UKF exhibited a negative trend with increasing base velocity (due to motion compensated samples being out of reach). For arrangement 1, increasing the initial covariance was found to improve the UKF interception rate. For arrangement 2, improved interception performance was attributed to increasing the sample gap⁵, and for both arrangements 2 and 3, performance

⁵The sample gap is a parameter of the Kalman filter that influences search point density. See Ref. [10] for

improvement was observed from increasing the process covariance [10].

Aside from further improvements to the hyper-parameters and tuning strategy, Ref. [10] leaves the implementation and evaluation of the UKF search strategy on real, commercially-targeted mobile manipulator hardware to future work, which is the focus of the project presented in this report. To this end, this report documents the design, implementation, and execution of a 2^3 factorial experiment that was conducted on the NIST AMR-CT mobile manipulator to compare the continuous mobile manipulator performance, as measured by the fiducial interception rate and the mean fiducial search time (in seconds), between 1) two search strategies (i.e., the stochastic UKF versus the deterministic spiral search), 2) two mobile base speeds (i.e., the “slow” speed of 0.01 m/s and a “fast” speed of 0.025 m/s), and 3) two different sides of the Type B CMMA with similar, but not identical, fiducial arrangements. As the previously cited works in Refs. [9, 10] were conducted by Marquette University under NIST awards 70NANB16H196 and 70NANBI8H259: “Performance Measurement of Mobile Manipulators using Coarse-to-Fine Deep Learning Methods”, respectively, a significant portion of this report focuses on the process of transferring the simulations, algorithms, and code from Marquette University. First, Sec. 2 details the hardware and software configuration and transference of the test implementation. Section 3 details the measurement processes established to acquire unknown coordinate transformations and/or robot model dimensions needed to command the robots. The design of the 2^3 factorial experiment and experimental procedure are presented in Sec. 4. It should be noted that Sec. 4.2 provides a summary of the data-set collected from this experiment and the dataset is publicly available at Ref. [35]. The data analysis and results of the experiment are presented in Sec. 5. The section begins by presenting the results of the statistical model assumption checks, followed by the appropriate non-parametric statistical test for comparing amongst the three aforementioned independent variables. Finally, Sec. 6 concludes with a discussion of potential improvements and future work.

2. Platform and Operating Environment

In general, the work conducted to integrate continuous performance measurement and registration utilized the same AMR-CT mobile manipulator platform⁶ from Ref. [34] and pictured in Fig. 2. The updated network architecture and software used for the NIST AMR-CT mobile manipulator is depicted in Fig. 3. This section will focus on detailing the key differences between the AMR-CT mobile manipulator and the original development mobile manipulator platform utilized at Marquette University and will highlight the required software and configuration changes during the course of the integration in the new environment [9, 10]. Readers are encouraged to refer to Ref. [34] for complete specifications of

details

⁶Certain commercial equipment, instruments, or materials are identified in this paper in order to specify the experimental procedure adequately. Such identification is not intended to imply recommendation or endorsement by the National Institute of Standards and Technology, nor is it intended to imply that the materials or equipment identified are necessarily the best available for the purpose.

the NIST AMR-CT mobile manipulator system utilized in this work.

2.1. Simulation and Programming Environment

Simulations and algorithms were transferred from Marquette University [10] to a NIST-owned laptop with the following specifications: A 6-core processor with a clock rate of 2.6 GHz, 32 GB of random access memory (RAM), a GPU running driver version 23.21.13.8908 and running Windows 10, Version 20H2, Build 19042.1288, 64-bit [36, 37]. Since the simulations and algorithms were previously implemented on a Linux personal computer (PC), version 1 of a Linux subsystem compatibility layer was installed to run the simulations on the NIST-owned laptop [38].

The code was originally written using Robot Operating System (ROS) Melodic Morenia, but recompiled using ROS Noetic Ninjemys [39, 40]. Installation of ROS was completed per [40], though first, it was confirmed that the Linux subsystem was configured to install software from the `restricted`, `universe`, and `multiverse` repositories [41, 42]. Additionally, to build the code, Python 3 version 0.9.2 of the catkin compiler was installed [43–45]. It should be noted that the additional dependency `python3-osrf-pycommon` (version 2.0.2-1) was also needed [46].

Version 11.11.0 of a publicly available simulator was used [47, 48]. Integration of the simulator with ROS required version 2.9.2-1focal.20230216.004103 of the `gazebo-ros-control` package [49]. In order to allow the Linux subsystem to display windowed, graphical applications including but not limited to the simulator, version 1.20.8.1 of an X11 server was installed [50–53]. Additionally, a shared library dependency, `libQt5Core.so.5` failed to load when attempting to run the simulator within the Linux subsystem. This problem was mitigated using the solution from Ref. [54]. Additional packages for installing further dependencies (version 0.22.2-1) and interfacing with ROS through the command line (1.15.8-1focal.20210726.192129) were also installed [55].

2.2. Cart Transporter

The AMR-CT hardware specifications and software⁷ were unchanged from prior experiments [34]. Reference [56] provides the manufacturer specifications for the AMR-CT⁸. However, the manufacturer-supplied map generation and configuration software was upgraded from version 4.5.2 to then-current version 4.7.7 and run on the laptop described in Sec. 2.1 [34, 57, 58]. The AMR-CT utilized laser-based localization and mapping, and a new environmental map of the lab space was created in the same fashion as in previous experiments and according to manufacturer procedure (i.e., manually driving the AMR-CT to scan static room physical features, such as walls) [34, 56, 58]. Unlike the work presented

⁷The on-board AMR-CT software previously included controller firmware version 1.9.0d, manufacturer software OS version 4.8.0, and automation management software version 4.9.9.

⁸Note that the AMR-CT used corresponded to model number LD-105CT and part number 37142-01014.

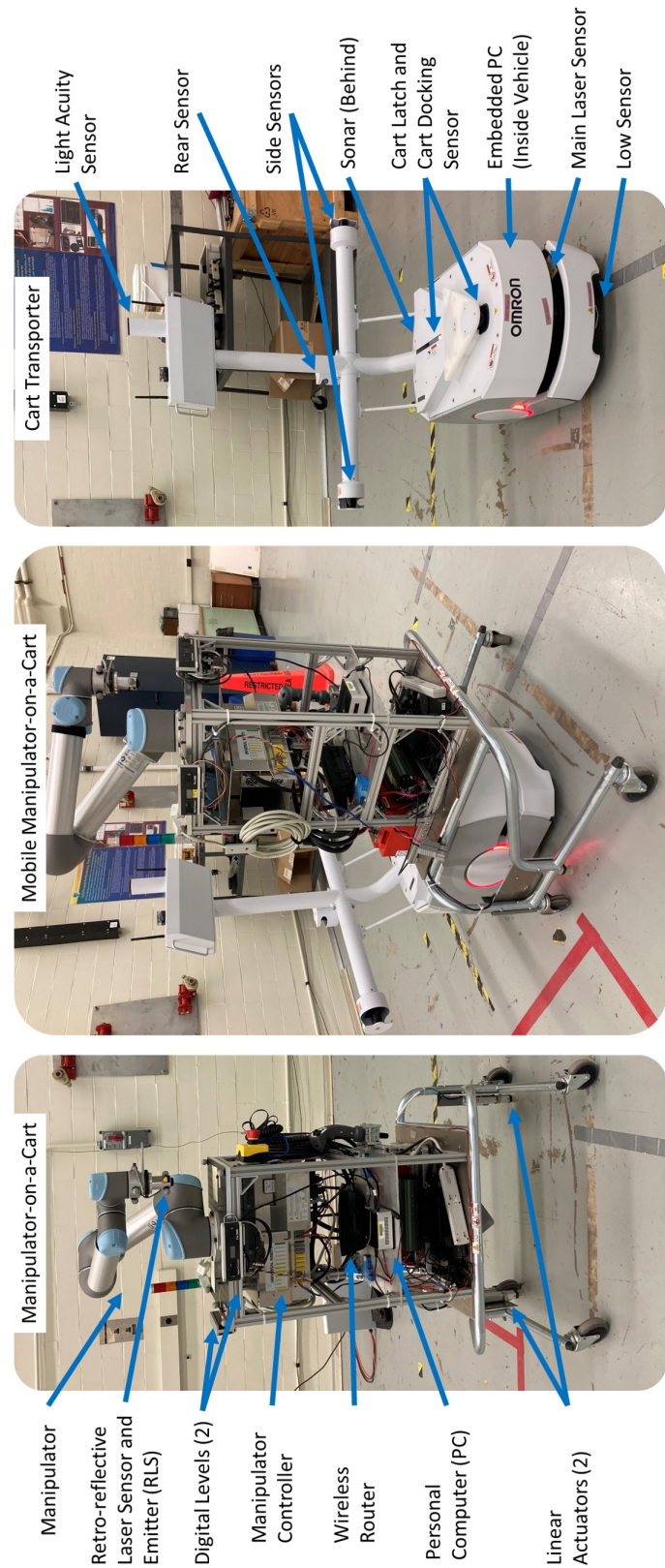


Fig. 2. Overview of AMR-CT mobile manipulator hardware components reproduced from Ref. [34]. Includes the manipulator-on-a-cart (left), the AMR-CT docking with the manipulator-on-a-cart (center), and the AMR-CT (right). For this work, the linear actuators were not used and the digital levels were detached for the cart and used only for the calibration experiment presented in Sec. 3.1.

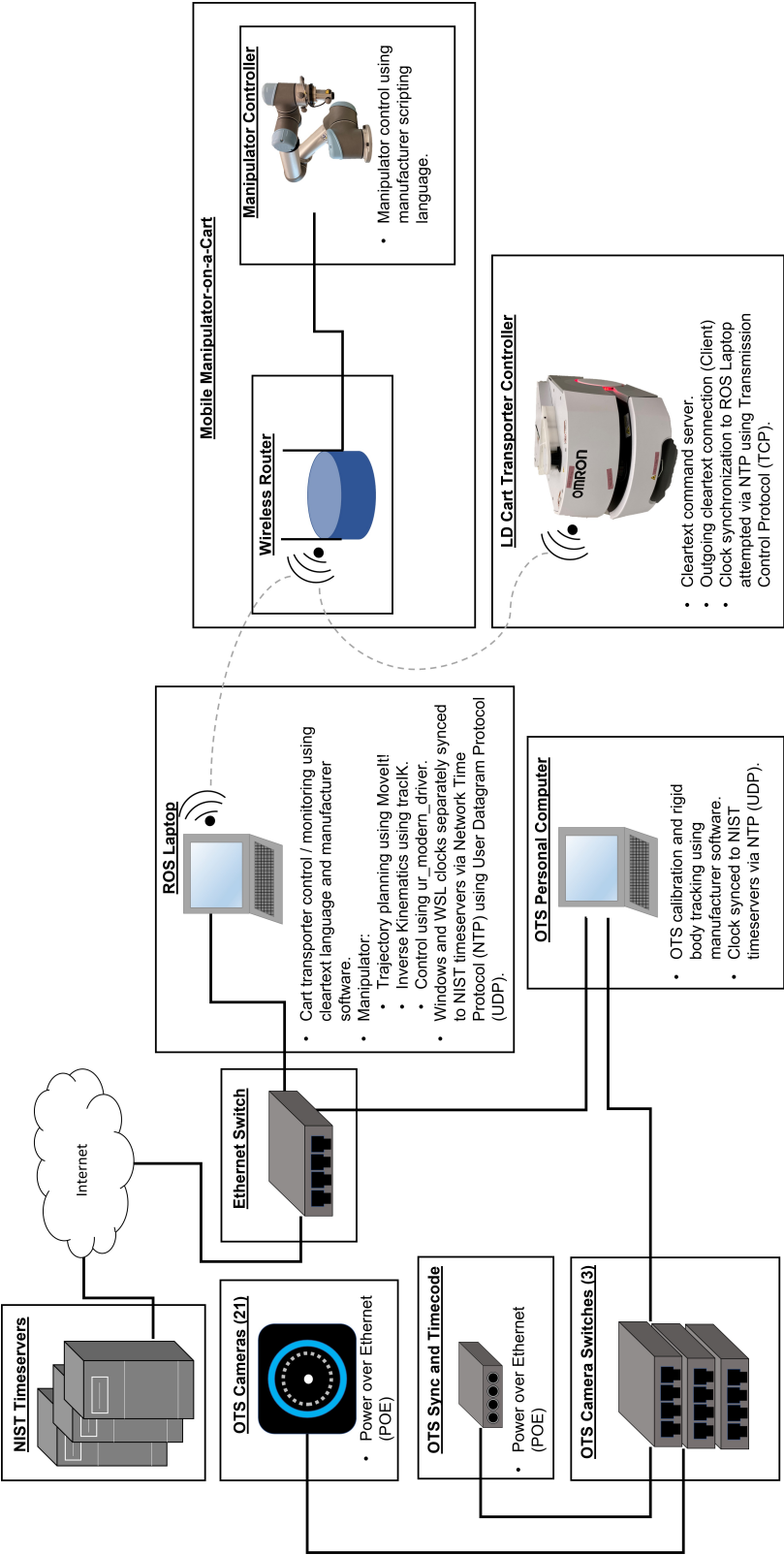


Fig. 3. Network and software diagram for the AMR-CT mobile manipulator operating environment.

in Ref. [34], in which separation between the AMR-CT base and detachable manipulator-on-a-cart was tested, the experiments documented in this report solely tested these two sub-components as a combined system (i.e., without detaching the manipulator-on-a-cart) since the objective was to test continuous and simultaneous coordination. Another difference related to the lab configuration was that additional wood and cardboard objects were placed underneath the Type B CMMA (see Fig. 7 in Sec. 2.4) to provide additional static feature references for the AMR-CT as it traveled alongside the CMMA. Such practice was found to potentially improve AMR-CT localization near the Type A CMMA in prior experiments and according to the manufacturer [33, 58]. Finally, before discussing further customization, it should be noted that the laser scans within the map were updated prior to experimentation to mitigate degradation of AMR-CT localization performance due to any small changes to the lab space that had occurred over time. This was accomplished by creating a new scan of the map and inserting it into the original map using the manufacturer-supplied software [58]. This process preserved the origin of the original map such that the coordinate system calibration detailed in Sec. 3.1 was not invalidated. However, small map alignment errors could have been introduced since the process required manually positioning the new laser scans within the previously generated map.

Figure 4 shows the AMR-CT map that was also customized with goals, forbidden zones, and an area in which the AMR-CT would ignore obstacles detected by tilted sensors. The position and orientation of each goal point is given in Tab. 1. It should be noted that ROS measures positions in meters and angles in radians. However, the AMR-CT controller was configured to specify units in millimeters and degrees, respectively. This report will use the latter units for consistency. The forbidden zones placed near the ends of the CMMA were added to prevent the AMR-CT from colliding with the CMMA when traveling between the starting goal and the two initial CMMA goals. Additionally, to prevent the side scanners of the AMR-CT from detecting the overhanging edges of the CMMA optical breadboard and the manipulator as obstacles, the area in which the AMR-CT would ignore its tilted side-sensors was introduced around the CMMA and the maximum height at which the left and right side scanners could detect obstacles was reduced in the AMR-CT controller configuration from 1000 mm to 250 mm. In addition to the lower front sensor, which was disabled in previous experiments to prevent anomalous obstacle detection that hindered movement, the rear laser scanner was also disabled in the robot configuration [34].

Table 1. Commanded position and orientation of the goal points within the AMR-CT map coordinate system used to dock the AMR-CT next to the CMMA.

Goal	x	y	θ
DynamicRMMA1	5174 mm	1108 mm	90°
DynamicRMMA2	3171 mm	6629 mm	-90°
LDStart	1613 mm	3443 mm	0°
TestPoint	5199 mm	-2096 mm	90°

A major difference between the AMR-CT mobile manipulator system used at NIST and

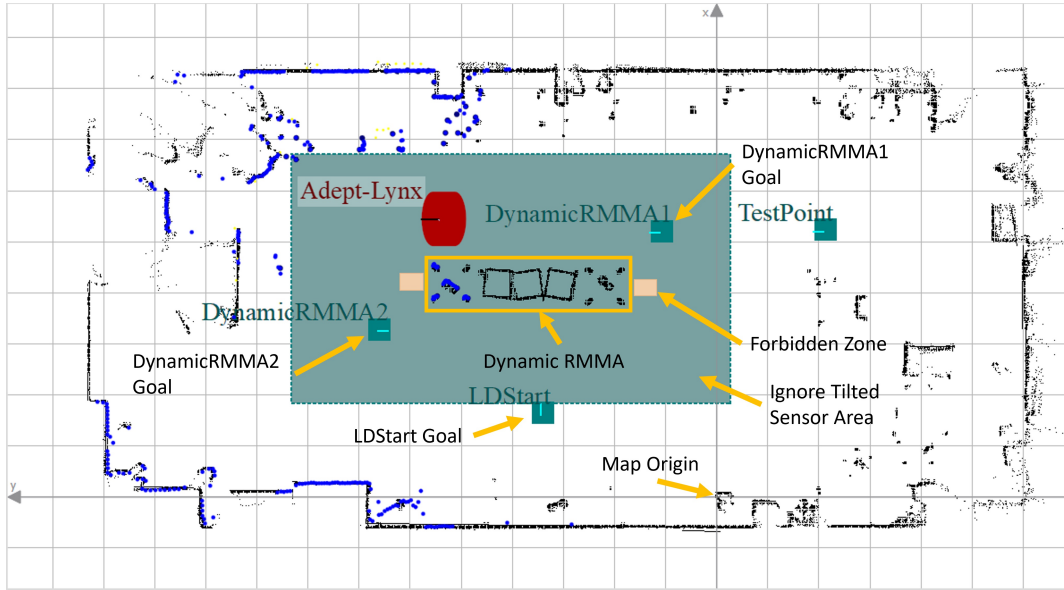


Fig. 4. Annotated AMR-CT map of the lab space, which was configured with goal points, forbidden zones (to avoid collision with the CMMA), and an area to ignore tilted sensors. Note that the map coordinate system origin is also visible and annotated.

the mobile manipulator system used at Marquette University was the mobility hardware. The AMR system utilized in Ref. [10] was a research platform with greater software re-configurability, existing ROS support with available Universal Robot Descriptor Format (URDF) models, and a software development kit (SDK) supplied by the manufacturer [59–61]. As a result, robot map generation, localization, navigation, and planning was achieved using existing ROS packages including the ROS navigation stack, the ROS Advanced Robotics Interface for Applications (ROSARIA), and Adaptive Monte Carlo Localization (AMCL) [10]. The ROS node graph for this system was reproduced from Ref. [10] in Fig. 5. However, the AMR-CT utilized at NIST was a commercially-oriented platform with all such functions already implemented on the on-board AMR-CT controller. Instead, functionality on the AMR-CT was primarily exposed through a clear-text Application Programming Interface (API) [34, 62]. As a result, the aforementioned packages related to AMR functionality for the Marquette AMR mobile manipulator were no longer needed and the corresponding ROS nodes (i.e., /amcl, /map_server, /move_base, and /RosAria) were replaced with two new nodes, called /ld_arcl_cmd and /ld_arcl_outgoing, respectively. The two new nodes were created to interface with the AMR-CT via the clear-text API (see Fig. 3, bottom) and will be further described in the following paragraphs. The result of these changes are shown in Fig. 6.

Custom C++ code to interface with the AMR-CT via its clear-text API was written as part of previous experiments [33, 34, 62]. The same code was re-used as a starting point to write the new /ld_arcl_cmd and /ld_arcl_outgoing ROS nodes. However, since the code was

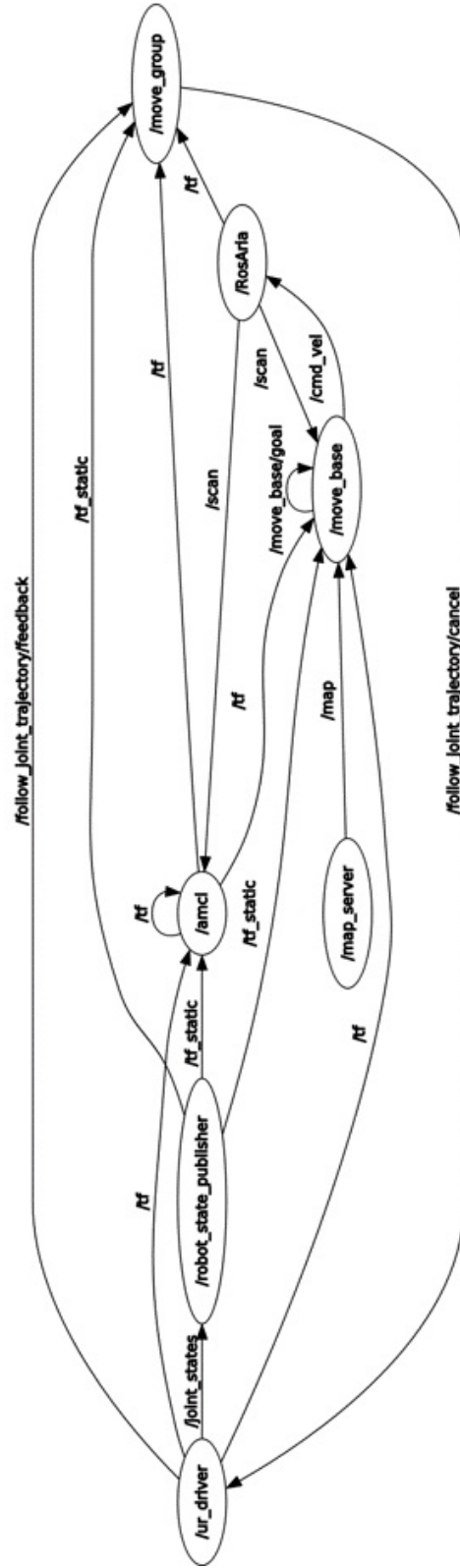


Fig. 5. ROS graph showing node and topics for ROS packages used with the Marquette University mobile manipulator platform reproduced from Ref. [10].

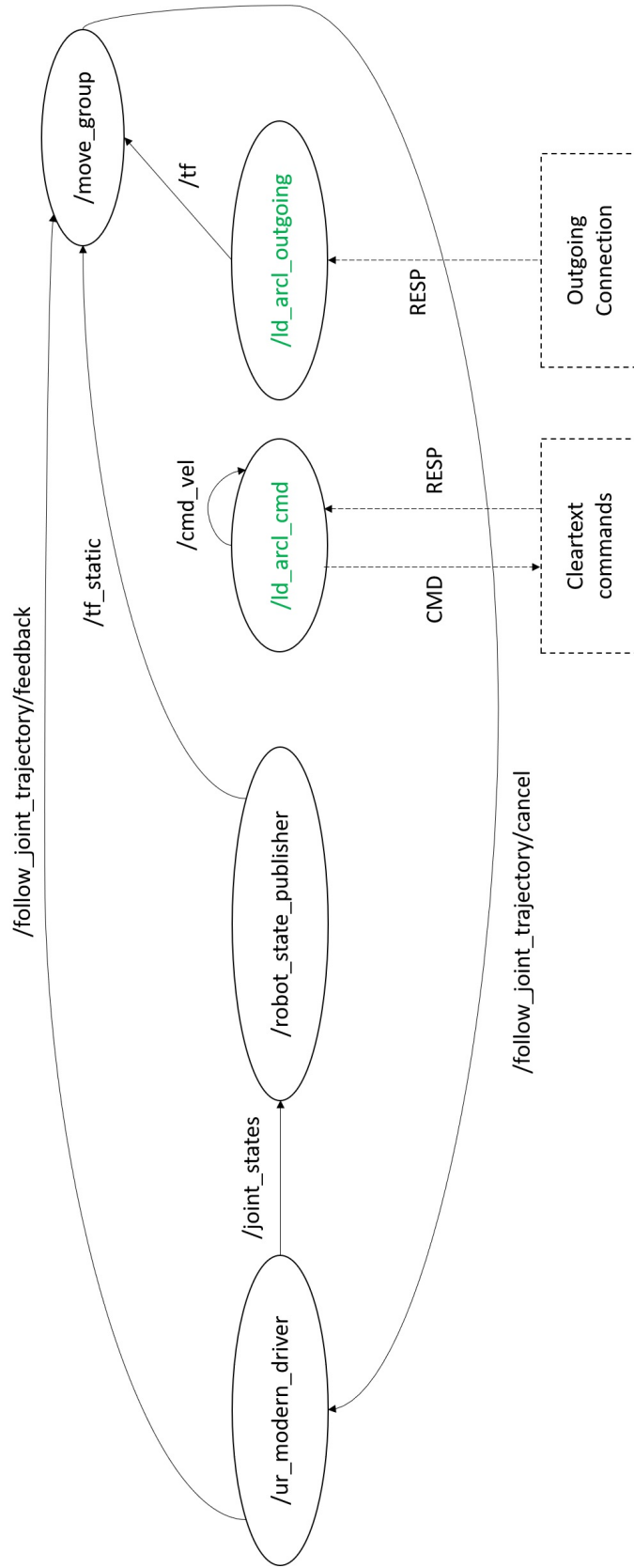


Fig. 6. Annotated ROS graph showing the nodes and topics that were removed in the process of integrating the code with the NIST AMR-CT mobile manipulator platform.

originally compiled for a Windows-based laptop, it was necessary to refactor and recompile the source code first for use with Linux and then as a catkin package for use with ROS. Additionally, in the prior work, the User-Level Application Programming Interface (ULAPI) provided a cross-platform implementation for Transmission Control Protocol / Internet Protocol (TCP/IP) sockets [63]. This assisted in recompiling the code for Linux with minimal changes, but ULAPI itself also had to be recompiled for Linux systems. The instructions packaged with the ULAPI software (see Ref. [64]) were followed to accomplish this, however, one deviation from these instructions was required to successfully install the library: namely, invocation of the autoconf utility to generate the configure script run as part of step 1 in Ref. [64] (see also Ref. [65]).

There were four key pieces of functionality utilized with the Marquette University AMR that were translated to the functionality exposed by the clear-text API of the AMR-CT. This functionality included translational and rotational velocity control, navigation to goals, re-enabling the motors after an emergency stop (e-stop), and reporting the current pose. The replication of the translational and rotational velocity control will be addressed first.

In the original ROS code developed at Marquette University, the `/move_base` node would publish to a topic called `/cmd_vel` that was also subscribed to by the `/RosAria` node, the latter of which was used to control the AMR, as shown in Fig. 5 [10]. The `/cmd_vel` topic accepted a ROS Twist message that was composed of two 3D vectors. The first 3D vector stored translational velocities along each of three orthogonal axes, in m/s, while the second 3D vector stored angular velocities for roll-pitch-yaw (RPY), in rad/s [66]. However, it should be noted that for the Two-Dimensional (2D) coordinate system of the AMR-CT, only the forward translational velocity axis and the yaw angular velocity were relevant. Therefore, in general, movement functionality was replicated by creating a new ROS topic subscriber in the `/ld_arc1_cmd` node, which was set to operate at 10 Hz and in which receipt of a Twist message as input would trigger a callback function that then forwarded the corresponding clear-text command to the AMR-CT for execution [67, 68]. Note that a ROS subscriber was chosen to re-create this functionality rather than a ROS service (discussed further in the next paragraph) because preemption of velocity-based control was required to either stop, adjust heading, or alter the translational speed of the AMR-CT during CMMA registration. The subscriber callback would prioritize execution of translational movement (i.e., if a valid forward translational velocity was received) before rotational movement. A stop command was forwarded if both yaw angular and forward translational velocities of zero were received [62].

Translational movement was implemented using a dead-reckoning move command⁹, which accepted a number of parameters for modifying the AMR-CT behavior during task execution [62]. The parameters changed from the default values included the total move distance,

⁹As an alternative, utilization of the `goToStraight` command, which commanded the AMR-CT to navigate between two goals in a straight line, was also considered and tested. Velocity control of the command was implemented by modifying the AMR-CT configuration through the clear-text language, but with the cart attached it was observed that the command execution could result in un-localization.

set to 4500 mm (i.e., long enough to ensure the AMR-CT could travel the entire length of the CMMA without reissuing the command), the front clearance, set to 25 mm, and the speed, which could range between and including 10 mm/s and 500 mm/s, depending on the received Twist message. In addition, two other key parameters of the move command required modification to allow for proper AMR-CT operation. The first was a “time-to-fail” value, which would cause task execution to stop if not completed within the specified time limit and was extended from the default of 30 s to 3600 s. The second parameter was a Boolean “fail-if-stopped” flag that would cause task execution to stop if an obstacle or collision was detected. It was found that enabling this value, which was the default, would sometimes result in sudden manipulator motion being mistakenly perceived by the AMR-CT as a collision. Disabling this feature prevented such occurrences.

Since dead-reckoning translational movement is susceptible to the propagation of both systematic and random errors, a gradual deviation of the AMR-CT heading from the original heading of the commanded goal was commonly observed as the AMR-CT traveled along a given side of the CMMA. However previous experiments also suggested that the AMR-CT controller could be used to accurately monitor this error [34, 69]. Therefore, rotational movement was required to periodically correct the error in the AMR-CT heading. This heading correction was implemented both as an initial correction after the AMR-CT traveled 250 mm towards the first fiducial target on a given CMMA side and each time after a fiducial interception with the RLS, provided that the heading error was greater than 0.01 rad. It should also be noted that it did not appear to be possible to simultaneously execute rotational and translational movements, so translational movement was halted during heading correction. While the AMR-CT clear-text API exposed a `setHeading` command, it was found that the heading could only be specified at a minimum of 0.017453 rad increments, which exceeded the 0.01 rad target tolerance [62]. Therefore, the `deltaHeading` command, which had parameters that could be used to specify the rotation velocity and a maximum rotation range was used instead. This command was used in conjunction with the `stop` command, so that the AMR-CT could continue rotating until preempted [62]. The maximum range parameter was set to ± 6.2832 rad with the sign indicating the direction of rotation and was varied depending on the sign of the heading error. All other input parameters for the `deltaHeading` command were kept to their default values.

In the `/ld_arcl_cmd` node, the tasks of goal navigation and re-enabling the motors after an e-stop were re-implemented as ROS service [70]. ROS services were selected to re-create these functions as opposed to topic Subscribers since, unlike velocity control of the AMR-CT, no preemption of the goal navigation or motor enable commands was needed and ROS services block until the associated callback function has completed execution and generated a response [71]. Re-creating the functionality of re-enabling the motors was not strictly necessary, but included to facilitate compatibility with the existing code. The ROS service used to re-enable the motors accepted and returned empty request and response messages, and the functionality was replicated by issuing an `enable motor` clear-text command to the AMR-CT [62, 71]. To re-create goal navigation functionality, Ref. [71, 72] were con-

sulted while writing the code to allow the `/ld_arcl_cmd` to accept a new `/go_to_goal` request message as input and issue a `goToPoint` clear-text command to the AMR-CT [62]. The `/go_to_goal` message consisted of a three-vector that stored positional Cartesian coordinates in the AMR-CT map coordinate system (in m), in addition to a heading (in radians). The `/go_to_goal` service was programmed to convert the coordinates to mm and the heading to degrees before passing the values directly to the `goToPoint` command as parameters.

The functionality of reporting the current position and orientation (pose) of the AMR-CT was implemented in a separate node called `/ld_arcl_outgoing`, which was named after the “Outgoing Connection” communication mode exposed by the AMR-CT controller [62]. As documented in previous experiments, this mode of communication followed a client-server model in which the AMR-CT acted as the client and would repeatedly execute one or more user-selected commands at a pre-specified interval [34, 62]. The result of the commands would then be forwarded to a custom server application, which, in this case, was the `/ld_arcl_outgoing` node. When two or more commands are executed in this fashion, they are executed by the AMR-CT controller within the same computation cycle, which made use of this method preferred to ensure synchronization of data since the clear-text language required separate commands to individually query each component of the pose, in addition to a timestamp [34, 62]. As just mentioned, while a timestamp could be retrieved from the AMR-CT controller along with the pose, a timestamp retrieved from ROS on the server side was ultimately used. The rate at which the clear-text commands were executed was set to the fastest supported setting in the AMR-CT configuration, which was 100 Hz. The `/ld_arcl_outgoing` node, then functioned as a ROS publisher (more specifically, a transform broadcaster) that received the queried pose, which consisted of a three-vector storing two Cartesian position components and one heading component, and then published the pose as a 6DoF “StampedTransform” object (see Ref. [73]) between the coordinate frames “map” and “base_link” [74]. The `/ld_arcl_outgoing` control loop rate was also set to 100 Hz to match the command execution rate of AMR-CT clear-text commands¹⁰.

2.3. Manipulator-on-a-Cart

Before discussing software, a few minor changes to the manipulator-on-a-cart hardware should be noted. First, the linear actuators (shown in Fig. 2) were not utilized for this work since the cart was intended to remain attached to the AMR-CT during testing. Second, the two digital levels, which could output angles between $\pm 180^\circ$, inclusive, and were specified to have an accuracy of $\pm 0.05^\circ$ when the level was between 0° and 10° and a repeatability of $\pm 0.05^\circ$, were used only to assist in the AMR-CT map to Optical Tracking System

¹⁰In practice, the publishing rate of the transformation was limited to 10 Hz despite the higher control loop rate of the `/ld_arcl_outgoing` node and command execution rate that could be specified in the AMR-CT configuration. This is because 10 Hz is the cycle rate of the application-level software running on the AMR-CT controller.

(OTS) coordinate transformation calibration presented in Sec. 3.1, rather than during the primary evaluation presented in Sec. 4 [75]. Therefore, the levels were not connected to the additional PC, and the latter was not used at all. Finally, to mitigate the reduced capacity of the degraded batteries that powered the manipulator-on-a-cart, the two 12 V, 28 Ah batteries that powered the 900 W, 24 V DC to AC power inverter were replaced. Additionally, a 2C40DL circuit breaker, with 250 V DC voltage rating and 10 kA short-circuit rating was introduced to completely disconnect the batteries from the inverter when the manipulator-on-a-cart was not in use [76–78]. Otherwise, the same hardware specifications and manipulator, which had a maximum reach and repeatability of 850 mm and 0.1 mm, respectively, was used [34, 79]. The same RLS from prior work was used to digitally detect the retro-reflective mock-assembly targets mounted to the CMMA. Detection of the retro-reflective targets served as an analog to peg-in-hole assembly performance (see also Sec. 2.4) [15, 29, 80].

The manipulator controller firmware was updated from version 3.1.18213 to version 3.7.2.40245, which was downloaded from the manufacturer website and installed following the procedure specified by the manufacturer [81, 82]. The controller firmware upgrade was necessary to use the Real-Time Data Exchange (RTDE) interface [83], which was used to monitor and publish the status of the RLS digital inputs to a ROS topic.

The motion planning framework at Refs. [84, 85] was used for high-level manipulator EOAT trajectory planning and the Marquette mobile manipulator code was updated to use the software driver at Ref. [86] to interface with the manipulator controller. The motion planning framework and software driver was originally integrated with the `ros_control` interface [87]. However, two differences between the NIST AMR-CT mobile manipulator and Marquette platform warranted the disabling of the `ros_control` interface. First, the NIST AMR-CT mobile manipulator relied on wireless communications as opposed to the wired communications of the Marquette AMR mobile manipulator. Additionally, the use of the Linux compatibility layer restricted the use of a low-latency OS kernel, which could impact performance when using the `ros_control` interface [88, 89]. Due to these differences, execution of the manipulator trajectories was found to be unstable and resulted in frequent protective stops. Therefore, the code was further modified to use a “low bandwidth” mode, which shifted the real-time constraints from the client to the manipulator controller and was intended to address performance degradation over wireless connections and non-low-latency OS kernels [89]. Use of this mode was mutually exclusive with the `ros_control` interface [89] and required fixing the bug at Ref. [88] as documented.

2.4. Type B CMMA

Between the two variants of the CMMA (i.e., the Type A and Type B CMMA as outlined in Sec. 1), the Type B CMMA was used for this project since 1) it provided a more direct representation of large-scale part manufacturing [15] and 2) the same CMMA design was utilized in Refs. [9, 10]. The Type B CMMA utilized for this project is depicted in Fig. 7

and consisted of a machined, anodized aluminum build. Unlike the Type A CMMA, the Type B CMMA top-hole plate was constructed from multiple semi-circles. Furthermore, each semi-circle top-hole plate had a general fabrication tolerance of ± 0.254 mm, which was the same as the Type A CMMA [90].

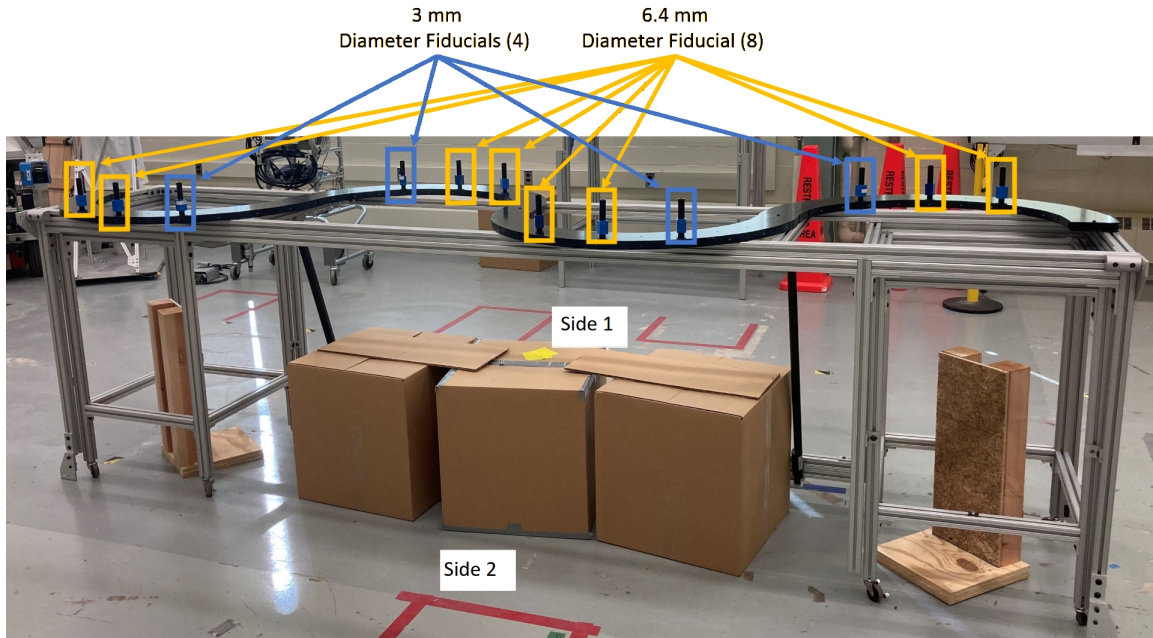


Fig. 7. The Type B Configurable Mobile Manipulator Apparatus (CMMA) configured with 12 retro-reflective fiducials (eight 6.4 mm diameter fiducials and four 3 mm diameter fiducials) and additional objects placed underneath to improve AMR-CT localization.

Unlike the Type B CMMA used at Marquette University, which had only three semi-circle top-hole plates, the NIST Type B CMMA consisted of four semi-circle top-hole plates (i.e., such that two of the four semi-circles were reachable by the manipulator on a given side) [9, 10]. The Type B CMMA at NIST, like the Type A CMMA, had an extruded aluminum support structure that permitted an adjustable height and top-hole plate rotation [15]. This was also unlike the Type B CMMA used at Marquette University, where the top-hole plate was simply laid flat on the floor [9, 10]. The NIST Type B CMMA was set to the minimum height of approximately 1800 mm off of the ground (as measured by tape measure) and configured with no top-hole plate rotation. Again, as mentioned previously in Sec. 2.2, additional wood and cardboard objects were placed under the CMMA to improve the localization performance of the AMR-CT.

The NIST Type B CMMA was outfitted with 12 mounted retro-reflective targets of 6.4 mm (eight of 12 targets) and 3 mm diameter sizing (four of 12 targets), as shown in Fig. 7. Detection of the fiducials by the manipulator using the RLS (again depicted in Fig. 1) was analogous to the task of typical peg-in-hole assembly [15, 29].

2.5. Optical Tracking System

The same OTS from prior experiments was utilized again for this work, with the exception of the manufacturer-supplied software, which was upgraded to the then latest version (3.0.0 beta 1) and ran on a new computer with the following specifications: An eight-core processor with a clock-rate of 3.70 GHz, 64 GB of RAM, a graphics card using driver version of 27.21.14.5671, and running Windows 10 build 19042 [91, 92]. The OTS consisted of 20, 4 MP resolution cameras that were placed to maximize the field-of-view (FOV) within the 9000 mm x 22000 mm x 7000 mm (width x length x height) lab space and had a data capture rate of 120 Frames-per-Second (FPS). Of the 20 cameras, eight were focused directly on the CMMA and included one camera mounted on the ceiling directly overhead. The Type B CMMA was placed in the same approximate location in the lab space as the Type A CMMA of past work [33, 34].

The OTS was re-calibrated prior to each use according to the manufacturer-specified procedure. This procedure generally involved removing or covering as many stray reflections in the lab as possible, masking the reflections that could not be removed or covered, generating and recording sample data from a calibrated, manufacturer-supplied wand artifact, and setting the OTS coordinate system origin and ground plane via placement of a calibrated, manufacturer-supplied square artifact [93]. Note that the procedure for setting the ground plane included making sure the square artifact was leveling using the two, built-in bubble levels. Additionally, the wand used in past experiments was replaced with a new, identical model prior to this work [94].

As in prior experiments, the rigid body tracking was used as a ground truth measurement reference for the coordinate system calibrations presented in Sec. 3 and the factorial experiment conducted in Sec. 4 since prior experiments had established the static (e.g., when the rigid body is not in motion) repeatability of the OTS positional rigid body tracking to be 0.022 mm and the dynamic (e.g., when the rigid body is in motion) positional tracking repeatability to be 0.26 mm [14, 95]. To facilitate rigid body tracking, 19 mm diameter retro-reflective markers were placed on the tracked objects and were selected to balance marker visibility with tracking of fine robot movements. Following manufacturer recommendations, the markers placement followed intentionally asymmetric configurations and on the same link of a robot (i.e., not spanning a movable robot joint), which helped the OTS discern rigid body orientation and marker labels [34, 96].

Additionally, when needed for analysis in Sec. 3, rigid body tracking data was exported to Comma Separated Value (CSV) formatted files. The ground plane artifact, mentioned earlier in this section, defines a 3D, right-handed coordinate system with the x and z axes parallel to the plane and the y axis perpendicular to the plane. However, when exporting rigid body tracking data as a CSV file, the axis convention can be re-defined. In general, the “Measurement” coordinate axis convention was used, in which the x and y axes were parallel to ground plane and the z axis perpendicular to the ground plane. This roughly aligned the x and y axes of the rigid body tracking data with the AMR-CT map shown in

Fig. 4 (see Sec. 2.2). Data exports for each frame of tracking data contained a frame number, a timestamp, the Cartesian position (along x , y , and z axes) and quaternion orientation (with components q_x , q_y , q_z , q_w) of each rigid body centroid, and the Cartesian position of each marker within each rigid body.

2.6. Time Synchronization

Time synchronization across the clients depicted in Fig. 3 was important to ensuring data correspondence and alignment across the different experimental logs and ground truth data as described later in Sec. 4.2. The OTS computer and ROS laptop running the mobile manipulator control code were synchronized to the NIST timeservers via Network Time Protocol (NTP) and over the internet (resulting in an unknown number of hops¹¹) [98]. The OTS computer was synchronized via the tool in Ref. [99]. For the ROS laptop, it was important to synchronize both the host Operating System (OS) and the Linux compatibility layer. In the former case, the same tool from Ref. [99] was used and, in the latter case, a script from Ref. [100] was used. Synchronization of the NIST AMR-CT controller with the NIST time servers was attempted via NTP as well, but there was no known way to synchronize the manipulator controller. It should be noted that time synchronization of the AMR-CT controller occurred only upon boot up of the vehicle. Latency verification for the OTS computer and the host OS control laptop was achieved via command line through a command that displays a strip chart [99]. Figure 8 contains an example screenshot demonstrating how the time synchronization of the OTS and, similarly, the control laptop was conducted. For the Linux compatibility layer of the control laptop, time synchronization delay verification was achieved via the NIST time server website as shown in Fig. 9 [98]. The verification involved installation of an internet browser via the command line and displaying the web page at Ref. [98] through the X Window server detailed in Sec. 2.1. Verification of the AMR-CT was attempted by visually inspecting the controller time as reported by the proprietary control software against the time reported by Ref. [98] within 1 s. However, due to the imprecision of this method and that, during the experiment, it was discovered that connection to the AMR-CT over the local network was prevented due to a firewall misconfiguration, time synchronization of the AMR-CT could not be assumed.

3. Coordinate System Transformation and Registration Measurements

One key problem encountered while integrating the existing Marquette University software and registration algorithms was coordinate system transformation inconsistencies. The previous implementation had used the OTS as a global coordinate system and ground truth reference for measuring dimensions used to define models of the mobile manipulator, CMMA, and the initial position of the fiducials mounted to the CMMA for localization. This was done so that trajectory planning for the spiral and Unscented Kalman Filter (UKF)-based

¹¹Here, “hops” refers to the number of switches or routers between a communication source and destination [97].

```
PS C:\WINDOWS\system32> w32tm /stripchart /computer:
Tracking
The current time is 2/2/2022 9:09:19 AM.
09:09:19, d:+00.0024259s o:+196.1907619s [
09:09:21, d:+00.0026616s o:+196.1863622s [
09:09:23, d:+00.0028464s o:+196.1818444s [
09:09:25, d:+00.0024140s o:+196.1770247s [
09:09:27, d:+00.0028326s o:+196.1727026s [
09:09:29, d:+00.0032046s o:+196.1682515s [
09:12:47, d:+00.0029895s o:+00.0010093s [
09:12:49, d:+00.0020690s o:+00.0004938s [
09:12:51, d:+00.0029390s o:+00.0009108s [
09:12:53, d:+00.0025565s o:+00.0006448s [
09:12:55, d:+00.0031046s o:+00.0010580s [
09:12:57, d:+00.0056856s o:+00.0022918s [
09:12:59, d:+00.0024756s o:+00.0008082s [
09:13:01, d:+00.0027023s o:+00.0008781s [
09:13:03, d:+00.0024687s o:+00.0008438s [
09:13:05, d:+00.0032588s o:+00.0011759s [
09:13:07, d:+00.0025750s o:+00.0008903s [
09:13:09, d:+00.0027888s o:+00.0004862s [
09:13:11, d:+00.0031953s o:+00.0012076s [
PS C:\WINDOWS\system32>
```

Fig. 8. Example screenshot showing how the OTS computer time synchronization was verified via command line.

search methods could be done globally, which simplified the process of adding velocity compensation to the planned trajectories. The globally-planned trajectories were then transformed to the local coordinate system of the manipulator base for execution [9, 10].

The physical and software differences between the NIST AMR-CT mobile manipulator system and the mobile manipulator system used at Marquette University described in Sec. 2.3 required the calibration of replacement coordinate system transformations to suit the newly integrated hardware and software. The remainder of this section describes the calibration procedures and results.

3.1. Cart Transporter Map to OTS Registration

3.1.1. Registration Problem Summary

As described in Sec. 2.2, the AMR used at Marquette University leveraged ROS packages for implementing robot map management, localization, navigation, and planning. With these packages it was possible to arbitrarily place the origin of the AMR map coordinate system such that it matched the placement of OTS coordinate system. However, with the AMR-CT utilized at NIST there was no known way to precisely specify the map coordinate system origin upon creating a new laser scan of the lab space or re-position the map coordinate system origin after map creation. Therefore, the first calibration problem was determining the transformation between the AMR-CT map coordinate system origin and the OTS coordinate system. In prior experiments with the AMR-CT, similar calibrations assumed a relatively flat, level floor surface in the lab space such that the problems and solutions could be simplified to use 2D projections. Since prior experiments suggested that the floor leveling may not have been consistently flat across the lab, a full 6DoF calibration

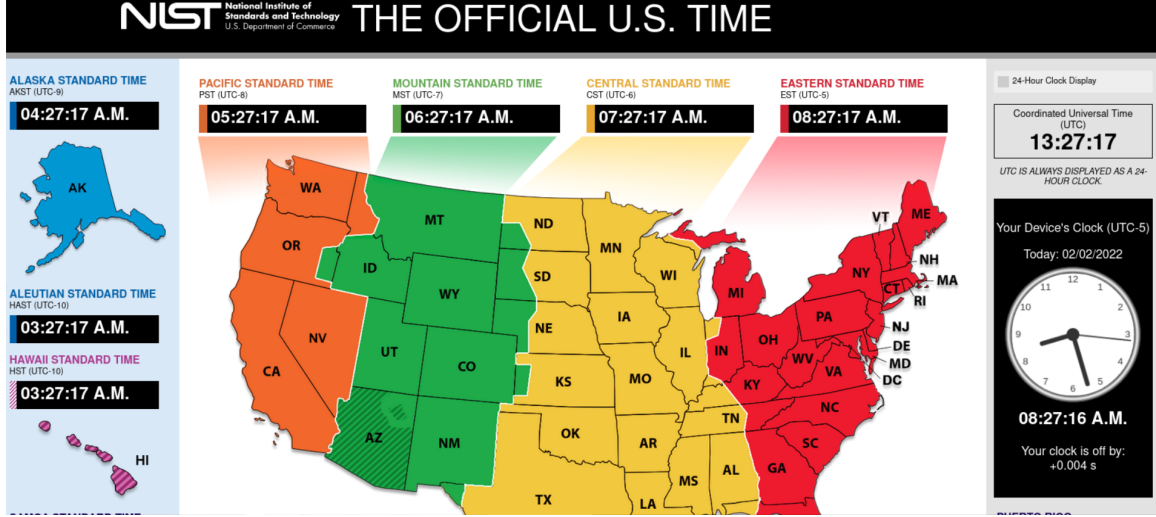


Fig. 9. Example screenshot showing how the time synchronization of the control laptop Linux subsystem was verified.

was adopted for this work, representing an improvement over past experiments [34].

Figure 10 describes the coordinate systems involved in the calibration problem. First, the variable, $vbase$, is used to denote the local coordinate frame placed at the base centroid of the AMR-CT used at NIST. This coordinate system was observed relative to three other coordinate frames that included the OTS (denoted OTS), the origin of the 2D map coordinate system used by the AMR-CT controller for navigation (denoted MAP_2D), and the origin corresponding to a 3D projection of the 2D map coordinate system used by the AMR-CT controller (denoted MAP_3D). Therefore, the 6DoF homogeneous transformation matrix, $\{^{MAP_2D}H_{i\{vbase\}}\}$, represents the i -th (out of n total) observed pose of the AMR-CT base relative to the MAP_2D coordinate system. Similarly, the matrices $\{^{MAP_3D}H_{i\{vbase\}}\}$ and $\{^{OTS}H_{i\{vbase\}}\}$ represent i -th (out of n total) observed pose of the AMR-CT base relative to the MAP_3D and OTS coordinate systems, respectively. Additionally, let

$$\begin{aligned} \{^{MAP_2D}H_{\{vbase\}}\} &= [\{^{MAP_3D}H_{1\{vbase\}}\}, \{^{MAP_3D}H_{2\{vbase\}}\}, \dots, \{^{MAP_3D}H_{n\{vbase\}}\}], \\ \{^{MAP_3D}H_{\{vbase\}}\} &= [\{^{MAP_3D}H_{1\{vbase\}}\}, \{^{MAP_3D}H_{2\{vbase\}}\}, \dots, \{^{MAP_3D}H_{n\{vbase\}}\}], \text{ and} \\ \{^{OTS}H_{\{vbase\}}\} &= [\{^{OTS}H_{1\{vbase\}}\}, \{^{OTS}H_{2\{vbase\}}\}, \dots, \{^{OTS}H_{n\{vbase\}}\}] \text{ (i.e., matrices contain-} \\ &\text{ing all } n \text{ observations in each coordinate system).} \end{aligned}$$

The aforementioned homogeneous transformation matrices constituted the known coordinate transformations of the calibration problem and were constructed from experimental data obtained from the AMR-CT controller, two digital levels, and the rigid body tracking of the OTS (more details on the data capture and construction of these matrices in Sec. 3.1.2).

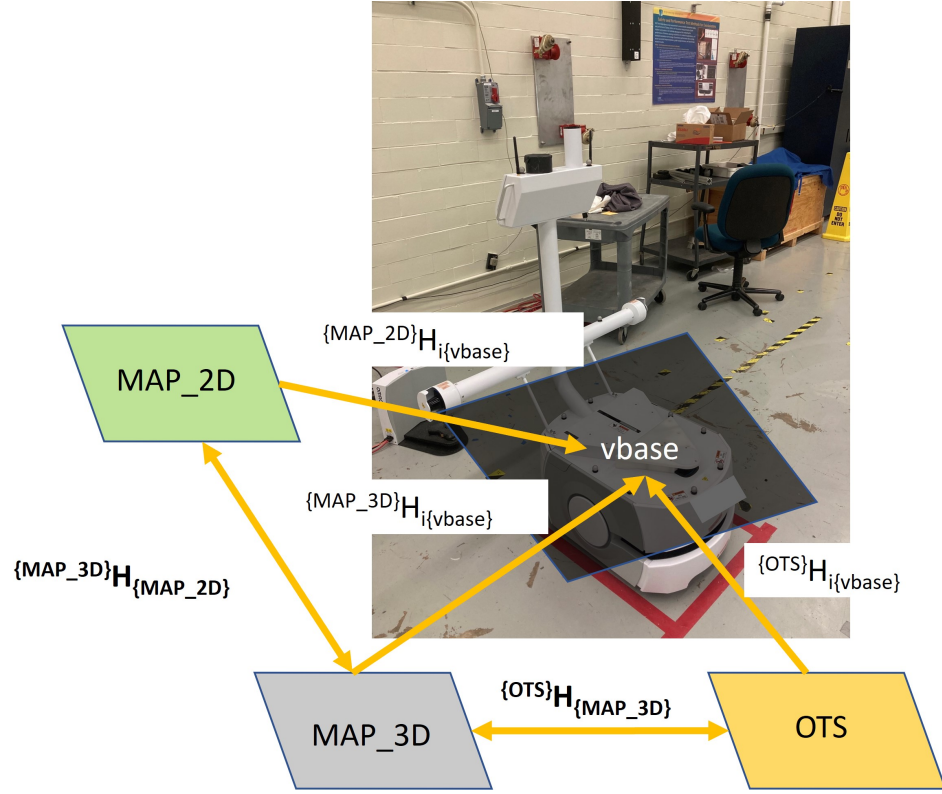


Fig. 10. Diagram of homogeneous transformations in different coordinate systems needed to calibrate the unknown coordinate transformation between the AMR-CT map and the OTS (left). Simplified graph showing displaying the same information (right). Note that the double arrows imply the inverse of the respective labeled transformation, though not shown, can also be applied.

The problem of measuring the transformation between the AMR-CT map coordinate system origin and the OTS coordinate system amounted to using the known homogeneous transformation matrices just described to solve for two unknown homogeneous transformation matrices: the transformation of the *MAP_2D* coordinate system relative to the *MAP_3D* coordinate system (denoted $\{MAP_3D\}H_{\{MAP_2D\}}$) and the transformation of the *MAP_3D* coordinate system relative to the *OTS* coordinate system $\{MAP_3D\}H_{\{OTS\}}$. Solving for the two unknown homogeneous transformation matrices was accomplished by solving the two least squares optimization sub-problems presented in Eq. 1 and Eq. 2, which can be solved via the closed-form solutions presented in existing work [101–103]. Note that, in Eq. 1 and 2, $\|\cdot\|$ denotes the Frobenius norm [101].

$$\min_{\{MAP_3D\}H_{\{MAP_2D\}}} \|\{MAP_3D\}H_{\{MAP_2D\}}\{MAP_2D\}H_{\{vbase\}} - \{MAP_3D\}H_{\{vbase\}}\|^2$$

such that: $\{MAP_3D\}H_{\{MAP_2D\}}\{MAP_2D\}H_{\{vbase\}} \approx \{MAP_3D\}H_{\{vbase\}}$ (1)

$$\min_{\{OTS\} \mathbf{H}_{\{MAP_3D\}}} ||\{OTS\} \mathbf{H}_{\{MAP_3D\}}^{\{MAP_3D\}} H_{\{vbase\}} - \{OTS\} H_{\{vbase\}} ||^2$$

such that: $\{OTS\} \mathbf{H}_{\{MAP_3D\}}^{\{MAP_3D\}} H_{i\{vbase\}} \approx \{OTS\} H_{i\{vbase\}}$ (2)

3.1.2. Materials and Data Collection

Prior to taking the measurements, which was done immediately following the creation of the AMR-CT map shown in Fig. 4 and described in Sec. 2.2, two digital levels were mounted to the base of the AMR-CT using adhesive strips as depicted in Fig. 11 (left). Each level had an output range of $\pm 180^\circ$, inclusive, an accuracy of $\pm 0.05^\circ$ when readings were between 0° and 10° , and a repeatability of $\pm 0.05^\circ$ [75]. Additionally, the calibration of each level was checked and re-calibrated as needed before each use and following the manufacturer-specified procedure [75]. The digital levels were used to identify seven distinct stops at which the AMR-CT was level (i.e., where both levels had a reading of $\pm 0.25^\circ$ or less) by manually driving the AMR-CT via joystick throughout the lab space and observing the digital level readings. Along with an assigned integer label (1-7), the approximate location and orientation of the stops was then visually marked on the floor using tape and permanent marker, as shown in Fig. 11 (right). The AMR-CT pose, as measured by the OTS, is shown in Fig. 12. The integer labels and a formula implemented in spreadsheet software were used to generate three random run-sequences among the stops, such that the measurements could be repeated three times per stop [104].

The OTS was re-calibrated according to the manufacturer-specified procedure, (see Sec. 2.5). For the calibration, the AMR-CT, cart base, and CMMA among other objects were covered using tarps, however, it should be noted that during calibration, floor reflections may not have been covered and may have been masked via software instead. Additionally, the CMMA remained covered during OTS calibration. The light at the top of the AMR-CT touchscreen console also remained covered using a roll of electrical tape (as seen in Fig. 11 (right)) throughout data capture to prevent the light and reflective material from causing OTS marker occlusions. The calibration summary generated by the manufacturer-supplied software reported a mean ray error of 0.621 mm and mean wand error of 0.113 mm, which had the best quality classification of “exceptional” (as determined by the manufacturer). To track the base of the AMR-CT, a rigid body was constructed by placing six 19 mm diameter retro-reflective markers to the AMR-CT, as shown in Fig. 13. The rigid body was named “VEHICLE_BASE”.

The AMR-CT localization was verified prior to measurement by sending the AMR-CT to the “LDStart” goal (see Fig. 4 in Sec. 2.2) and observing the localization score in the manufacturer software. The initial AMR-CT localization score was 98.1%.

The measurement procedure was as follows. For each stop in the randomly generated sequence, the operator would move the AMR-CT to the stop position marked by the tape

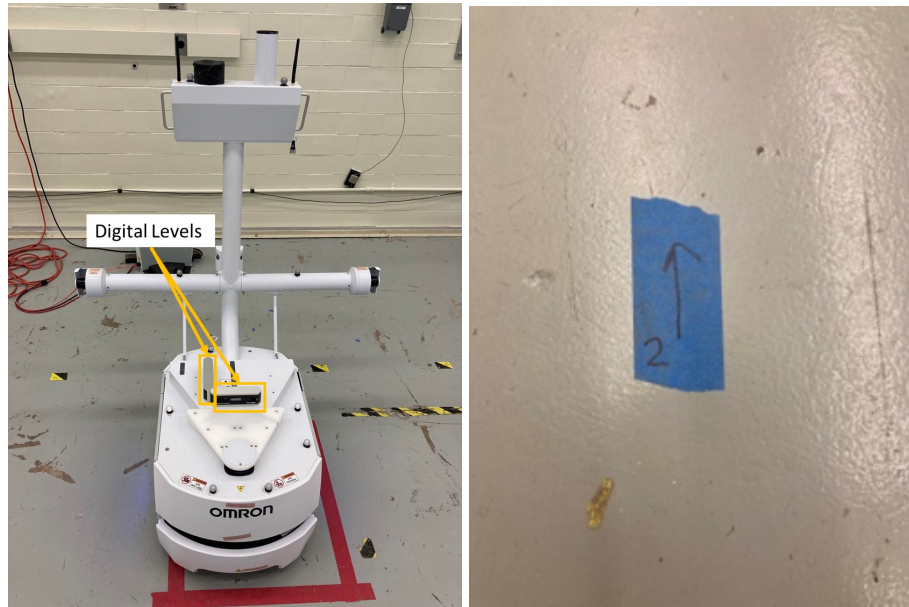


Fig. 11. Digital levels mounted to the base of the AMR-CT using adhesive strips (left). Example stop label indicating the stop number, as well as the approximate location and orientation of the AMR-CT at a level location in the lab (right).

on the floor using the joystick. The operator would then monitor the digital levels and adjust the AMR-CT pose using the joystick until both digital levels had readings of less than $\pm 0.02^\circ$. Note that, due to this, the AMR-CT pose at each stop could vary between repetitions, but having a variety of poses was determined to be beneficial for the measurement procedure. The signed values of the digital levels, the AMR-CT pose (including x , y , θ components as read from the manufacturing monitoring software) in the map coordinate system, and the AMR-CT localization score, which is a measure of confidence based on the percentage of laser readings the AMR-CT receives that matches the scans in its current map, were then manually recorded in a spreadsheet [58, 105]. The operator would then record the OTS tracking data of the static “VEHICLE_BASE” rigid body for a 10 second interval. This was repeated for each of the seven stops in the random sequence and for three repetitions per stop (a total of $n = 21$ observations). After all repetitions were completed, the digital level readings and AMR-CT pose recorded in the spreadsheet were transferred and saved in a single CSV file. The OTS rigid body tracking data was also exported to a CSV file as described in Sec. 2.5.

3.1.3. Data Pre-processing

Once all of the data was collected, a Python script was written to pre-process the generated CSV files. Version 4.11.0 of Anaconda, a publicly available Python distribution, which in turn used Python language version 3.8.8, was used for the data pre-processing [106]. Essential software packages used for performing the data pre-processing included Pandas

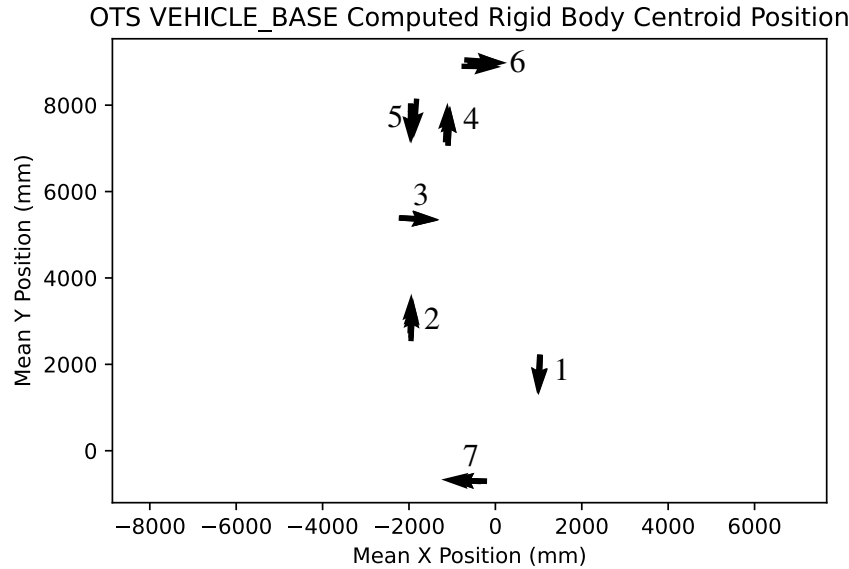


Fig. 12. Quiver plot showing the location and heading of the AMR-CT measured using the OTS at all seven stops selected for measurement in the lab. The plot has been annotated to show the assigned stop numbers.

(version 1.2.4) for data manipulation, Scipy (version 1.6.2) for 3D rotation representation conversions, Matplotlib (version 3.4.4) for data visualization, and NumPy version 1.20.1 for common matrix and vector operations. References [107–114] were consulted when writing the script.

First, essential components of the pre-processing for the OTS rigid body tracking data are described. Since a total of 21, 10 second data captures of the “VEHICLE_BASE” rigid body were generated and the rigid body was static, the first part of the script read and cleaned each CSV file. Since the data was stored in a relation-like data structure, part of the data cleaning involved adjusting the column names so that each component of the rigid body pose and each individual marker positions could be easily accessed by label. To check the quality of the data for each tracked the OTS rigid bodies, the percentage of marker occlusions was also computed and plotted for each capture file and no occlusions were observed [109]. The Cartesian position and quaternion rotation of the “VEHICLE_BASE” rigid body centroid and the Cartesian positions of each marker were aggregated across all frames for each capture by computing the component-wise average, and the markers for the first file were plotted to verify the marker labels (see Fig. 13, left). The component-wise average of the rigid body quaternion rotation was then normalized to approximate the average rigid body orientation. Additionally, the component-wise standard deviation for each marker across all data captures was computed and displayed in box plots, as shown in Fig. 14.

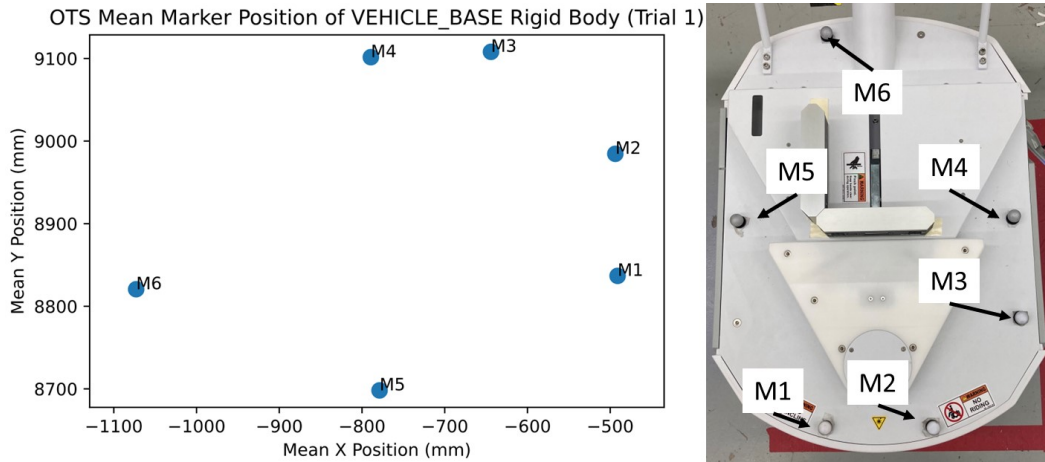


Fig. 13. Marker plot of the VEHICLE_BASE rigid body created for obtaining OTS data needed to measure the unknown coordinate transformation between the AMR-CT map and the OTS (left). Corresponding picture of the physical marker placement on the AMR-CT (right).

Figure 14 was used to verify the repeatability of the captured static OTS data against previously obtained measurements and the established measurement repeatability of the OTS. From the plot, it was observed that the component-wise standard deviations for all markers were lower than the 0.26 mm dynamic rigid body tracking repeatability. Additionally, the central tendency of the component-wise standard deviation for most markers generally hovered around the 0.022 mm static measurement repeatability. However, the central tendency of marker M6, especially in the z component, was closer to 0.03 mm. In general, the z -component of each marker appeared to have the most variability, with the maximum component-wise standard deviation of 0.055 mm observed for markers M5 and M6. One to two outlier values were detected for the x , y , and z components of marker M1, the x and y components of marker M2, the y and z components of marker M3, and the z component of markers M4 and M6 [14, 95]. In general, the plot indicated consistency with previously established measurement uncertainties and that no major issues, such as occlusion or the mislabeling of the VEHICLE_BASE rigid body markers occurred.

Next, a plot of the individual markers for the VEHICLE_BASE rigid body for one data capture (Fig. 11, (left)) was generated to verify the rigid body marker positions and labels. While the data captures included the Cartesian position and quaternion orientation of the rigid body centroid as defined in the OTS software, the placement of this centroid was not guaranteed to align with the physical centroid of the AMR-CT base because it was determined by the (in this case asymmetrical) placement of the individual markers [96]. Therefore, the Cartesian position (with components along the x , y , and z) axes of the physical AMR-CT base centroid in the OTS coordinate system was estimated as the midpoint of markers M4 and M5 (see Fig. 11) for each data capture. The corresponding rotation of the AMR-CT base in the OTS coordinate system was estimated for each data capture using the normalized, component-wise average of the VEHICLE_BASE rigid body centroid quater-

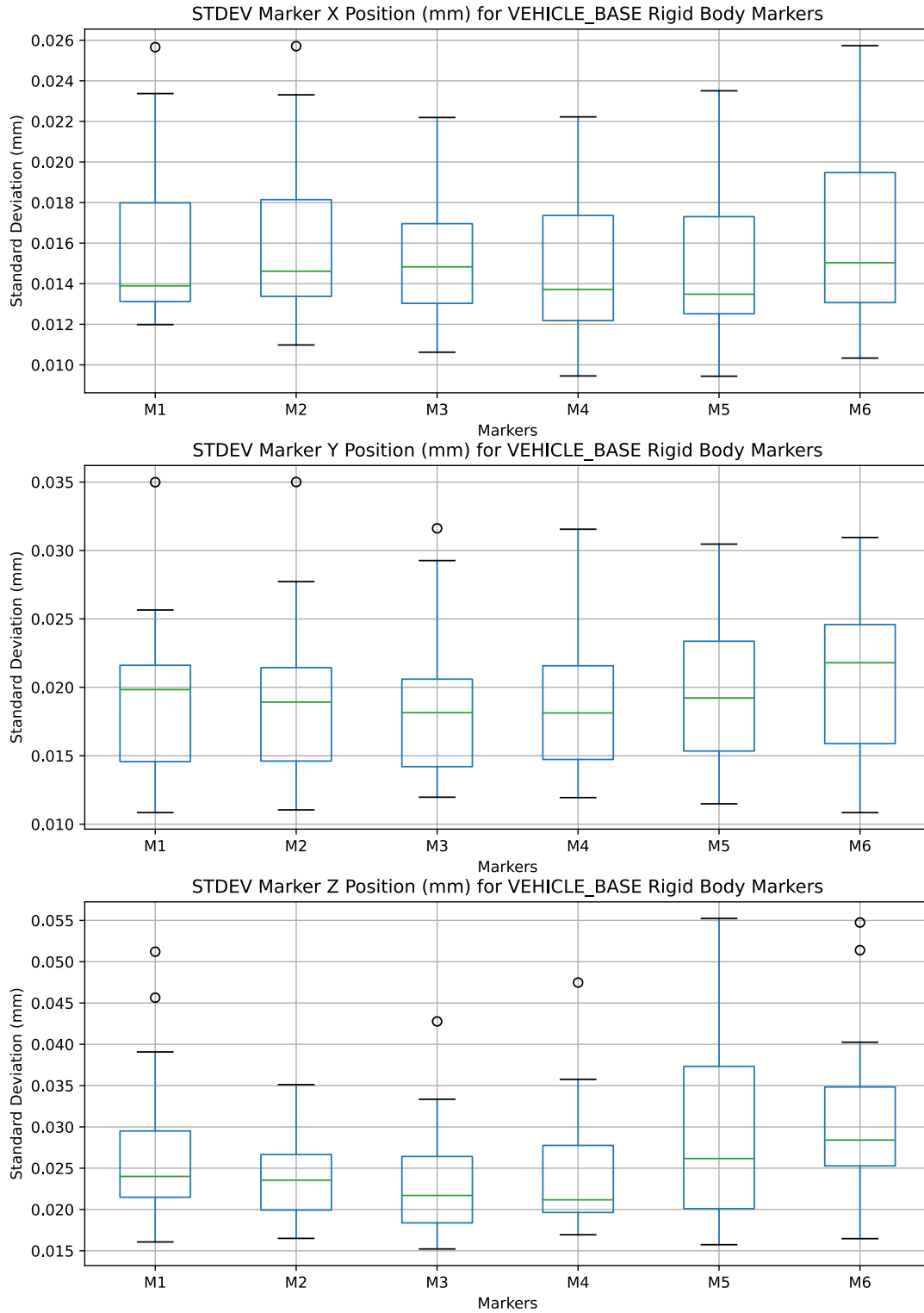


Fig. 14. Boxplots of the component-wise standard deviations for each marker of the VEHICLE_BASE rigid body and for each data capture obtained during the AMR-CT map to OTS coordinate transformation calibration experiment.

nion orientation¹². This information was used by the script to populate each homogeneous transformation matrix, $\{{OTS}\}H_{i\{vbase\}}$, for all $n = 21$ OTS data captures. Additionally, the estimated position and orientation of the AMR-CT base centroid was used to create the plots in Fig. 12 and Fig. 15. In Fig. 15 (top), it was observed that the range in height of the estimated AMR-CT centroid was approximately randomly scattered between 50 mm to 60 mm. The randomness of the scatter was indicative of the randomized run sequence employed and the corresponding standard deviation height was about 20 mm. Furthermore, when comparing the height grouped by stop number, shown in Fig. 15 (bottom), with Fig. 12, it seemed that as the y component of the stop increased, the height decreased (recall that stop 7 was closer to stop 1 than stop 6 in the y component). Since the OTS ground plane was properly leveled during calibration (see Sec. 2.5), scatterplots in Fig. 15, when considered together, suggested the presence of both a constant tilt (or large dip) in the floor and non-constant unevenness in the floor leveling.

The AMR-CT map data pre-processing is now described, which mainly consisted of changing the pose representation such that each $\{MAP_2D\}H_{i\{vbase\}}$ and $\{MAP_3D\}H_{i\{vbase\}}$ could be populated for all $n = 21$ poses recorded from the AMR-CT controller. Since the poses obtained in the *MAP_2D* coordinate frame had 3DoF (i.e., the positional x and y components, as well as the θ heading component), the poses were embedded into the 6DoF $\{MAP_2D\}H_{i\{vbase\}}$ homogeneous transformation matrices by setting the height positional component, as well as the roll and pitch angles of the orientation to zero for each transform. Then, to form the $\{MAP_3D\}H_{i\{vbase\}}$ homogeneous transformations (i.e., the $\{MAP_2D\}H_{i\{vbase\}}$ transformations projected into 3D), OTS data was used instead to initialize the height component and the digital levels readings were substituted for the roll and pitch angles of the orientation. Specifically, for the height component, the first transformation still assumed the height to be zero, but for subsequent transforms the script computed and substitute the difference in height between the first measured pose and the current pose being processed based on the corresponding estimated AMR-CT centroid data obtained from the OTS.

Before proceeding to the calibration results, a few additional plots were generated and examined. A scatter plot of the localization scores reported by the AMR-CT at each stop location and for all 21 data captures are presented in Fig. 16. Note that all of the recorded localization scores were better than 80%, which met the manufacturer recommendation for proper navigation [58]. Additionally, scatter plots showing the recorded digital levels readings are displayed in Fig. 17. The plots show that the level readings had an approximately random scatter within the range of $\pm 0.25^\circ$, although a slight positive trend may be visible in Fig. 17 (top). This observation could be related to operator fatigue that could have occurred towards the end of the experiment, and randomizing the run-order, as was done in the procedure, should have mitigated the impact of any such systematic biases.

¹²The two-vector representation similar to that presented in Sec. 3.2.3 and using markers M2, M4, M5, and M6 of the VEHICLE_BASE rigid body was also tested [115]. However, no noticeable difference was observed in the final calibration results.

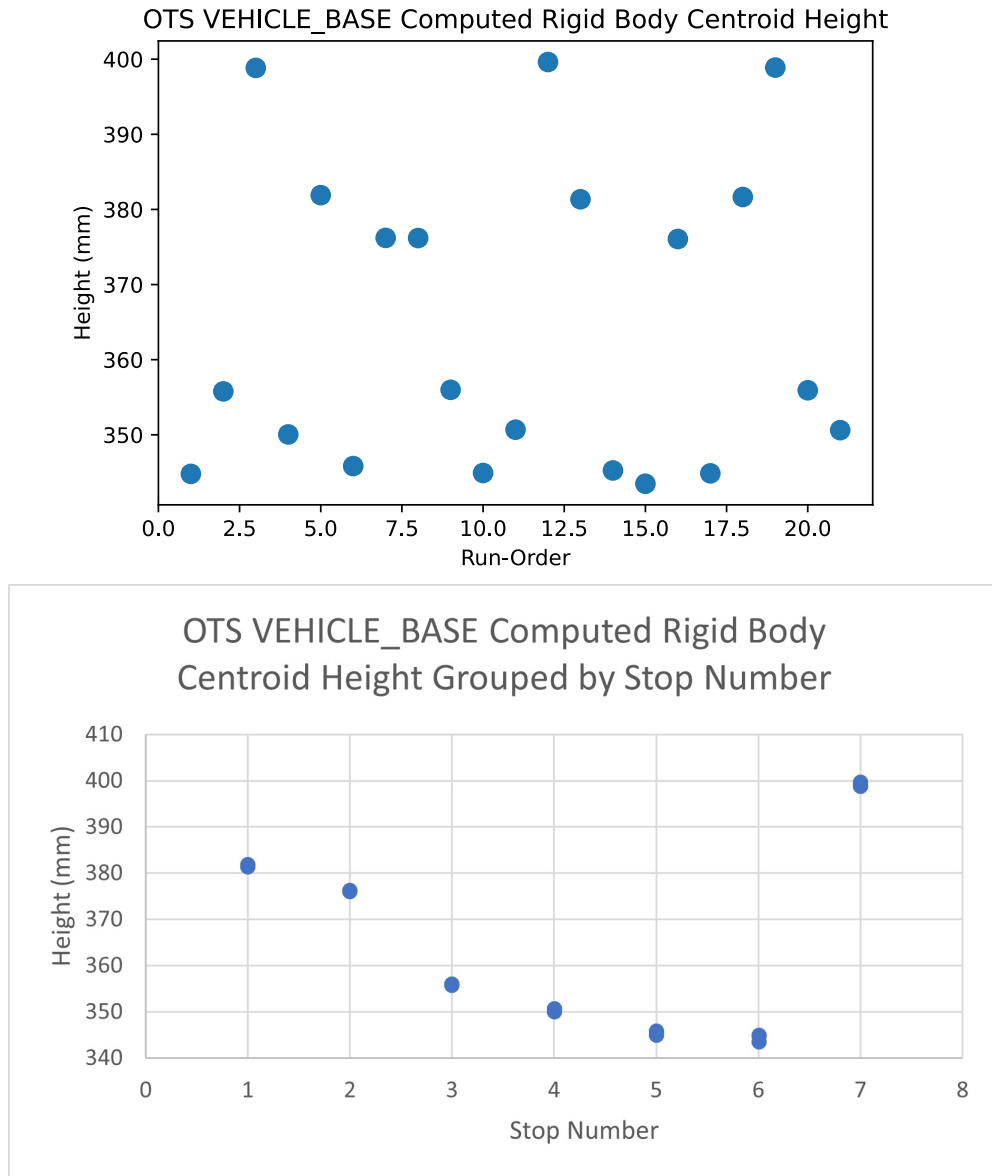


Fig. 15. Scatterplots showing the estimated height of the AMR-CT base centroid across all 21 data captures by run-order (top) and grouped by stop number (bottom).

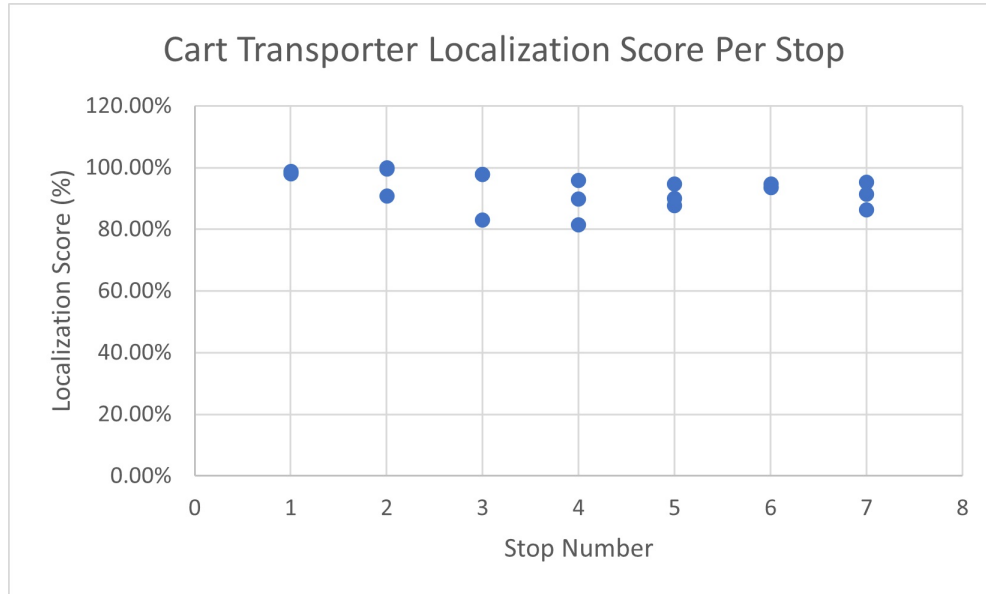


Fig. 16. Scatterplot of localization score vs stop number as determined by the AMR-CT controller for all 21 data captures.

3.1.4. Registration Measurement Results

The pre-processed data samples outputted from the Python script then served as input to another script written in a separate numerical computing programming language (version 9.7.0.1190202). The script read in the pre-processed pose data and applied existing code written to solve the 6DoF and 3DoF calibration sub-problems expressed by Eq. 1 and 2 using the appropriate closed-form solution [116, 117]. References [118–122] were also consulted while writing the portions of the script that plotted the calibration results.

Scatter plots comparing the transformed AMR-CT measured positions with the corresponding OTS measured positions in the *OTS* coordinate frame are presented in Fig. 18. The plots were visually inspected to validate that applying the calibrated coordinate transformation to AMR-CT measured position resulted in approximate alignment with the ground truth AMR-CT positions measured by the OTS. Additionally, comparing Fig. 18 (top), in which the full 6DoF calibrated coordinate transformation was used, against Fig. 18 (bottom), no visually noticeable difference was observed. This was supportive of the calibration procedure having adequately compensated for floor level variation between the selected stops.

Furthermore, the boxplots of Fig. 19 quantified the error between the AMR-CT measured positions (after being transformed to the *OTS* coordinate frame using the calibrated coordinate transformation) and the corresponding OTS measured positions. The error was computed both as the Euclidean distance (in mm) between corresponding points (far-left) and as component-wise differences (remaining right-ward boxplots). From the boxplots, it was observed that the central tendency of the Euclidean distances was between 15 mm and

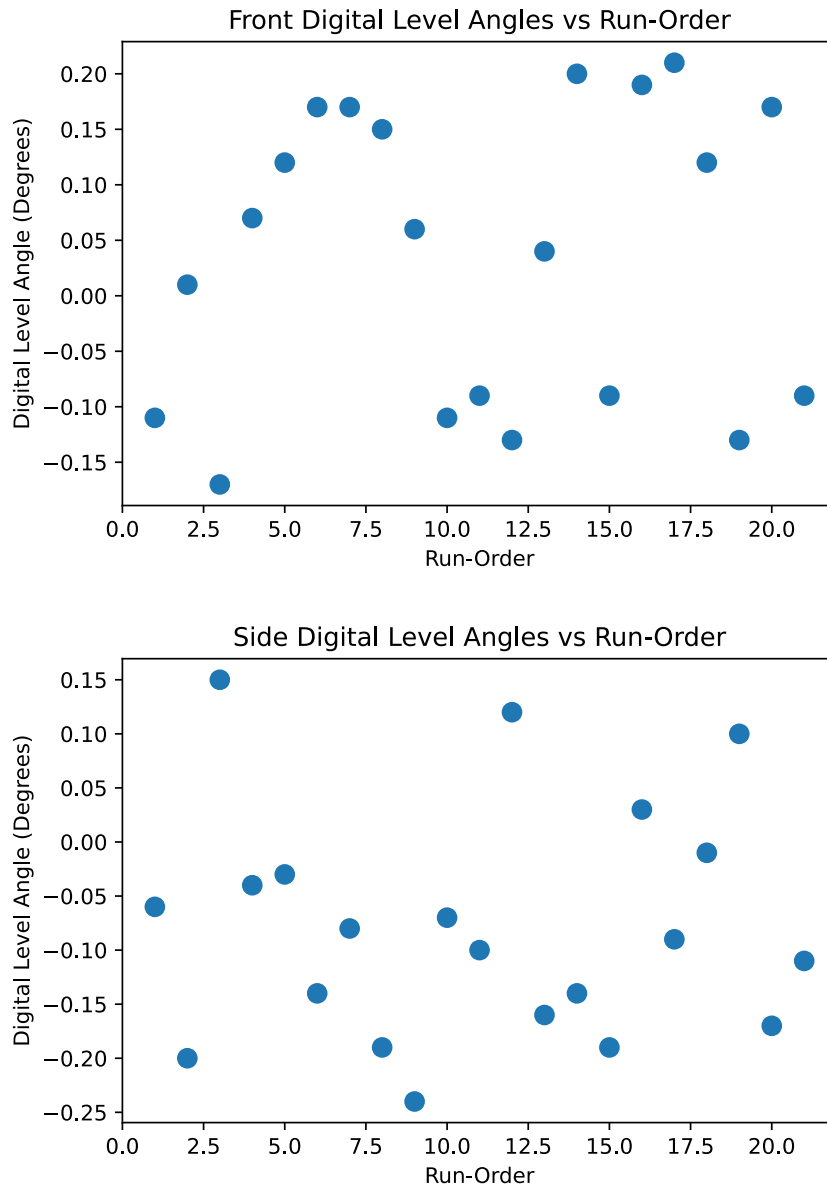


Fig. 17. Scatterplot of the front (top) and side (bottom) digital levels angle readings for all 21 data captures.

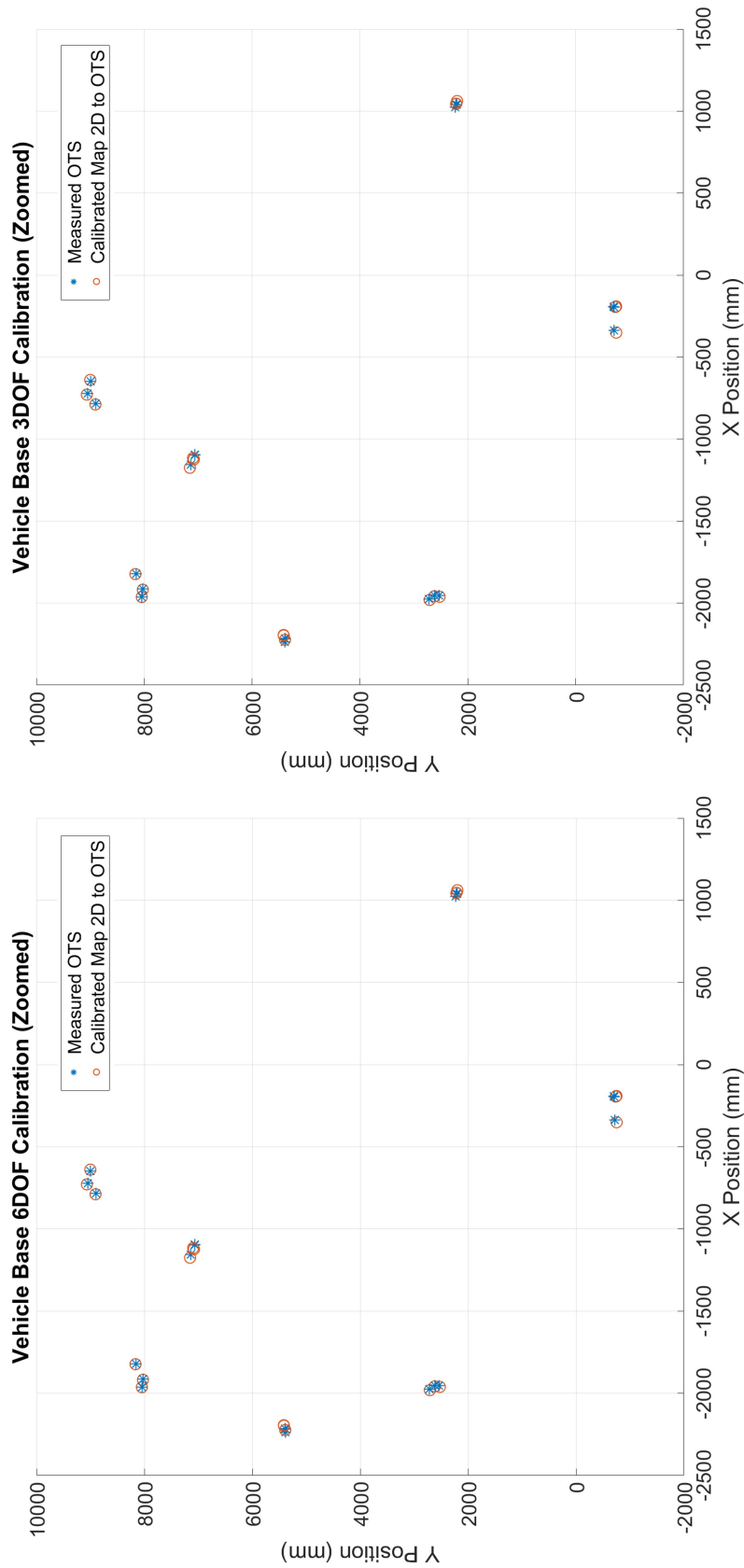


Fig. 18. Scatterplot comparing the AMR-CT controller points after applying the 6DoF (top) and 3DoF (bottom) calibrated coordinate transformations with the position of the AMR-CT as measured by the OTS.

20 mm and the Euclidean distances for the middle 50% of the positions were between 10 mm and approximately 32 mm. Furthermore, the maximum Euclidean distance was 41.348 mm while the minimum Euclidean distance was 2.348 mm. These results represented an improvement over a similar coordinate system transformation calibration conducted in Ref. [34] in which the minimum Euclidean distance between AMR-CT positions and OTS positions was over 10 mm and the maximum Euclidean distance between AMR-CT positions and OTS positions was over 60 mm. The results further supported that floor leveling did not likely impact the new coordinated system calibration since there was no difference in error between the positions transformed using the 6DoF transformation and the positions transformed using the 3DoF transformation. Furthermore, the component-wise differences between the positions suggest that most of the calibration error was present along the x and y position components and not the z component. Note that no outlier or potential outlier values were observed in the boxplots.

The final, measured coordinate system transformations were:

$$\{{\text{MAP}}_{2\text{D}}\} \mathbf{H}_{\{{\text{MAP}}_{3\text{D}}\}} = \begin{pmatrix} 1 & 0 & -0.005 & 0.022 \\ 0 & 1 & 0.006 & -0.020 \\ 0.005 & -0.006 & 1.000 & 3.148 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

and

$$\{{\text{MAP}}_{3\text{D}}\} \mathbf{H}_{\{{\text{OTS}}\}} = \begin{pmatrix} 1 & 0.001 & 0.003 & -3890.520 \\ -0.001 & 1 & 0 & 1890.553 \\ -0.003 & 0 & 1 & 372.133 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

However, it should be noted that an additional constant rotation of -90° was also applied when using these transformations prior to publishing AMR-CT poses in the `/ld_arcl_outgoing` node. This was done because the Marquette University code assumed that the primary direction of travel for the AMR was along the positive x axis of its OTS, but the NIST lab configuration required the primary direction of travel to be along the y .

3.2. Robot Model Transformations

3.2.1. Measurement Problem Summary

As mentioned in Sec. 2.3, the manipulator motion planners and ROS simulations required a model of the mobile manipulator in URDF format. Due to the differences between the AMR used by Marquette University and the AMR-CT used by NIST, as well as the differences in payload structure construction and manipulator base mounting, the remaining transformation that needed to be measured was that between the AMR-CT base and the manipulator base. Figure 20 depicts the coordinate systems and transformations required to measure this missing transformation using the OTS. First, let $vbase$, $vframe$, and $cbase$ denote the coordinate systems of the AMR-CT base, AMR-CT frame (i.e., the metal structure

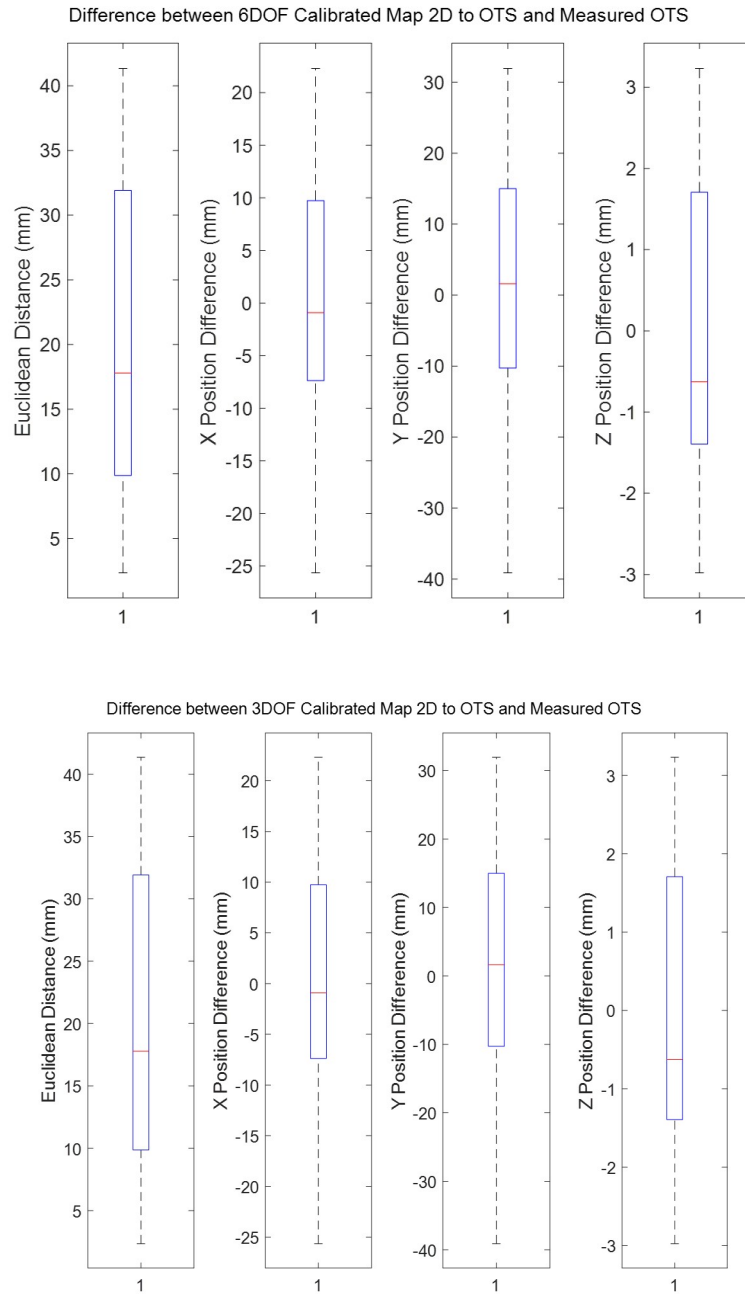


Fig. 19. Box plots showing the error between the AMR-CT controller measured positions after applying the 6DoF (top) and 3DoF (bottom) calibrated coordinate transformations and the position of the AMR-CT as measured by the OTS. The far-left boxplot presents the error as the Euclidean distance (in mm) between corresponding points while the other three boxplots presents the component-wise error as the difference between the AMR-CT controller measured positions and the OTS measured positions.

supporting the side laser scanners and holding the top user interface panel), and the manipulator base, respectively. The $vframe$ coordinate system was needed as an intermediary rigid body between the $vbase$ and $cbase$ coordinate frames because the rigid body associated with the $vbase$ coordinate frame would be occluded whenever the cart was attached to the AMR-CT. The homogeneous transformation matrices, ${}^{\{OTS\}}H_{\{vbase\}}$, ${}^{\{OTS\}}H_{\{vframe\}}$, and ${}^{\{OTS\}}H_{\{cbase\}}$ denote the relative poses of the AMR-CT base, AMR-CT frame, and manipulator base, respectively as observed by the OTS. Therefore, the objective was to find the unknown coordinate transformation, ${}^{\{vbase\}}H_{\{cbase\}}$, or the pose of the cart relative to the AMR-CT base. To accomplish this, the additional unknown transformations, ${}^{\{vbase\}}H_{\{vframe\}} = ({}^{\{OTS\}}H_{\{vbase\}})^{-1}{}^{\{OTS\}}H_{\{vframe\}}$ and

${}^{\{vframe\}}H_{\{cbase\}} = ({}^{\{OTS\}}H_{\{vframe\}})^{-1}{}^{\{OTS\}}H_{\{cbase\}}$, had to be measured such that

${}^{\{vbase\}}H_{\{cbase\}} = {}^{\{vbase\}}H_{\{vframe\}}{}^{\{vframe\}}H_{\{cbase\}}$. Since, in prior testing (see Sec. C.2 in Ref. [15]), the variability in alignment between the AMR-CT base and cart was not considered significant¹³, the unknown transformations were assumed to be approximately rigid.

3.2.2. Data Collection Procedure

The following OTS rigid bodies were created and defined for the measurement. First, the rigid body for the AMR-CT base (Fig. 21) was defined and named similarly to the VEHICLE.BASE rigid body of Fig. 13, albeit with a slightly different order in marker labeling. The marker configurations for rigid bodies for the vehicle frame, named “VEHICLE.FRAME”, and the manipulator base, named “CART.BASE”, are shown in Fig. 22 and 23, respectively. Note that, similar to the calibration in Sec. 3.1, the top light of the AMR-CT was covered during measurement.

Since the unknown coordinate transformations were assumed to be approximately rigid, the OTS was used directly to measure them. The OTS was re-calibrated prior to measurement using the same manufacturer-specified procedure as used in Sec. 2.5 and Sec. 3.1.2 (and also described in Sec. 2.5). The calibration summary displayed a mean ray error of 0.661 mm and mean wand error of 0.131 mm, which had the best quality classification of “exceptional” (as determined by the manufacturer).

The same seven locations from Fig. 12 were reused for measuring the rigid transformations and the measurement procedure consisted of two sets of data, each of which had a separate randomly-generated run sequence. For the first set, the AMR-CT, with its base and frame outfitted with OTS markers as shown in Fig. 21 and 22, was manually driven via joystick to each stop in the run sequence once. At each location, the OTS was used to record tracking data for each rigid body for 10 s. For the second set, the same procedure was repeated, except the OTS markers placed on the AMR-CT base were removed in order to attach the

¹³A standard deviation of 1.059 mm or less was measured, though it was observed that alignment variability may be impacted by external conditions, such as floor leveling.

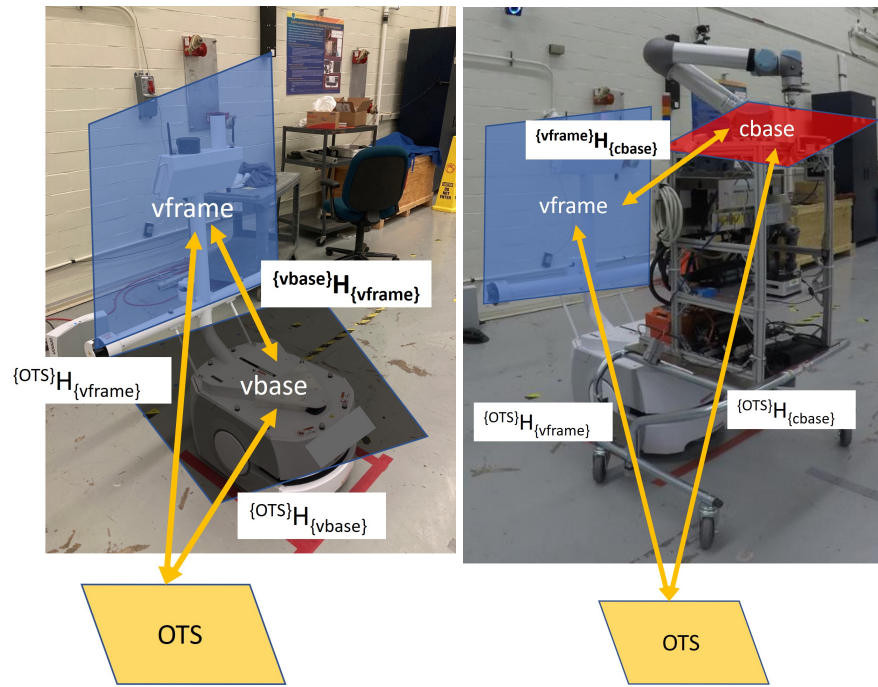


Fig. 20. Diagram of homogeneous transformations needed to determine the rigid transformation between the AMR-CT base and the manipulator base. Due to the cart payload structure obstructing the AMR-CT base while attached, the transformation was measured in two parts: The transformation between the AMR-CT base and the AMR-CT frame (left) and the transformation between the AMR-CT frame and the manipulator base (right).

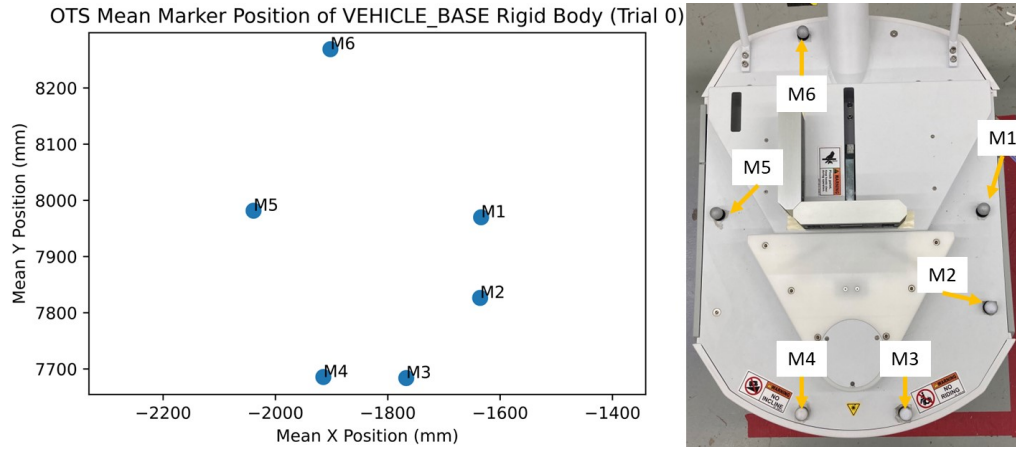


Fig. 21. Marker plot of the VEHICLE_BASE rigid body created for obtaining OTS data needed to calibrate the unknown rigid transformation between the AMR-CT base and the manipulator base (left). Corresponding picture of the physical marker placement on the AMR-CT (right). Note the slight difference in marker labeling in comparison to Fig. 13.

cart (with markers placed as in Fig. 23) to the AMR-CT. Note that, even though seven locations were recorded per set (a total of 14 data captures), only one location (preferably level and towards the center of the capture volume) per set would be sufficient in the future.

3.2.3. Data Pre-processing and Results

Data pre-processing was conducted similarly as described in Sec. 3.1.3, and the same software packages and versions were used. References [108, 110, 113–115] were also consulted while writing the script. First, data cleaning included adjusting the column names and plotting the percentage of marker occlusions for each rigid body. Again, no occlusions were observed [109]. The Cartesian position of each rigid body centroid and individual marker position was then aggregated by computing the component-wise average within each capture file. These average positions for the first capture file containing each rigid body were plotted in Fig. 21, 22, and Fig. 23 to verify the marker labels.

The component-wise standard deviation for each marker and each rigid body across all data captures was computed and displayed in box plots, as shown in Fig. 24, 25, 26, and 27. These plots were used to verify the repeatability of the captured static OTS data against previously obtained measurements and the established measurement repeatability of the OTS. From the plots, it was observed that the component-wise standard deviations for all markers across all rigid bodies were lower than the 0.26 mm dynamic rigid body tracking repeatability. The central tendencies of the component-wise standard deviations were generally greater than the 0.022 mm static rigid body tracking repeatability, but were still sub-millimeter across all markers and all rigid bodies. For the AMR-CT base (Fig. 24), the following outlier values were observed: one outlier value for marker M6 in the x component, one outlier each for markers M2, M3, and M4 in the y component, and one outlier

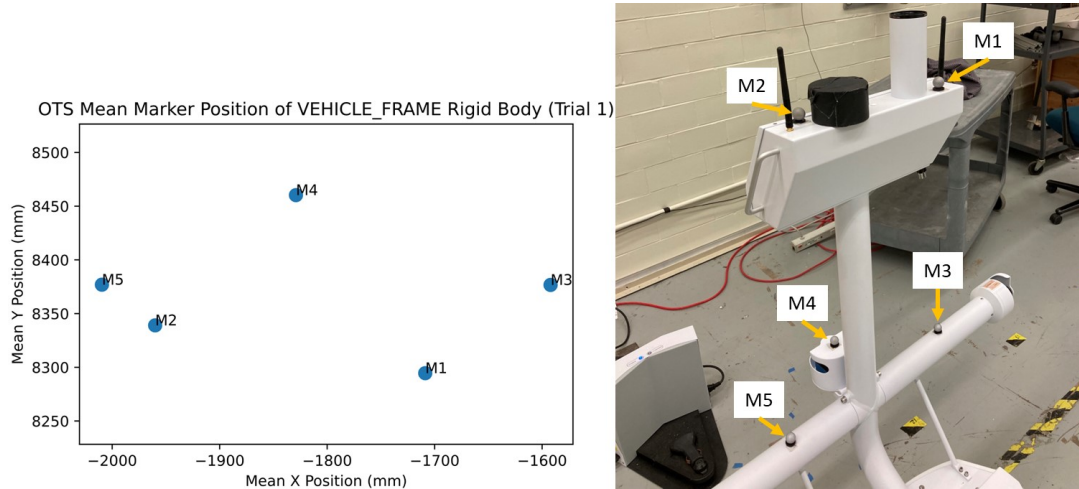


Fig. 22. Marker plot of the VEHICLE_FRAME rigid body created for obtaining OTS data needed to calibrate the unknown rigid transformation between the AMR-CT base and the manipulator base (left). Corresponding picture of the physical marker placement on the AMR-CT (right).

each for all markers except marker M3 in the z component. The outlier on marker M6 in the z component was the only instance where sub-millimeter repeatability was not observed in the AMR-CT base rigid body data. For the AMR-CT frame in Set 1 (Fig. 25), the following outliers were observed: one outlier for marker M3 in the x component, one outlier each for markers M3, M4, and M5 in the y component, and one outlier each for markers M1, M3, and M4 in the z component. The largest outlier, on marker M3 in the y component was less than 0.8 mm. For the AMR-CT frame in Set 2 (Fig. 26), the following outliers were observed: one outlier each for markers M2, M3, M4, and M5 in the x component, no outliers in the y component, and one outlier marker M4 in the z component. The largest outlier, on marker M5 in the x component was, less than 0.7 mm, however the maximum for the y component of marker M3 was larger and had a value less than 0.8 mm. Finally, for the cart base (Fig. 27), the following was observed. In the x component, all markers had at least one outlier, however marker M3 had an additional outlier with a value below the minimum. For the y component, outliers were observed on markers M2, M3, M4, and M5, with two outliers for marker M2 (again below the minimum). However, the maximums of marker M1 and M6 were larger than these outliers and less than 0.07 mm. Finally, for the z component, every marker except for marker M2 had at least one outlier, and markers M4 and M5 had an additional outlier below the minimum. The largest outlier was less than 0.8 mm. Overall, the OTS data quality was deemed sufficient for the purposes of the measurement, but showed potential for slight improvement. Since no complete occlusions were found and lower standard deviations values were observed at the same location for the AMR-CT base in Fig. 14, it was speculated that stray reflections could have affected measurement repeatability. Covering the cart base surface with non-reflective tape and covering the windows with black-out curtains could reduce such reflections in the future.

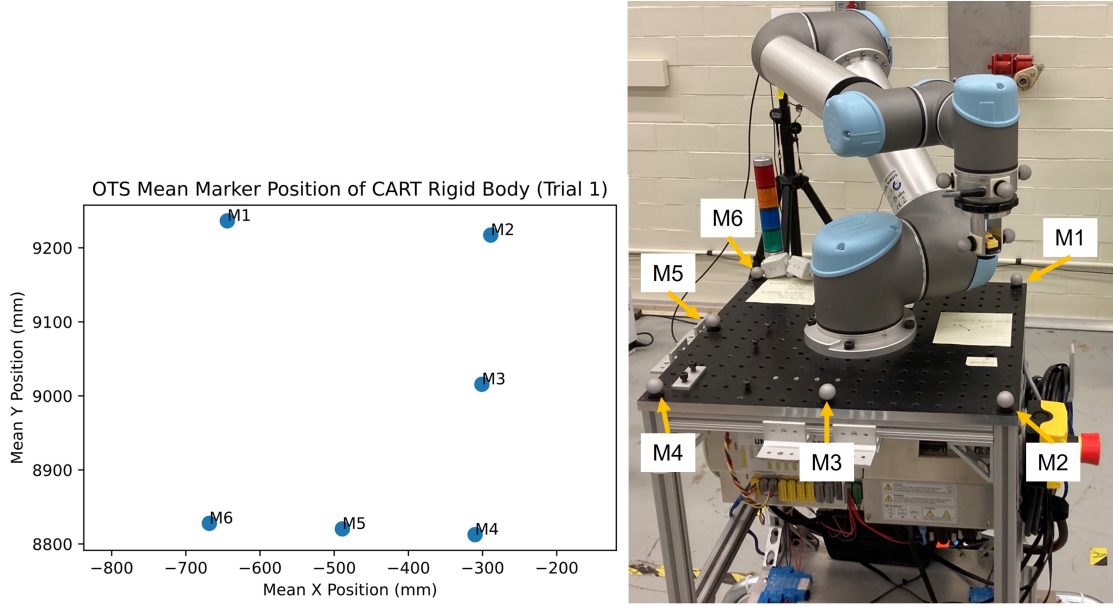


Fig. 23. Marker plot of the CART_BASE rigid body created for obtaining OTS data needed to calibrate the unknown rigid transformation between the AMR-CT base and the manipulator base (left). Corresponding picture of the physical marker placement on the cart (right).

Ultimately, the data from step 5 in Set 1 and step 6 in Set 2 was used to obtain the rigid transformations. The final, measured rigid transformations were:

$$\begin{aligned}
 \{{\text{vbase}}\} \mathbf{H}_{\{{\text{vframe}}\}} &= \begin{pmatrix} 0.996 & -0.0890 & -0.004 & -392.798 \\ 0.089 & 0.996 & -0.011 & 9.057 \\ 0.005 & 0.011 & 0.100 & 528.830 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\
 \{{\text{vframe}}\} \mathbf{H}_{\{{\text{cbase}}\}} &= \begin{pmatrix} -0.643 & -0.766 & 0.011 & 519.248 \\ 0.766 & -0.643 & 0.012 & -51.719 \\ -0.002 & 0.016 & 0.100 & 290.463 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\
 \{{\text{vbase}}\} \mathbf{H}_{\{{\text{cbase}}\}} &= \begin{pmatrix} -0.708 & -0.706 & 0.006 & 127.859 \\ 0.706 & -0.708 & 0.002 & 0.505 \\ 0.003 & 0.006 & 0.100 & 821.242 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (3)
 \end{aligned}$$

To serve as input to the URDF model, a few final adjustments were made to $\{{\text{vbase}}\} \mathbf{H}_{\{{\text{cbase}}\}}$. The rotation component of $\{{\text{vbase}}\} \mathbf{H}_{\{{\text{cbase}}\}}$ was converted to RPY format (in Radians) using the 3D rotation converter in Refs. [123, 124], and the translation was converted from millimeters to meters. Additionally, it was discovered that the manipulator base coordinate frame origin within the existing manipulator URDF model was flipped 180°, therefore an

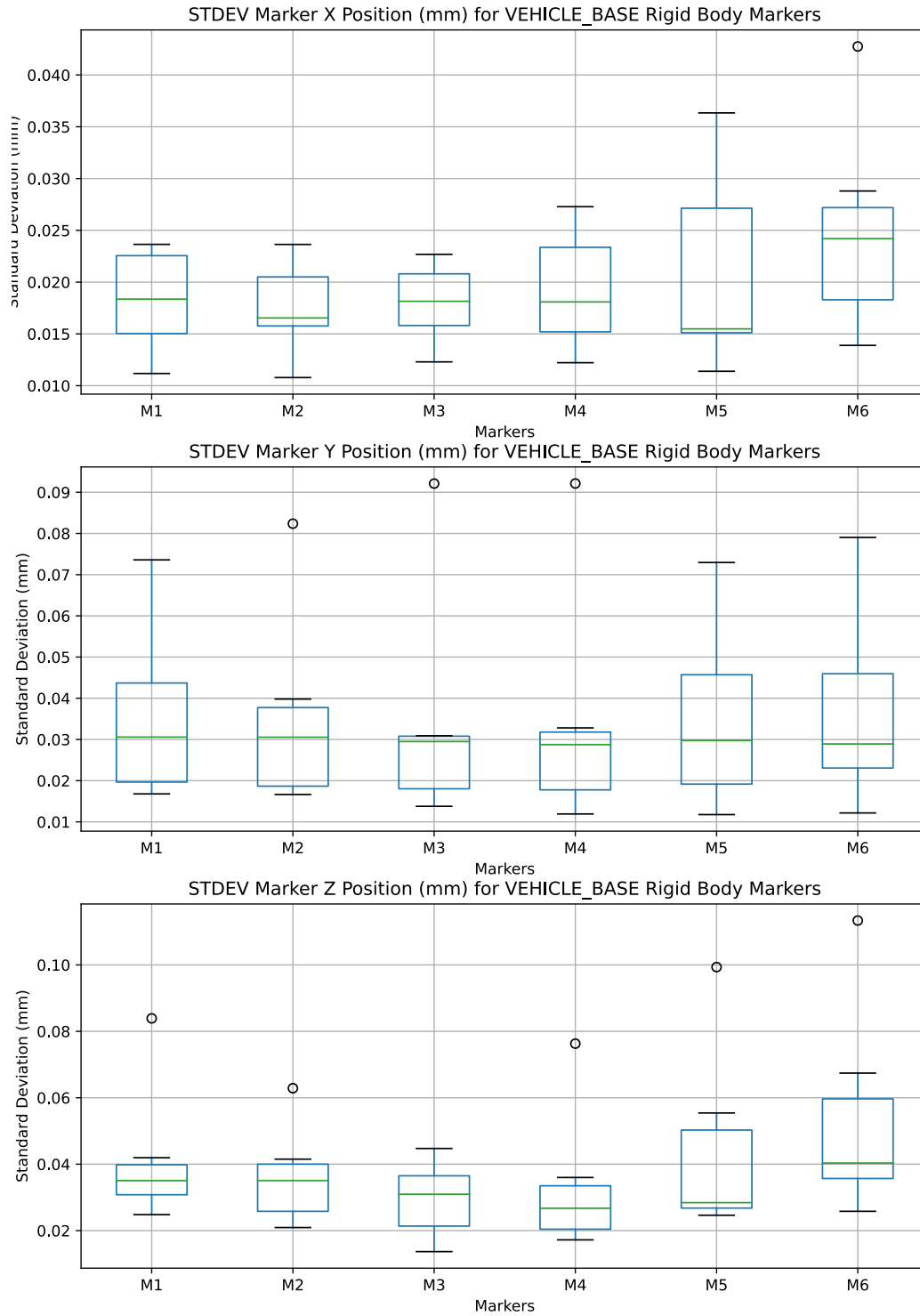


Fig. 24. Boxplots of the component-wise standard deviations for each marker of the VEHICLE_BASE rigid body and for each data capture obtained during the rigid transformation measurement.

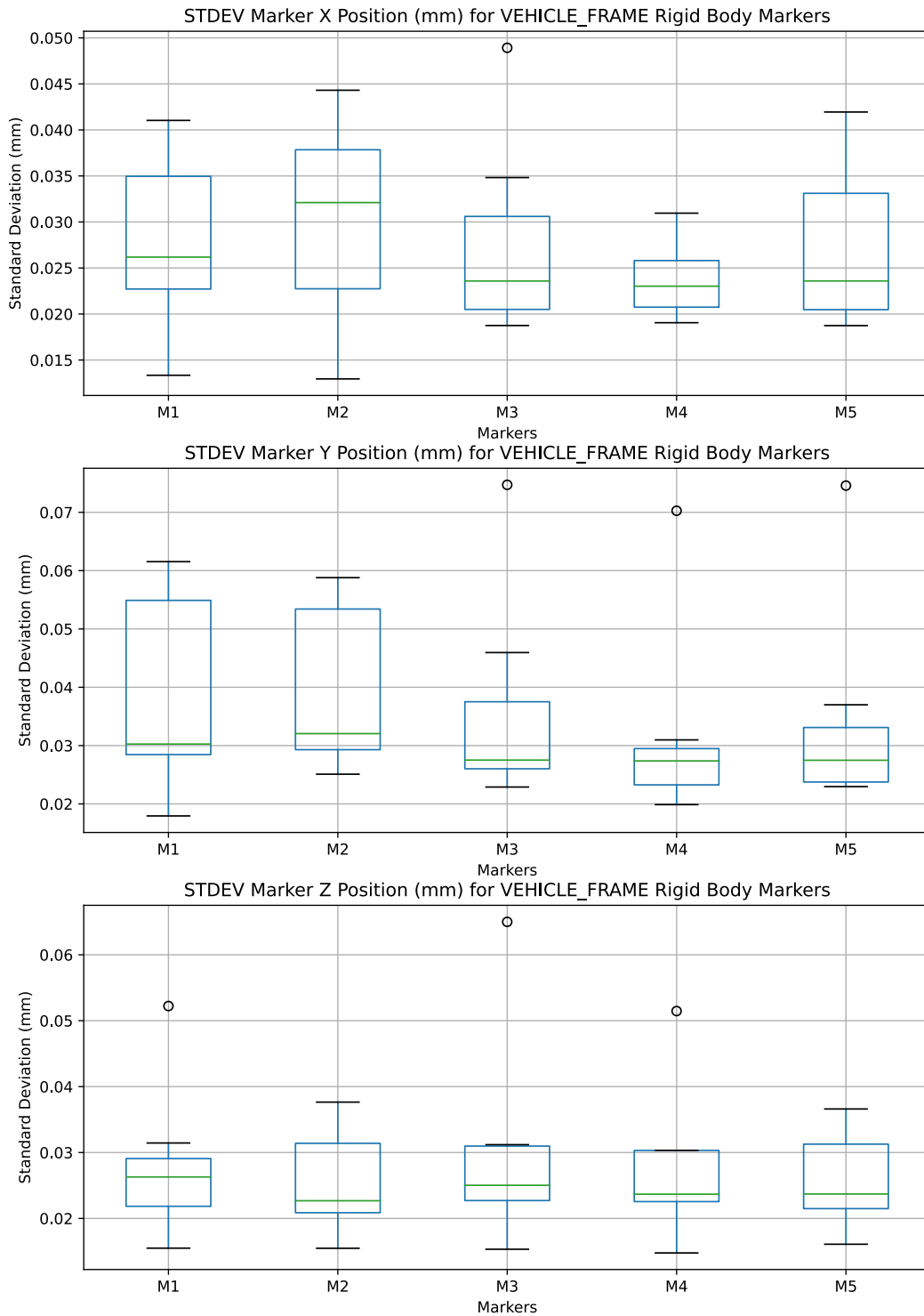


Fig. 25. Boxplots of the component-wise standard deviations for each marker of the VEHICLE_FRAME rigid body and for each data capture obtained during Set 1 of the rigid transformation measurement.

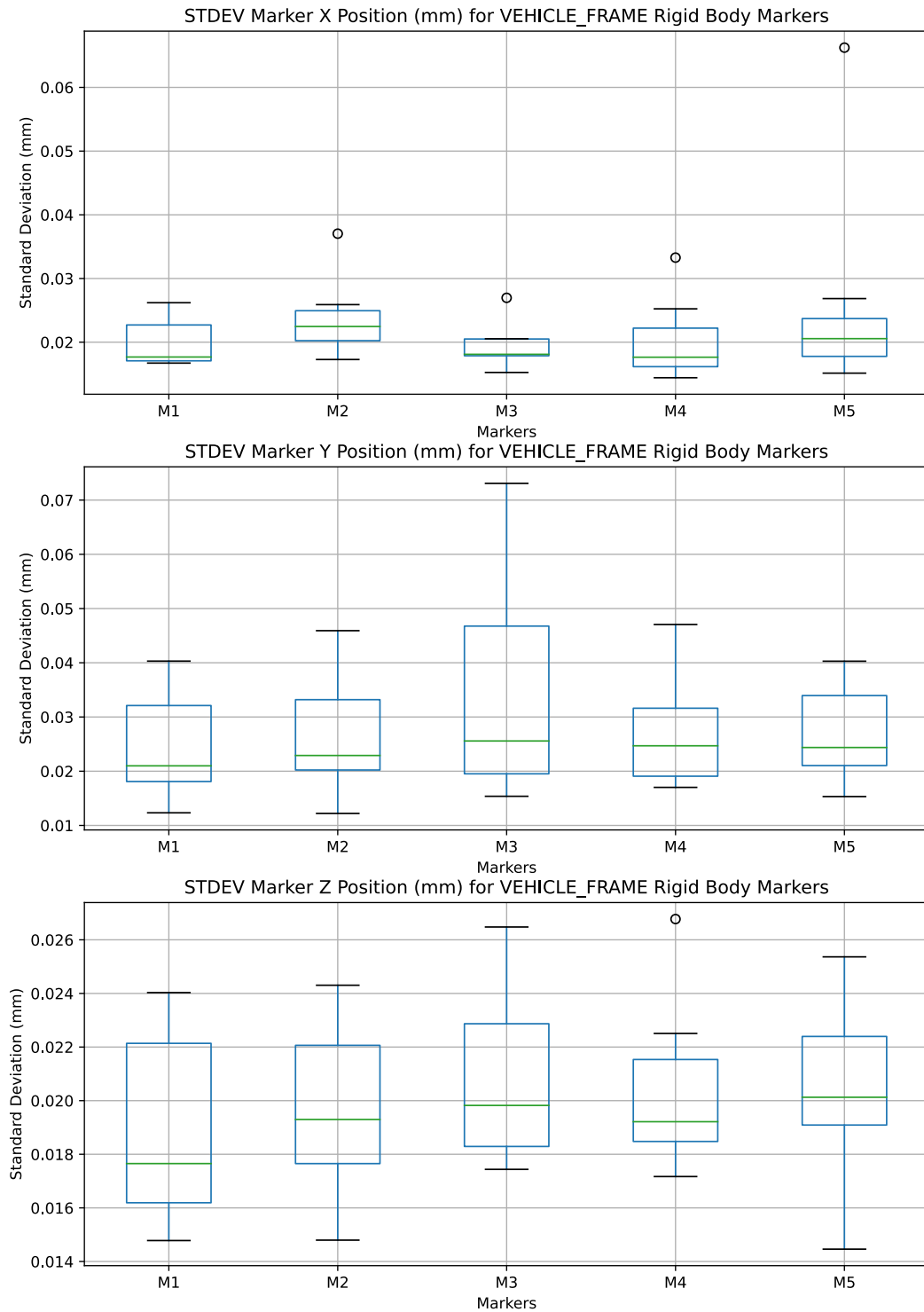


Fig. 26. Boxplots of the component-wise standard deviations for each marker of the VEHICLE_FRAME rigid body and for each data capture obtained during Set 2 of the rigid transformation measurement.

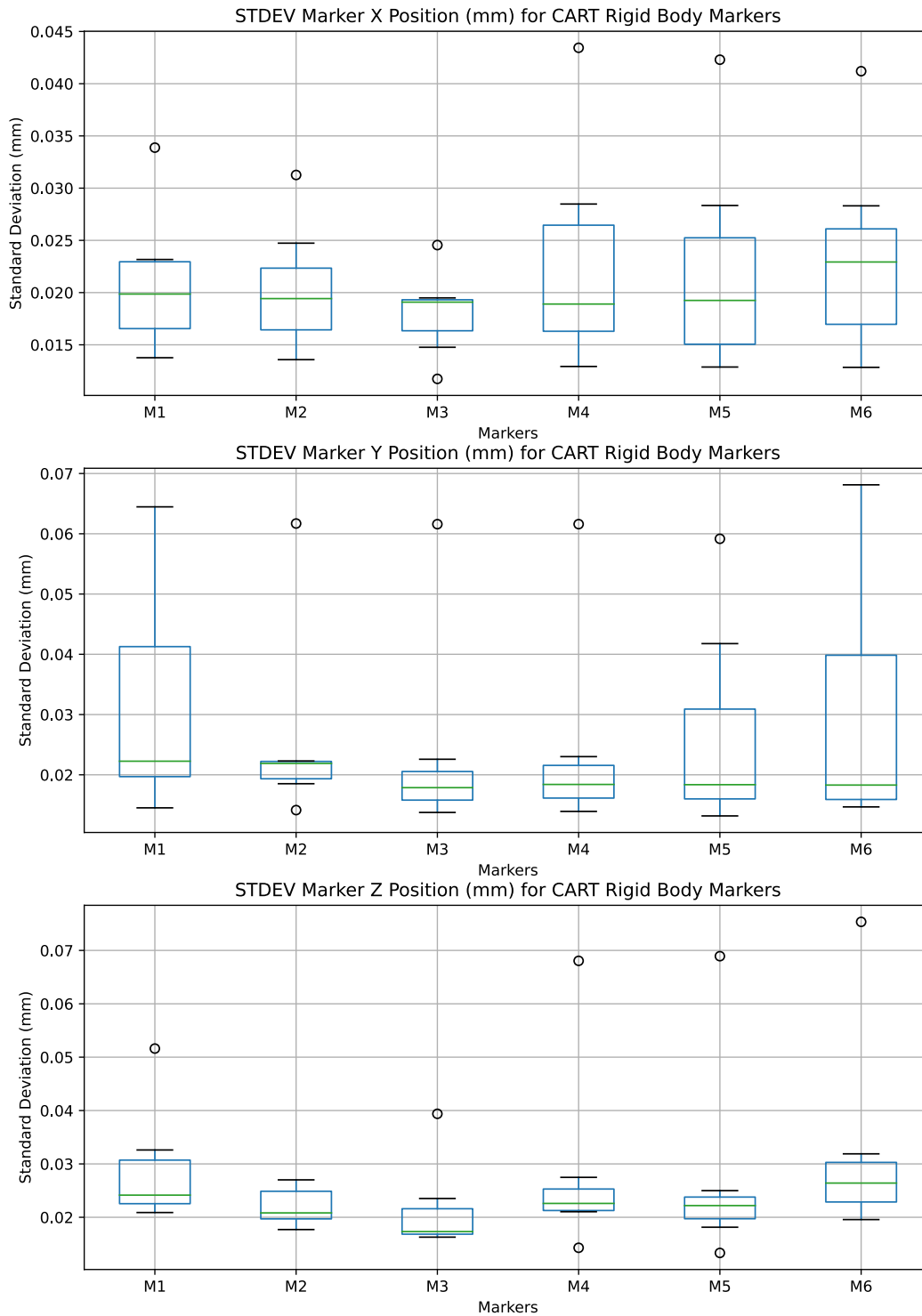


Fig. 27. Boxplots of the component-wise standard deviations for each marker of the CART.BASE rigid body and for each data capture obtained during Set 2 of the rigid transformation measurement.

additional constant rotation of 180° was also applied to ${}^{\{vbase\}}H_{\{cbase\}}$ before entering into the URDF model.

3.2.4. Updated Robot Model

The robot model was updated to accommodate the differences in transformations and hardware configuration between the Marquette AMR mobile manipulator and NIST AMR-CT mobile manipulator. This required making changes to several existing files generated in the moveIt! configuration. First, the XML Macro (xacro) file defining the URDF model was modified as follows. The full model of the Marquette AMR was no longer needed and was therefore replaced with a single static link, called “base_link”. The joint going from “base_link” to “arm_base_link” (i.e., the AMR-CT base and the manipulator base) was replaced with the measured coordinate system transformation from Eq. 3. Additional unneeded links and/or associated joints for the AMR laser scanner and an additional camera were also removed. Finally, new links were created to assist the manipulator trajectory planner with collision avoidance. These included representations for the AMR-CT Acuity camera sensor (called “ld_acuity_link”), the AMR-CT touchscreen panel (called “ld_panel_link”, and the emergency light of the manipulator, called “ur5_light_joint”. Since these links were only included for collision avoidance purposes, their transformations were approximately measured via tape measure for specification as URDF joints. An additional Semantic Robot Description Format (SRDF) file in the moveIt! configuration was also updated to remove declarations for deleted links and/or joints and add declarations for the new links and/or joints. The check_urdf tool was used to validate the syntax of the updated URDF model and the view_frames tool was used to generate the visual representation of the transformation tree shown in Fig. 28, which was also used for validating the correctness of the updated model [125, 126]. In the ROS controller’s configuration file, the publishing rate of the joint_state_controller (see Ref. [127]) to the joint_states topic, which is subscribed to by the robot_state_publisher node (see Ref. [128]) was changed from 300 Hz to 125 Hz to match the maximum controller rate of the NIST AMR-CT mobile manipulator [129].

3.3. Fiducial Ground Truth Measurements

The final set of OTS position measurements consisted of acquiring a ground truth reference for initial fiducial target locations on the CMMA ¹⁴. Note that this data was acquired the same day as the data in Sec. 3.1 and used the same OTS calibration (see Sec. 3.1.2). The measurement procedure simply consisted of first placing 19 mm diameter OTS markers on the CMMA such that they overlapped with the retro-reflective target fiducial positions to be tested (see Fig. 29). Two additional markers (M1 and M11) were included ensure asymmetry of the resulting rigid body, which prevents marker mis-labelings from occurring. Then,

¹⁴Recall from Sec. 3 that these were needed to allow for global planning to simplify velocity compensation [9, 10]

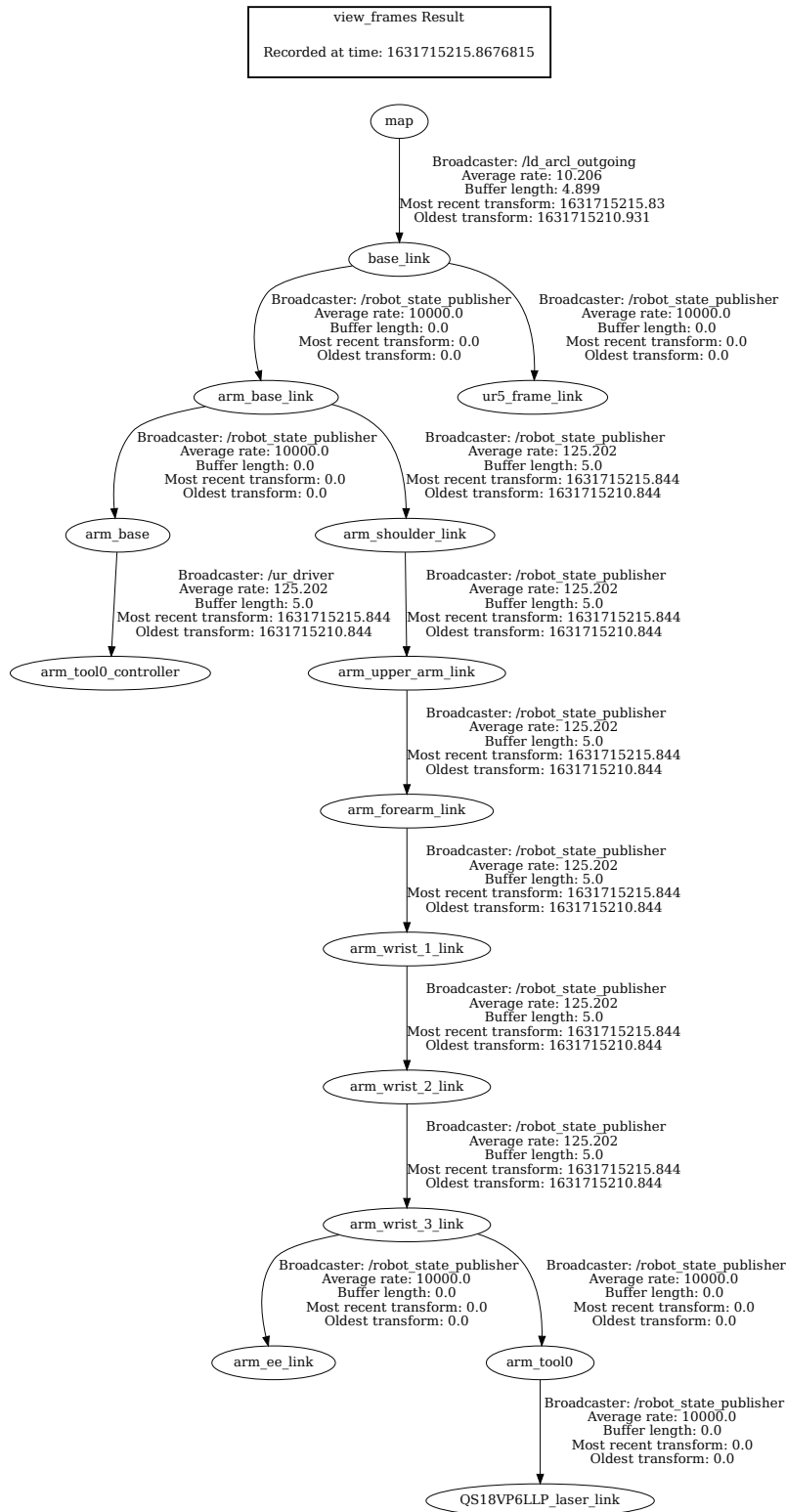


Fig. 28. Output of the ROS view_frames transform debugging tool for updated robot model.

five, 10 s static captures of the resulting CMMA rigid body was obtained using the OTS and exported to a CSV file.

Data pre-processing was again conducted similarly to Sec. 3.1.3 and Sec. 3.2.3 with the same software packages and versions. References [107, 108, 110] were also referred to in developing the data pre-processing script. As shown in Fig. 30, a very small number of occlusions (less than 0.5% of the total data capture) were observed for two of the five data captures. However, the data capture selected for use (capture 1) had no occlusions. Again, the component-wise standard deviation for each marker of the CMMA rigid body across all data captures was computed and displayed in box plots, as shown in Fig. 31. This type of plot was again used to verify the repeatability of the ground truth initial fiducial position measurements to ensure no significant data capture anomalies, such as occlusions, reflections, and marker mis-labelings, occurred. The plot shows that the component-wise standard deviations were within the previously established static OTS measurement uncertainty of 0.022 mm [14, 95].

The list of initial fiducial target positions measured using the OTS is given in Table 2. To reduce errors caused by misalignment between the RLS detection distance and angle offset [90], the initial search positions were updated by running a single sample trial of the continuous mobile manipulator experiment on the real-mobile manipulator hardware. The updated fiducial positions are given in Table 3. The spiral search method and slow base speed of 0.01 m/s was used such that all fiducial targets could be detected by the manipulator EOAT. The logged global coordinate pose of the EOAT upon each fiducial interception was substituted as the final initial search positions for the full continuous mobile manipulator performance experiment (to be detailed in Sec. 4).

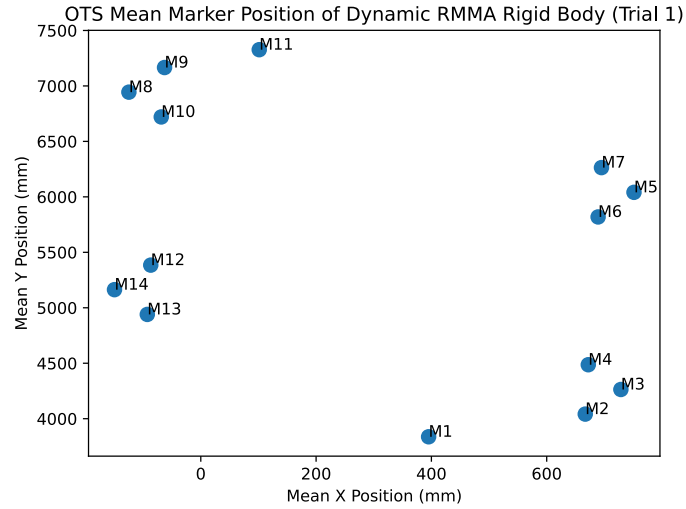


Fig. 29. Scatterplot showing the measured ground truth initial positions of the retro-reflective fiducials, as measured by the OTS. Note that markers M1 and M11 were included to introduce rigid body asymmetry to improve tracking quality.

Table 2. List of initial positions for the retro-reflective fiducial targets on the CMMA as measured by the OTS.

Fiducial	Marker	x (mm)	y (mm)
Side 1			
1	M2	4041.458	-666.871
2	M3	4263.348	-728.657
3	M4	4486.508	-672.135
4	M6	5818.917	-689.166
5	M5	6040.750562	-751.380916
6	M7	6263.738	-695.006
Side 2			
7	M9	7165.920	062.870
8	M8	6943.980	124.718
9	M10	6720.430	68.559
10	M12	5384.589	86.803
11	M14	5163.315	149.684
12	M13	4940.144	92.783

Table 3. Final list of initial search positions for the retro-reflective fiducial targets^a.

Fiducial	x (mm)	y (mm)	Diameter (mm)
Side 1			
1	3938.4	-564.3	6.4
2	4232.7	-679.0	6.4
3	4553.4	-558.4	3
4	5696.3	-577.1	6.4
5	6014.7	-708.5	6.4
6	6341.1	-576.1	3
Side 2			
7	7277.3	13.3	6.4
8	6947.4	133.0	6.4
9	6625.2	1.9	3
10	5470.1	4.2	6.4
11	5142.0	153.8	6.4
12	4827.9	16.5	3

^a Obtained from running a preliminary continuous mobile manipulator performance trial of the spiral search method at a base speed of 0.01 m/s and recording the EOAT pose at each fiducial detection.

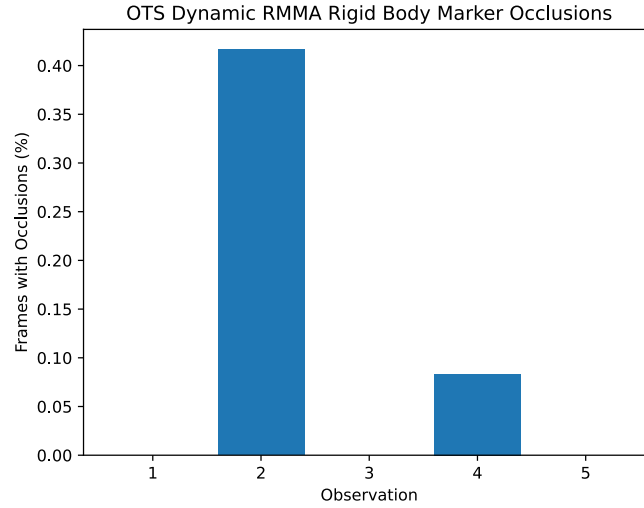


Fig. 30. Bar chart showing the percentage of occlusions for each of five, 10 s OTS captures of the ground truth fiducial positions on the CMMA.

4. Experiment Design

4.1. Experimental Model

The design of the 2^3 factorial experiment is now detailed. As the goal was to determine which of the three factors (i.e., search method, AMR-CT speed, and measured CMMA side) had a statistically significant effect on manufacturing performance (i.e., measured by either the percentage of intercepted fiducials or the mean of search times for all fiducials in seconds), the primary purpose was considered to be factor screening /characterization [130]. To simplify the experimental design, separate statistical analyses were conducted for each of the two response variables. However, when conducting hypothesis tests, this may have slightly increased the experiment-wise error rate (that is, the Type I error of all the hypothesis tests), since only the Type I error of each individual hypothesis test would be controlled [131–133]. Note also that the factors, search method and measured CMMA side, are qualitative, while the factor, AMR-CT speed is quantitative¹⁵. Each factor included two fixed levels, to be detailed shortly, which resulted in a fixed effects design. The 2^3 factorial design was selected since it potentially allows for testing of first and second order interactions amongst the three factors [136, 137].

The initial experimental model is given by Eq. 4 [138]. Here, y_{abc} was the recorded re-

¹⁵Note that the 2^3 factorial design assumes that the response is approximately linear over the range of the factor levels [134]. Center points may be added to the model to improve robustness against curvature introduced by interaction effects, however, this is only possible if all factors are quantitative [135]. Since the linearity property need only be very approximate and no additional information was available, this assumption was deemed reasonable [134, 135].

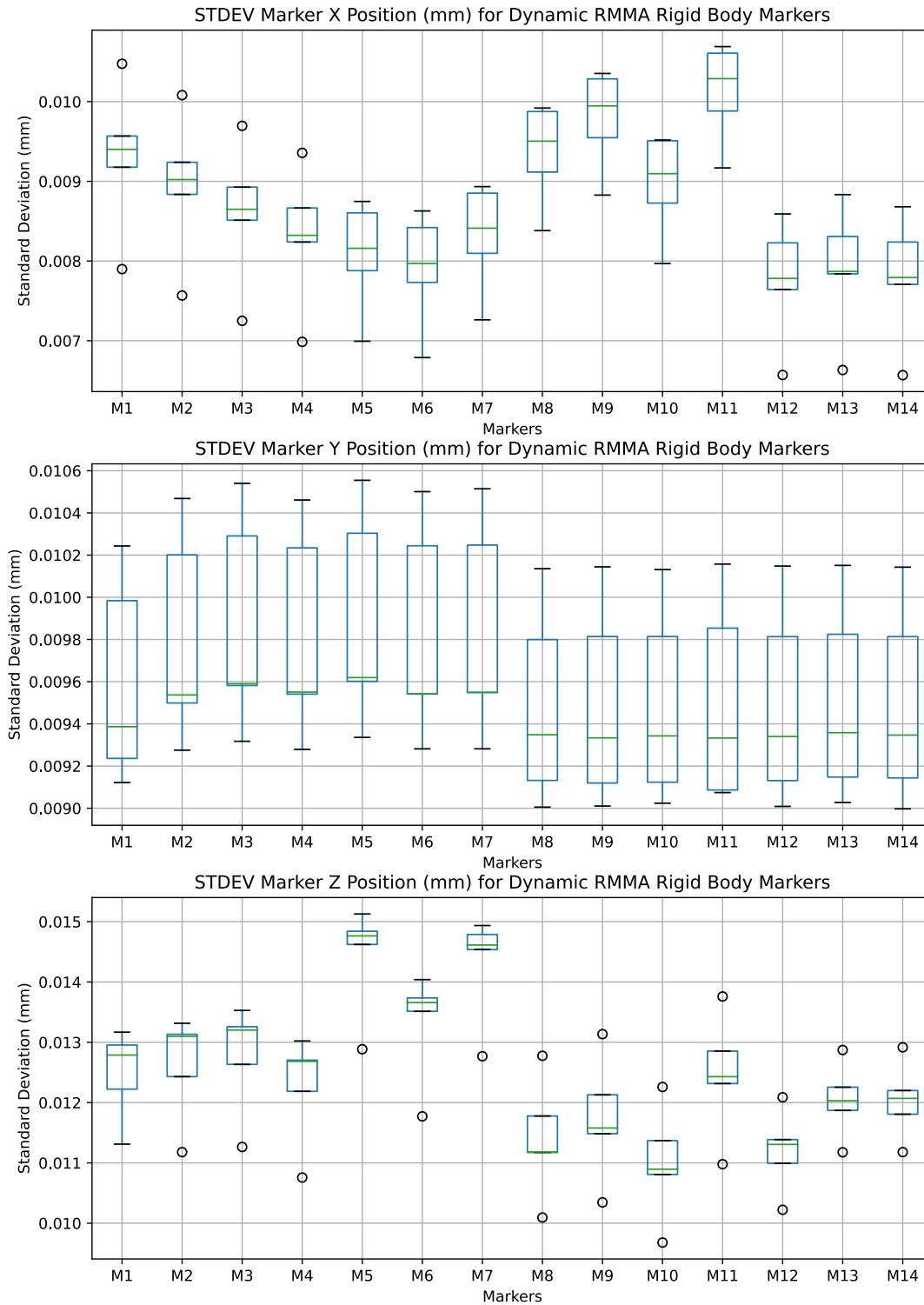


Fig. 31. Boxplots showing the component-wise standard deviation per OTS marker on the CMMA.

sponse (again, either the percentage of intercepted fiducials or the mean search time in seconds) and μ was the effect of the overall, constant response mean. The effect due to the a -th level of the factor search method was given by τ_a , where $a = 1$ was the deterministic spiral search and $a = 2$ was the UKF search. The effect due to the b -th level of the factor AMR-CT speed was given by β_b , where $b = 1$ was the “low” AMR-CT speed of 0.01 m/s and $b = 2$ was the “high” AMR-CT speed of 0.025 m/s. The effect due to the c -th level of the factor measured CMMA side was given by γ_c , where $c = 1$ is Side 1, which was closest to the lab window, and $c = 2$ is Side 2, which was closest to the interior lab wall. The first-order interactions were denoted as follows. The interaction between search method and AMR-CT speed was denoted by $(\tau\beta)_{ab}$. The interaction between search method and measured CMMA side was denoted by $(\tau\gamma)_{ac}$. Then, the interaction between AMR-CT speed and measured CMMA side was denoted by $(\beta\gamma)_{bc}$. The second order interaction between search method, AMR-CT speed, and measured CMMA side was given by $(\tau\beta\gamma)_{abc}$. Finally, ε_{abcd} was the effect due to random experimental error, where $d = 1, 2, \dots, n$ and n was the number of replicates. Six complete replicates were collected for each of the 2^3 (eight) factor combinations resulting in a balanced design of 48 total observations.

For each of the two response variables, the hypotheses given in Eq. 4 - 11 were tested using an appropriate statistical test. The decision rule for each hypothesis test was to reject the null hypothesis if the corresponding p-value was less than 0.05. Otherwise, the null hypothesis was not rejected. Note that, if an Analysis of Variance (ANOVA) were to be used to tests these hypotheses, then it is assumed $\varepsilon_{abcd} \sim NID(0, \sigma^2)$ (i.e., the errors are assumed to be normally, independently distributed with equal variance) [139, 140]. Model adequacy checks were conducted for these assumptions and presented in Sec. 5.3 and used to select the appropriate statistical test used for data analysis [140].

$$y_{abc} = \mu + \tau_a + \beta_b + \gamma_c + (\tau\beta)_{ab} + (\tau\gamma)_{ac} + (\beta\gamma)_{bc} + (\tau\beta\gamma)_{abc} + \varepsilon_{abcd} \begin{cases} a = 1, 2 \\ b = 1, 2 \\ c = 1, 2 \\ d = 1, 2, \dots, n \end{cases} \quad (4)$$

$$\begin{aligned} \mathcal{H}_0 : (\tau\beta\gamma)_{abc} &= 0 \text{ for all } a, b, c \\ \mathcal{H}_a : \text{At least one } (\tau\beta\gamma)_{abc} &\neq 0 \end{aligned} \quad (5)$$

$$\begin{aligned} \mathcal{H}_0 : (\tau\beta)_{ab} &= 0 \text{ for all } a, b \\ \mathcal{H}_a : \text{At least one } (\tau\beta)_{ab} &\neq 0 \end{aligned} \quad (6)$$

$$\begin{aligned} \mathcal{H}_0 : (\tau\gamma)_{ac} &= 0 \text{ for all } a, c \\ \mathcal{H}_a : \text{At least one } (\tau\gamma)_{ac} &\neq 0 \end{aligned} \quad (7)$$

$$\begin{aligned}\mathcal{H}_0 &: (\beta\gamma)_{bc} = 0 \text{ for all } a, c \\ \mathcal{H}_a &: \text{At least one } (\beta\gamma)_{bc} \neq 0\end{aligned}\tag{8}$$

$$\begin{aligned}\mathcal{H}_0 &: \tau_1 = \tau_2 = 0 \\ \mathcal{H}_a &: \text{At least one } \tau_a \neq 0\end{aligned}\tag{9}$$

$$\begin{aligned}\mathcal{H}_0 &: \beta_1 = \beta_2 = 0 \\ \mathcal{H}_a &: \text{At least one } \beta_b \neq 0\end{aligned}\tag{10}$$

$$\begin{aligned}\mathcal{H}_0 &: \gamma_1 = \gamma_2 = 0 \\ \mathcal{H}_a &: \text{At least one } \gamma_c \neq 0\end{aligned}\tag{11}$$

4.2. Summary of Collected Data

Note that the full data-set collected as part of the experiment is available in Ref. [35]. The data collected prior to the experiment included the following:

- Data pertaining to the AMR-CT map and configuration, including [58]:
 - File containing the full AMR-CT general configuration (e.g., interface settings such as those related to the clear-text API, operation settings impacting behavior such as localization and path planning, central controller configuration, and debug information.) Note that modifications to these settings were outlined in Sec. 2.2.
 - File containing the AMR-CT model configuration, which includes values related to the robot type, sensor types and transformations, firmware configuration, and movement maximums among others.
 - File containing calibrations such as general robot physical dimensions and information and their respective parameters.
 - The raw laser scans for the map generated as per 2.2.
 - Light localization scan file, however, it should be noted that the light acuity sensor of the AMR-CT was disabled.
 - The map file, which, in addition to the laser scans, also includes the configured goal points, exclusion areas, etc., as configured per Sec. 2.2 and depicted in Fig. 4.
 - Screenshot of the map file (i.e., Fig. 4).

- Full AMR-CT debug info file.
- OTS calibration file [93] and screenshot of OTS calibration summary.
- OTS ground truth reference data of the fiducial target positions on the CMMA, similar to that documented in Sec. 3.3. Data includes both tracking files and exported as CSV as per Sec. 2.5. See also Sec. 4.3.
- Full backup of the ROS catkin workspace containing all code used to run the experiment [141].
- Full backup of the data and code described from Sec. 3.1 and Sec. 3.2.
- Screenshots verifying the time synchronization of clients as per Sec. 2.6. Note that this data was collected immediately prior to the start of each replicate.

The data collected during the experiment consisted of the following:

- A matrix file, readable by numerical computing software including but not limited to those in Refs. [142, 143]). The matrix contained the following data, for each experimental run, in addition to the primary response variables already described:
 - An array of 8 computer date and wall clock time strings for the start each run.
 - An $8 \times 6 \times 2$ matrix of 2D AMR-CT positions, in m, corresponding to when a fiducial was intercepted by the RLS.
 - An $8 \times 6 \times 2$ matrix of 2D AMR-CT positions, in m, corresponding to when the manipulator began searching for a fiducial.
 - An array of size 8 indicating the speed of the AMR-CT for each run in m/s.
 - An $8 \times 6 \times 2$ matrix of 2D EOAT positions, in m, corresponding to when a fiducial was intercepted by the RLS.
 - An $8 \times 6 \times 2$ matrix containing the component-wise position error, in m, between the EOAT and each fiducial immediately prior to searching for each run.
 - An $8 \times 6 \times 2$ matrix of 2D EOAT positions, in m, corresponding to when the manipulator began searching for a fiducial.
 - An array of size 8 indicating the max search radius of the spiral search conducted at the first marker, in m and for each run.
 - An array of size 8 containing the interception rate for each run as already described.
 - An 8×6 matrix of binary flags indicating which individual fiducials were intercepted for each run.

- An $8 \times 6 \times 2$ matrix of component-wise 2D errors between the EOAT after detecting a fiducial and the initial search position, in m.
- An array of size 8 containing distance thresholds, in m, determining when a fiducial was out of range for each run.
- An array of size 8 containing the average fiducial search time for each run as already described.
- An array of size 8 containing the max search radius of the spiral search conducted for all other markers, in m and for each run.
- An array of size 8 containing the replicate number.
- An array of size 8 containing integers (1 and 2) indicating the side of the CMMA being tested for each run.
- An array of size 8 containing the starting ROS time of each run.
- An array of size 8 containing the run numbers (0-7).
- An array of size 8 containing strings indicating which search algorithm was used for each run.
- An 8×6 matrix of binary flags indicating which individual fiducials were skipped in each run.
- An array of size 8 containing counts of the total number of skipped markers for each run.
- Several hyper-parameters and outputs of the UKF model for each run where it was used [10]:
 - * An array of size 8 containing the covariance value used to initialize the other covariance values (hyper-parameter) for each run.
 - * An array of size 8 containing the cumulative innovation for each run.
 - * An array of size 8 containing the sample gap (hyper-parameter) used for each run.
 - * An array of size 8 containing the process noise covariance for each run, denoted Q (hyper-parameter).
 - * Arrays of size 8 containing the observation noise covariance for each run, denoted R (hyper-parameter) and for two cases:
 - When a fiducial had successfully been intercepted.
 - When a fiducial failed to be intercepted.

- * An array of size 8 containing the initial estimate noise covariance, denoted P (hyper-parameter).
 - * An 8 x 2 matrix of noise values for each run, which went unused (hyper-parameter).
 - * An 8 x 6 x 4 matrix containing the UKF prediction for each marker and each run.
- Another matrix file containing the 2D AMR-CT pose and EOAT pose, in m, for each ROS timestep of each experimental run and replicate [144]. The intended velocity of the AMR-CT was also included in this file, but was incorrectly recorded due to a programming bug in the controlling Python script. Also due to a programming bug, the data in this matrix file was not fully recorded for the replicates in which the program was interrupted mid-replicate due to the occurrence of anomalies (see Sec. 4.3 and 5.1).
 - Text files recording the ROS console output for the programs spawned by the main ROS launch file and for the Python control script. The text file output was obtained by combining the commands in Refs. [145, 146].
 - A rosbag file, which contains all data published to any ROS topics during the run [147].
 - OTS rigid body tracking data, as shown in Fig. 32, including:
 - The AMR-CT frame, with an OTS marker configuration identical to that shown in Fig. 22.
 - The manipulator base with an OTS marker configuration identical to that of 23 albeit with slightly different marker labeling as shown in Fig. 35.
 - The manipulator EOAT with an OTS marker configuration show in Fig. 33.
 - The CMMA with an OTS marker configuration as shown in Fig. 34.
- The OTS tracking data was captured to provide ground truth validation against the other logged data. It should be noted that tracking data was stored in both CSV and the proprietary OTS tracking file format as described in Sec. 2.5.
- A spreadsheet of experimental notes containing the following for each replicate and each run:
 - A sequence of random numbers used to generate the run sequence (see Sec. 4.3).
 - The run number (i.e., 1-8).
 - Search method used (i.e., spiral search or UKF).
 - Measured RMMA side (i.e., side 1 or side 2).

- The AMR-CT base speed (i.e., 0.01 m/s or 0.025 m/s).
- The start time for each run based on the wall clock time.
- Description of anomalies that occurred during the run, if any.
- Video data of the experiment was recorded and live-streamed for viewing by NIST and ASTM Committee F45. The video capture included a four-way windowed split featuring:
 - A zoomed-out view of the experimental area.
 - A close-up view of the mobile manipulator.
 - A feed of the OTS computer screen.
 - A feed of the laptop controller screen.

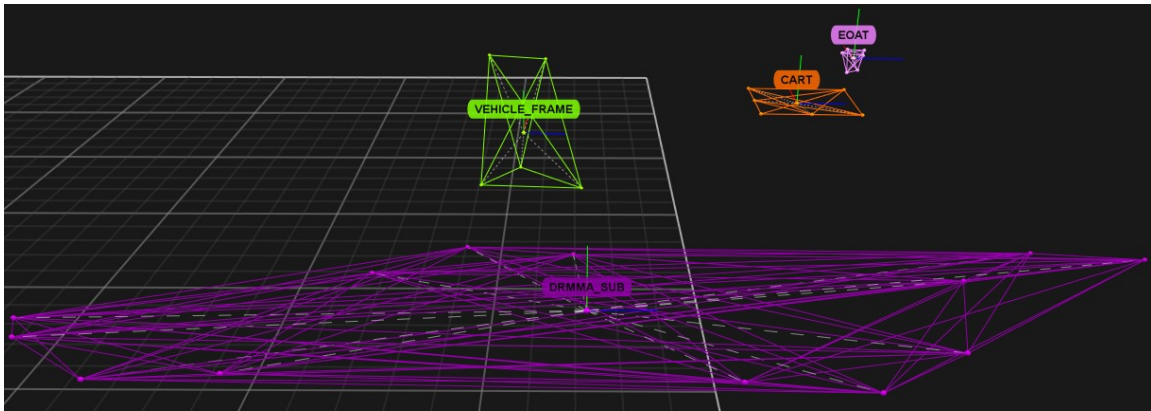


Fig. 32. Screenshot of OTS rigid bodies tracked during the experiment for ground truth comparison.

4.3. Procedure

The day prior to the experiment, the following steps were performed. First, the NIST manipulator and AMR-CT batteries were fully charged. The OTS was again re-calibrated as per manufacturer specification. The calibration summary generated by the proprietary software had an observed mean ray error of 0.642 mm and mean wand error of 0.110 mm that was labeled by the software as “exceptional”. Following this, the OTS was used to capture new ground truth fiducial position data, similar to that taken in Sec. 3.3, for potential future use (not used in this experiment). As video data of the experiment was captured, wall clocks, with radio synchronization capabilities to NIST timeservers, were placed on the CMMA to assist with later approximate data correspondence and note-taking. Synchronization of these clocks was attempted, but poor radio signal reception prevented proper synchronization, and the clocks were instead synchronized manually. Finally, the run order, which was randomized with respect to the factor level combinations, was generated

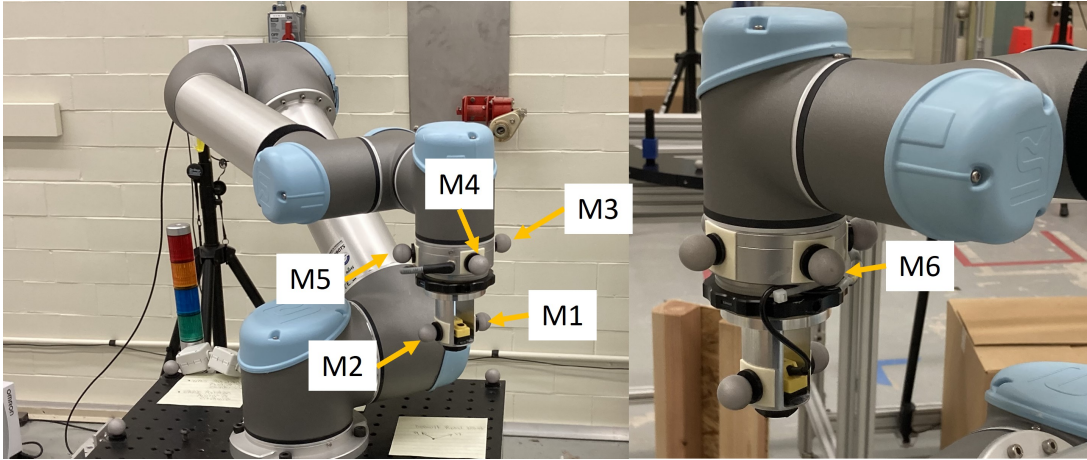


Fig. 33. The EOAT with six 19 mm diameter OTS markers attached to form a rigid body.

using a random number generator in spreadsheet software [104]. These random run order numbers were entered into the Python control script, which indexed into a stored dictionary of the possible factor combinations. The formatting of the note sheet, as outlined in Sec. 4.2 was also generated in advance.

The main experimental procedure for each replicate was as follows. The AMR-CT and manipulator were powered on and placed into their initial poses. For the former, this was the “LD_Start” goal shown in Fig. 4. The quality of each of the tracked OTS rigid bodies was checked and the correct run order in the Python script was verified. The time synchronization of the control laptop (both the host OS and Linux subsystem time) and the OTS computer was verified and recorded via screenshot as per Sec. 2.6. For the OTS computer and the host OS of the control laptop, multiple measurements were recorded via the command line output. For replicate 1, around 10 measurements were taken. For the remaining replicates, including prior to eighth run of replicate 6, around 5 measurements were recorded. Since the website at Ref. [98] reported only a single value, only one measurement was recorded for verifying the time synchronization of the control laptop Linux subsystem. Time synchronization of the AMR-CT was attempted similarly as per Sec. 2.6, but, as discussed in Sec. 2.6, what was later discovered to be a firewall misconfiguration prevented proper time synchronization. This did not significantly impact the experimental data alignment, as the ROS time of the control laptop was favored over the AMR-CT time for the experiment[144]. The control program was then started, along with rosbag recording, which automatically cycled through the randomized factor combinations for each run in the replicate. The AMR-CT then proceeded to randomly navigate to one of two goals placed on either side of the CMMA. Upon reaching the goal, the manipulator. The AMR-CT then began to approach the fiducial, first traveling about 250 mm towards the fiducial before halting and correcting the heading if needed. This and all other heading corrections were implemented as described as per Sec. 2.2. After this, the manipulator moved to intercept the first fiducial as the AMR-CT continued the approach. Once the first fiducial was



Fig. 34. The CMMA with 13 19 mm diameter OTS markers attached to form a rigid body.

in range, the AMR-CT was programmed to stop and the arm to begin searching for the first fiducial. This stationary search for the first fiducial was needed to acquire the error needed to initialize the UKF, when used [10]. Assuming the first fiducial was found, the AMR-CT continued to travel along the CMMA as the manipulator used the randomly selected search method in attempt to find the remaining fiducials, when in range. Once the final fiducial had been found or was no longer in range, the manipulator stowed and the AMR-CT returned to the “LD_Start” goal. The mobile manipulator would then be ready to start the next run using a different, randomly selected factor combination. During this process, one operator monitored the control programs and took notes on anomalies that occurred, one operator manually started and stopped the OTS recordings at the beginning and end of each run, respectively, and one operator controlled the video recordings.

The design of the experimental procedure accounted for the occurrence of possible anomalies. Anomaly 1 was if the manipulator were to fail to find the first fiducial, which would cause the rest of the run to be aborted and the remaining fiducials skipped. Anomaly 1 was considered a valid measurement where the interception rate was zero and the average search time equal to the max time spent searching for the first fiducial. Anomaly 2 occurred if the AMR-CT were to stop or a motion command fail due to obstacle detection. Prior to the experiment, it was determined that of Anomaly 2 occurred after a run, then the run would still

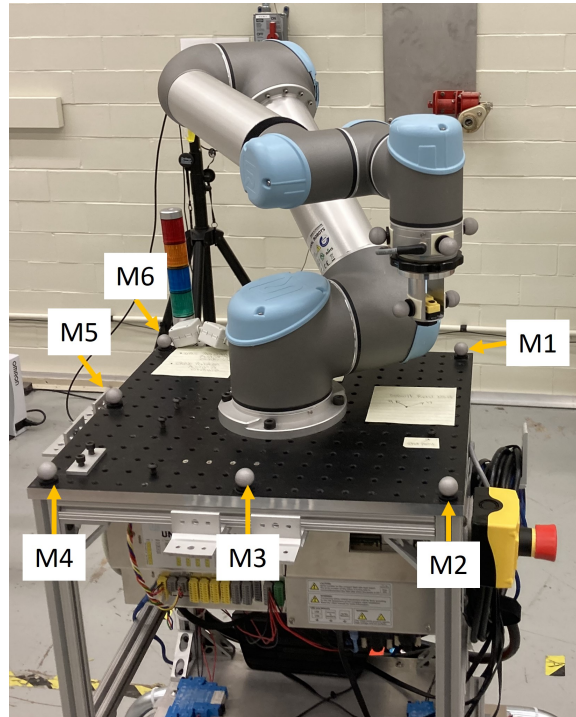


Fig. 35. Cart base rigid body configuration. Note that the configuration is identical to Fig. 23 except for the marker labeling order.

be considered valid. However, if this occurred during a run, the program would be stopped, an operator would manually free the vehicle and return it to the “LD_Start” goal, and restart the program with the factor combination that was being tested when the anomaly occurred. Anomaly 3 was if the manipulator batteries died during a run. If Anomaly 3 occurred, then all running programs would be halted, the data up to that point in the run saved, the mobile manipulator position again reset, the batteries re-charged, and the program re-started using the factor combination that was under test at the time the batteries died. Finally, Anomaly 4 corresponded to unknown error that could intermittently occur where the manipulator joint state publisher would deadlock, which caused the manipulator trajectory planner and/or the ROS driver to fail. If Anomaly 4 occurred, then the run would simply be restarted in the same fashion as described earlier in this paragraph, and the data captured from the run in which the error occurred would be considered incomplete and thus invalid. Note that if Anomaly 1 or Anomaly 4 occurred more than once within a replicate, then the rest of the experiment would have been aborted the underlying cause could be remediated. Since the error did not occur more than once within a replicate, the experiment proceeded to completion.

5. Analysis and Results

The raw response data and descriptive statistics (per factor combination) obtained from the experiment are presented in Table 4 and 5 and Table 6 and 7, respectively. The following subsections detail the analysis conducted on this data.

5.1. Anomalies

Of the total 48 runs in the experiment, anomalies, as described per Sec. 4.3 (i.e., Anomalies 1-4), 11 occurred during the experiment. Anomaly 1 occurred five times total, which included during replicate 1 (run 1), replicate 2 (run 4), replicate 5 (runs 6 and 7), and replicate 6 (run 8). As per Sec. 4.3, all of these runs were considered legitimate observations and were not discarded. Notably, all of the runs where Anomaly 1 occurred were testing the UKF search. Furthermore, the anomaly occurred on both sides of the CMMA and at both the high and low AMR-CT speeds. Occurrence of this anomaly can be mitigated in the future with several changes. First, UKF hyper-parameters can be further tuned, specifically by increasing the sample gap and initial search covariance. These changes impact the UKF search at the first fiducial by balancing search point density with physical search area coverage, albeit at the expense of search time. Additionally, failure to find the first fiducial seemed to occur when the AMR-CT would stop either too early or too late resulting in a larger difference in x position between the manipulator base and fiducial target. This behavior could have been caused by AMR-CT localization error, which can be calibrated out using an additional manipulator registration (e.g., using the RLS to detect a piece of retro-reflective tape placed on the CMMA at a pre-known distance away from the first fiducial).

Anomaly 2 occurred four times in total: during replicate 1 (run 5 and 6), replicate 3 (run 6), and replicate 5 (run 4). All occurrences of Anomaly 2 happened as the AMR-CT was returning to the initial pose (i.e., the “LD.Start” goal) after all fiducial searches were attempted, therefore all of the corresponding observations for these runs were considered legitimate. Upon reviewing the video footage, it was determined that the AMR-CT would occasionally get stuck within the forbidden zone placed near the “DynamicRMMA2” goal, which caused navigation to fail. This behavior can be corrected in the future by removing the forbidden zones from the map and instead modifying a parameter in the AMR-CT path planning configuration to increase the distance between the AMR-CT base and any obstacles encountered along the planned path.

Anomaly 3 did not occur during the experiment. The changes to the experimental procedure (i.e., recharging the batteries between replicates) and the longevity of the replaced manipulator cart batteries provided uninterrupted mobile manipulator operations throughout each replicate.

Anomaly 4 occurred three times in total: during replicate 3 (run 3), replicate 5 (run 1), and replicate 6 (run 8). Since Anomaly 4 occurred only once with a given replicate, the run was

Replicate	Run	Search Algorithm	CMMA Side	Speed (m/s)	Interception Rate	Search Time (s)	Total Marker Skipped
1	1	UKF	2	0.01	0	177.337	5
1	2	Spiral	2	0.01	1	10.030	0
1	3	UKF	1	0.01	0.833	10.688	0
1	4	UKF	2	0.025	0.667	8.427	0
1	5	Spiral	1	0.01	1	3.487	0
1	6	Spiral	1	0.025	0.833	6.733	0
1	7	UKF	1	0.025	0.5	7.845	0
1	8	Spiral	2	0.025	0.833	9.516	0
2	1	UKF	2	0.025	0.5	8.480	0
2	2	UKF	1	0.025	0.333	7.697	0
2	3	Spiral	1	0.025	0.833	3.547	0
2	4	UKF	1	0.01	0	175.774	5
2	5	UKF	2	0.01	0.667	11.746	0
2	6	Spiral	1	0.01	0.833	6.599	0
2	7	Spiral	2	0.01	0.833	9.705	0
2	8	Spiral	2	0.025	0.667	7.829	0
3	1	UKF	2	0.025	0.667	8.501	0
3	2	Spiral	1	0.025	0.833	12.177	0
3	3	Spiral	2	0.025	0.5	9.136	0
3	4	UKF	1	0.025	0.5	7.153	0
3	5	UKF	1	0.01	0.5	16.171	0
3	6	Spiral	1	0.01	0.667	17.274	0
3	7	Spiral	2	0.01	1	7.535	0
3	8	UKF	2	0.01	0.833	12.593	0

Table 4. Raw recorded response for each of the runs and replicates of the experiment (1 of 2).

Replicate	Run	Search Algorithm	CMMA Side	Speed (m/s)	Interception Rate	Search Time (s)	Total Marker Skipped
4	1	UKF	1	0.025	0.5	7.164	0
4	2	Spiral	1	0.01	1	3.876	0
4	3	UKF	2	0.025	0.5	7.795	0
4	4	Spiral	2	0.025	0.833	7.725	0
4	5	Spiral	1	0.025	1	3.088	0
4	6	Spiral	2	0.01	1	6.187	0
4	7	UKF	2	0.01	0.833	11.987	0
4	8	UKF	1	0.01	0.667	12.964	0
5	1	Spiral	1	0.01	0.833	8.496	0
5	2	Spiral	2	0.025	1	7.577	0
5	3	UKF	2	0.01	1	8.675	0
5	4	Spiral	1	0.025	0.667	8.078	0
5	5	Spiral	2	0.01	1	4.258	0
5	6	UKF	2	0.025	0	165.437	5
5	7	UKF	1	0.01	0	174.037	5
5	8	UKF	1	0.025	0.333	7.781	0
6	1	Spiral	2	0.01	1	10.494	0
6	2	Spiral	1	0.01	0.833	8.892	0
6	3	Spiral	1	0.025	0.5	9.386	0
6	4	UKF	1	0.025	0.333	8.715	1
6	5	Spiral	2	0.025	0.167	14.130	0
6	6	UKF	1	0.01	0.667	11.481	0
6	7	UKF	2	0.025	0.333	7.835	1
6	8	UKF	2	0.01	0	174.470	5

Table 5. Raw recorded response for each of the runs and replicates of the experiment (2 of 2).

Table 6. Descriptive Statistics for Fiducial Interception Rate By Factor Combination

Search Algorithm	CMMA Side	Speed (m/s)	Mean	Stdev	N
UKF	1	0.01	0.444	0.360	6
		0.025	0.417	0.091	6
	2	0.01	0.556	0.443	6
		0.025	0.444	0.251	6
Spiral	1	0.01	0.861	0.125	6
		0.025	0.778	0.172	6
	2	0.01	0.972	0.068	6
		0.025	0.667	0.298	6

Table 7. Descriptive Statistics for Fiducial Search Time (s) By Factor Combination

Search Algorithm	CMMA Side	Speed (m/s)	Mean (s)	Stdev (s)	N
UKF	1	0.01	68.853	83.720	6
		0.025	7.726	0.573	6
	2	0.01	66.135	85.042	6
		0.025	34.413	64.189	6
Spiral	1	0.01	8.104	5.027	6
		0.025	7.168	3.487	6
	2	0.01	8.035	2.480	6
		0.025	9.319	2.491	6

considered invalid and restarted as per Sec. 4.3.

In addition to the anomalies described in Sec. 4.2, the following human operator anomalies, which did not impact the data quality, occurred. For replicate 2 (run 1) and replication 6 (run 1) the operator accidentally started the code using the replicate 1 factor combination run-order. This was corrected during the experiment without issue. During replicate 2 (run 4), the search method in the video recording was mislabeled as showing the spiral search, when in fact the UKF was being tested. During the first run of replicate 4, the factor combination for run 6 was accidentally used due to operator error. The operator let this test run to completion as an extra experimental run before restoring the correct factor combination. During replicate 5 run 8, the spreadsheet notes were initially mislabeled with the incorrect factor combination, however, this was later corrected.

5.2. Time Synchronization Offset

As discussed in Sec. 2.6 and Sec. 4.3, client time synchronization for the OTS computer and the control laptop (both via the host OS and the Linux subsystem) was verified prior to each experimental run. In reporting the results of the offset between the synced client and remote time server clocks, the following was done. For the OTS PC and the host OS of the control laptop, the absolute value for each of the five or 10 measurements of the time

offset was computed. The mean of the 5 or 10 measurements was then computed. For the Linux subsystem of the control laptop, the absolute value of the single measurement was reported. These offsets are displayed in Table 8 and plotted in the bar graphs of Fig. 36. From the plot, it was observed that, prior to each replicate, the OTS computer was synced to the remote time server within 1.4 ms. With the exception of replicate 6 (run 8), lower time offsets were observed for the Linux subsystem of the control laptop when compared to the host OS. The former was synced to the remote time server within the 31 ms while the latter was synced within 35 ms.

Table 8. Time synchronization offset of clients prior to each replicate.

Replicate	Client	Measurements	Offset ^a (ms)
1	OTS PC	10	1.0388
1	Control Laptop (Host OS)	10	31.3272
1	Control Laptop (Linux Subsystem)	1	4
2	OTS PC	5	0.5398
2	Control Laptop (Host OS)	5	30.680
2	Control Laptop (Linux Subsystem)	1	28
3	OTS PC	5	0.5996
3	Control Laptop (Host OS)	5	30.468
3	Control Laptop (Linux Subsystem)	1	29
4	OTS PC	5	0.5198
4	Control Laptop (Host OS)	5	33.3748
4	Control Laptop (Linux Subsystem)	1	31
5	OTS PC	5	1.006
5	Control Laptop (Host OS)	5	32.882
5	Control Laptop (Linux Subsystem)	1	30
6	OTS PC	5	1.382
6	Control Laptop (Host OS)	5	32.377
6	Control Laptop (Linux Subsystem)	1	27
6 (Run 7)	OTS PC	5	0.574
6 (Run 7)	Control Laptop (Host OS)	5	17.066
6 (Run 7)	Control Laptop (Linux Subsystem)	1	24

^a Averaged over multiple measurements where applicable.

5.3. Model Adequacy Checks

As detailed in Sec. 4, the experimental model of Eq. 4 assumes that $\epsilon_{abcd} \sim NID(0, \sigma^2)$ (i.e., the errors are assumed to be normally and independently distributed with equal variance). The independence assumption was deemed to be reasonably satisfied by randomizing the factor combination for each run in the experimental procedure (Sec. 4).

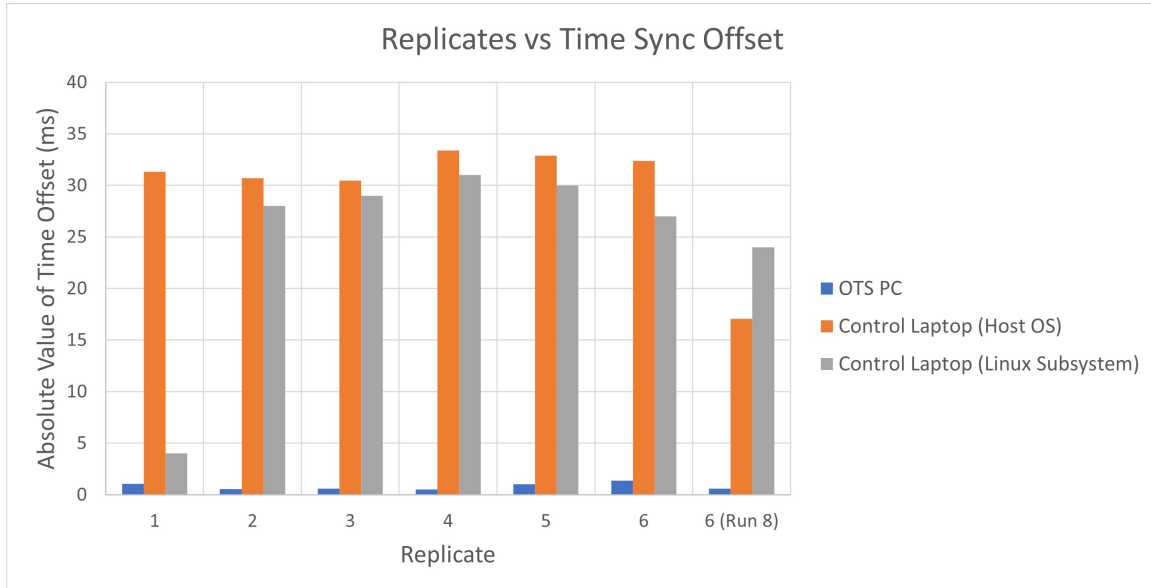


Fig. 36. Bar chart showing the time synchronization offset between the clients synchronized to the NIST time servers over NTP. The offsets were measured prior to each replicate.

5.3.1. Normality

The assumption of normality was assessed via graphical means using quantile-quantile (Q-Q) plots. The plots were generated using Jeffreys’s Amazing Statistics Program (JASP) (version 0.16.0.0) [148, 149]. Graphical analysis was selected over statistical tests, such as the Anderson-Darling test, since such tests often assume that the data under test are independent and residuals have been noted to not necessarily meet this assumption [139]. Furthermore, significance of such statistical tests can be influenced by the sample size [148]. The Q-Q plot corresponding to the interception rate data is given in Fig. 37. Note that, to generate the Q-Q plot, the interception rate data is first sorted in ascending order, the standardized residuals for each data point computed, then plotted against the quantiles (i.e., percentiles) of a theoretical distribution, which, in this case, is the normal distribution. If the errors of the interception rate data was approximately normal, then the standardized residuals of the interception rate data should have had an approximately linear relationship with the theoretical quantiles [148]. Such a linear relationship was observed in Fig. 37, though the slight snaking shape of the plot may suggest mild skewness of the data [148]. The Q-Q plot for the search time data is shown in Fig. 38. Unlike the Q-Q plot of Fig. 37, the standardized residuals of Fig. 38 show greater departure from this linear relationship. Note that the group of five standardized residuals of value greater than +2 corresponded to the runs where Anomaly 1 was observed (i.e., the first fiducial could not be found). Aside from these values, which as per Sec. 4.3 and 5.1 were not discarded, the remaining standardized residuals of Fig. 38 also show departure from linearity. Therefore, it was determined that, for the interception rate data, there was not significant evidence to suggest

that the errors of the experimental model were not normally distributed. However, for the search time data, it was determined that there was sufficient evidence to suggest that the data was not normally distributed.

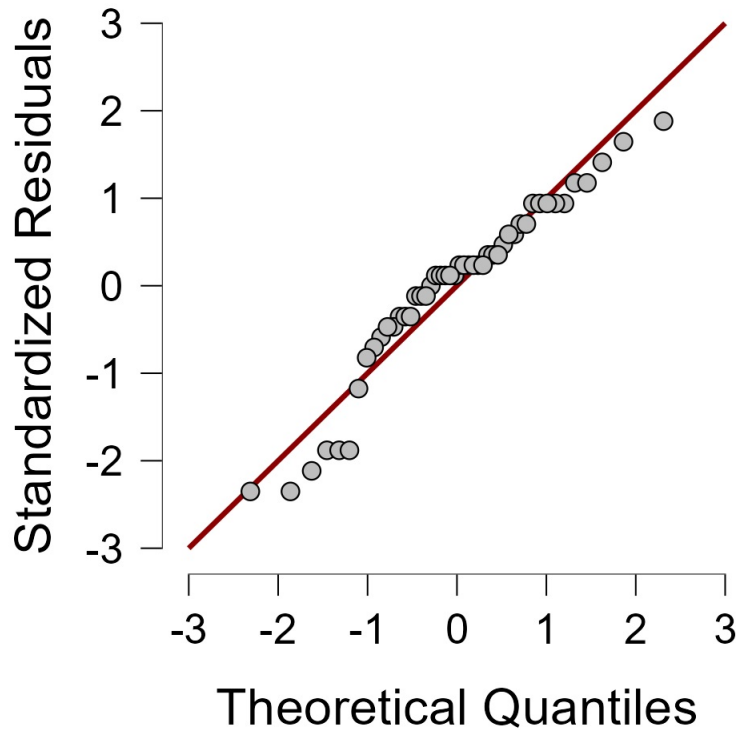


Fig. 37. Q-Q Plot for fiducial interception rate.

5.3.2. Equal Variance

For the test of equal variance, Levene's test was applied to the fiducial interception rates and search times using statistical software (version 0.16.0.0) [148, 149]. It should be noted that Levene's test is considered to be more robust when non-normality is suspected [150]. First, for the fiducial interception rates, the null hypothesis was that the samples of fiducial interception rates came from a population with equal variance. The alternative hypothesis was that the samples of fiducial interception rates did not come from a population with equal variance. The decision rule was based on the p-value computed from Levene's test: If the p-value was less than 0.05, the null hypothesis would be rejected; otherwise, the null hypothesis would not be rejected. The results for Levene's test on the sample of fiducial interception rates is presented in Table 9. Since the p-value was less than 0.05, the null hypothesis was rejected, and it was concluded that there was sufficient evidence to suggest that the samples of the fiducial interception rates do not come from a population with equal variance.

For the fiducial search times, the null hypothesis was that the samples of fiducial search

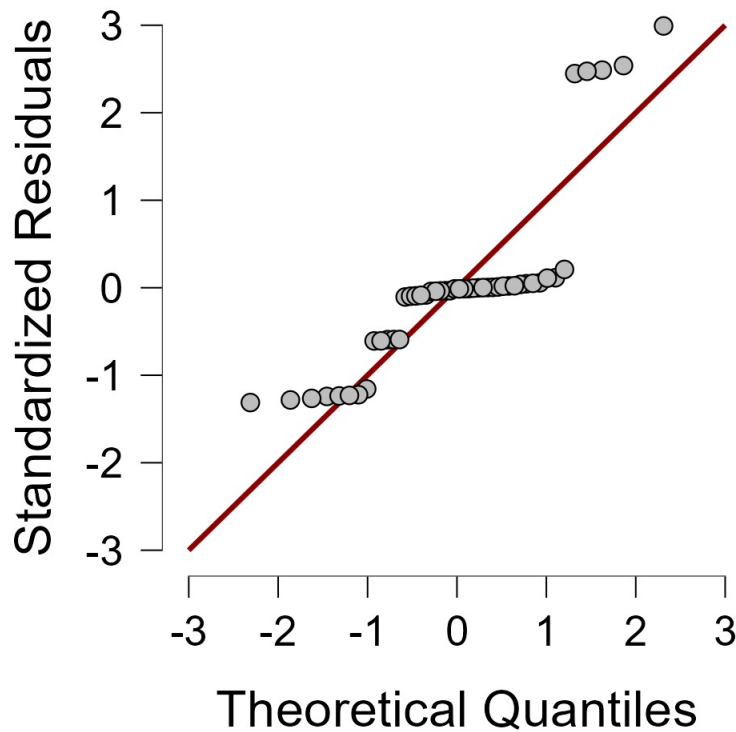


Fig. 38. Q-Q Plot for fiducial search times.

Table 9. Results for Equality of Variance Test (Levene's) on the samples of fiducial interception rates.

F	df1	df2	p
4.904	7.000	40.000	4.591e-4

times came from a population with equal variance. The alternative hypothesis was that the samples of fiducial search times did not come from a population with equal variance. The decision rule was based on the p-value computed from Levene's test: If the p-value was less than 0.05, the null hypothesis would be rejected; otherwise, the null hypothesis would not be rejected. The results for Levene's test on the sample of fiducial search times is presented in Table 10. Since the p-value was less than 0.05, the null hypothesis was rejected, and it was concluded that there was sufficient evidence to suggest that the samples of the fiducial search times do not come from a population with equal variance.

Table 10. Results for Equality of Variance Test (Levene's) on the samples of fiducial search times.

F	df1	df2	p
15.000	7.000	40.000	2.017e-9

5.4. Exploratory Data Analysis

Since the results of the model adequacy checks of Sec. 5.3 uncovered sufficient evidence to suggest that the assumption of normality and equal variance may not have been valid, the conclusions derived from conducting an ANOVA on the fiducial interception rates and fiducial search times may not be reliable¹⁶. Therefore, an Exploratory Data Analysis (EDA) on the sample data, statistics, and effects was instead used as the primary analysis method to determine which of the factors and interaction most impacted the response variables (i.e., the fiducial interception rate and average fiducial search time). Specifically, the first focus of the EDA was to produce a ranked list of factors to determine which of the main factors, including interactions, had the most impact on the response variables. Secondly, the objective was to determine the best settings for these factors. For the fiducial interception rate, optimization corresponded to maximization (i.e., a value of 1), while for the average fiducial search time, optimization corresponded to minimization (i.e., a value approaching 0 s). Note that the EDA approach has built-in redundancy whereby multiple techniques yielding the same conclusions increases the confidence in said conclusions [151].

To implement the EDA, another Python script, using the same language and library versions from Sec. 3.1.3 were used. Additionally, the following references were used in writing the code [107, 108, 110, 114, 152–164].

5.4.1. Alternate Variable Coding

Before proceeding to detail the results of individual plots, it should be noted that certain plots, namely from Sec. 5.4.4 - 5.4.6 required an alternate coding of the factor levels and interactions [165–167]. This coding is presented in Table 11 and was derived using the guidance from Ref. [165] as follows. First, the levels for each of the three main factors were mapped to the values ± 1 . Since the factors search algorithm and measured CMMA side were nominal variables, this mapping was arbitrarily chosen as per row two of Table 11. However, since the factor AMR-CT speed was an interval/ratio variable, the slower speed of 0.01 m/s was assigned to -1 and the faster speed of 0.025 m/s was assigned to +1. Since each factor had two levels and could be coded as such, the interactions follow a special case whereas +1 and -1 levels can be derived via cross product multiplication of the values for the corresponding main factor levels [165].

5.4.2. Ordered Data Plots

An ordered data plot was formed by plotting the raw response values of each run, in order from smallest to largest, against the corresponding settings for each of the main factors. This type of plot was used to 1) determine the best settings for each of the three main factors and 2) determine the most important main factor, based on the raw data. The ordered data

¹⁶For completeness, the results of the ANOVA are presented in Appendix B, however, readers are cautioned against adopting the conclusions derived from the ANOVA.

Table 11. Alternate Coding for Factor Combinations (with Interactions)

Search Algorithm	CMMA Side	Speed (m/s)	Search Algorithm * CMMA Side	Search Algorithm * Speed	CMMA Side * Speed	Search Algorithm * CMMA Side * Speed
-1 = UKF +1 = Spiral	-1 = Side 1 +1 = Side 2	-1 = 0.01 m/s +1 = 0.025 m/s				
-1	-1	-1	+1	+1	+1	-1
-1	-1	+1	+1	-1	-1	+1
-1	+1	-1	-1	+1	-1	+1
-1	+1	+1	-1	-1	+1	-1
+1	-1	-1	-1	-1	+1	+1
+1	-1	+1	-1	+1	-1	-1
+1	+1	-1	+1	-1	-1	-1
+1	+1	+1	+1	+1	+1	+1

plots for the fiducial interception rate and average fiducial search time are given in Fig. 39 and 40, respectively.

For the fiducial interception rate, Fig. 39 was interpreted as follows. Of the 48 total runs, 10 yielded the optimal response of 1. Of the 10 optimal runs, 90% used the spiral search, 80% used the slow AMR-CT speed of 0.01 m/s, and 70% were measured at side 2 of the CMMA. Out of the best half of the data, 79.17% used the spiral search, 66.67% used the slow AMR-CT speed, and 50% were measured at side 1 of the CMMA. Therefore, the best settings to optimize the fiducial interception rate, based on the raw data, corresponded to the spiral search algorithm, the slow AMR-CT speed, and side 2 of the CMMA. Overall, the search algorithm was considered the most important main factor impacting the fiducial interception rate [168].

For the average fiducial search time, Fig. 40 was interpreted as follows. Of the 48 total runs, one run (rep 4, run 5) resulted in the most optimal response of 4.258 s. This run used spiral search algorithm, the fast AMR-CT speed of 0.025 m/s, and was measured at side 1 of the CMMA. Of the nearest 8 most optimal runs, all used the spiral search algorithm, 62.50% used the slow AMR-CT speed, and 75% were measured at side 1 of the CMMA. Of the 5 least optimal runs (which corresponded to the 5 instances of Anomaly 1 described Sec. 5.1), all used the UKF search, 80% used the slow AMR-CT speed, and 60% used were measured at side 2 of the CMMA. Of the best half of the data, 58.33% used the spiral search, 70.83% used the fast AMR-CT base speed, and 54.17% were measured at side 1 of CMMA. Therefore, the best settings to optimize the average fiducial search time were the Spiral search algorithm, the fast AMR-CT speed, and side 1 of the CMMA. Interpretation of the most important main factor was less consistent. Since the 8 best runs used the spiral search and the 5 worst runs used the UKF search, search algorithm was again selected as the most important main factor. However, since the best half of the data most frequently used the fast AMR-CT speed, the AMR-CT speed may also be considered an important main factor.

5.4.3. DOE Scatter Plots

The design of experiments (DOE) scatter plot was formed by plotting the raw response value against each setting for each of the main factors. Additionally, the overall grand mean of the data is plotted to aid in detecting location shifts of the raw data across factor settings. The plot was used to 1) identify outliers, 2) determine the most important factor, and 3) determine the best settings for each factor, all with respect to the raw response data. The DOE scatter plots for the fiducial interception rate and average fiducial search time are given in Fig. 41 [169].

For the fiducial interception rate, Fig. 41 (top) was interpreted as follows. The five runs where Anomaly 1 occurred (see Sec. 5.1) stood out the most, which identified them as potential outliers. The outliers occurred across all settings of all the main factors except for the spiral search algorithm. As per Sec. 4.3, these were considered legitimate measure-

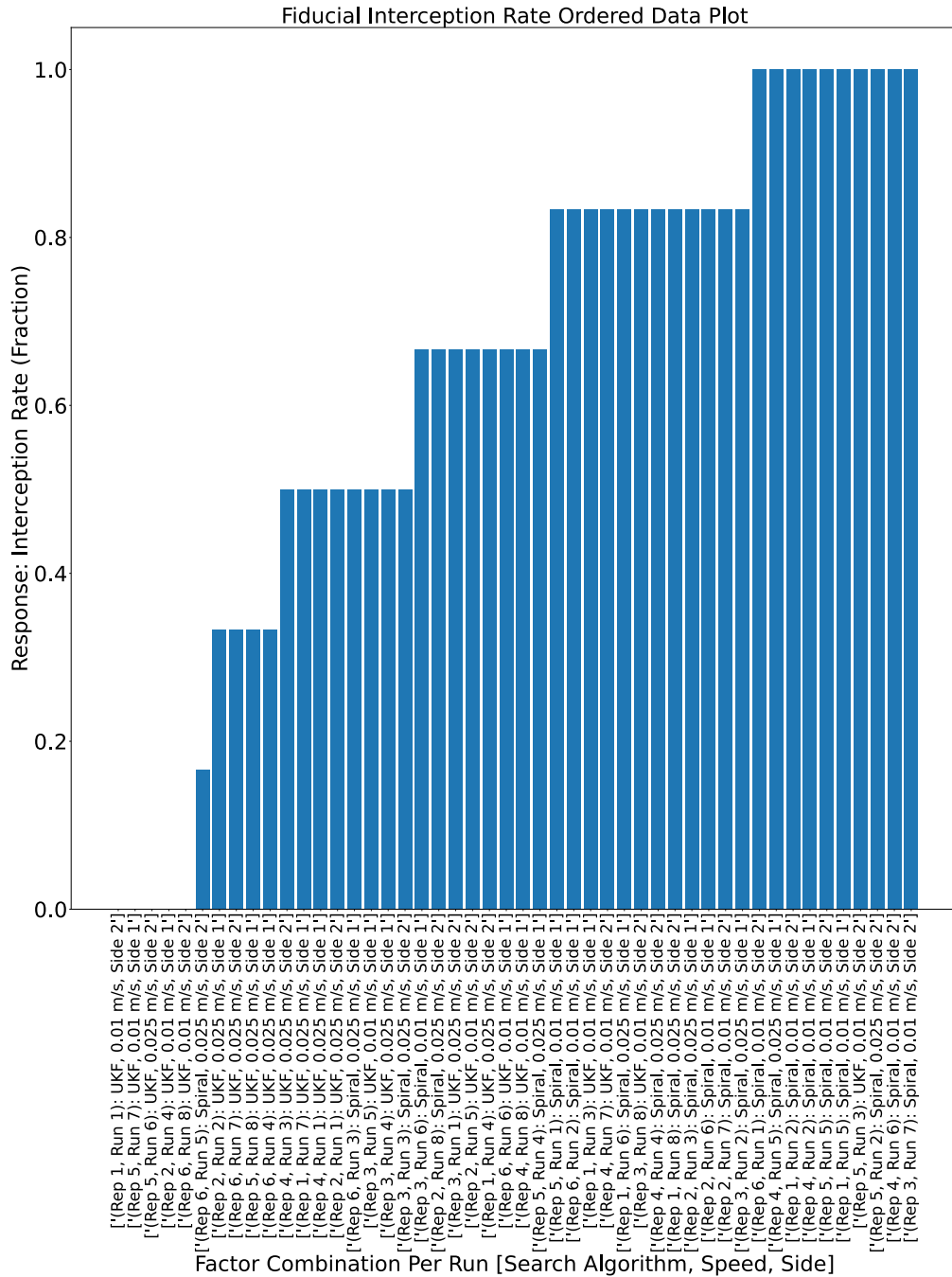


Fig. 39. Ordered data bar graph for the fiducial interception rate comparing the raw response against the factor combination used for each run.

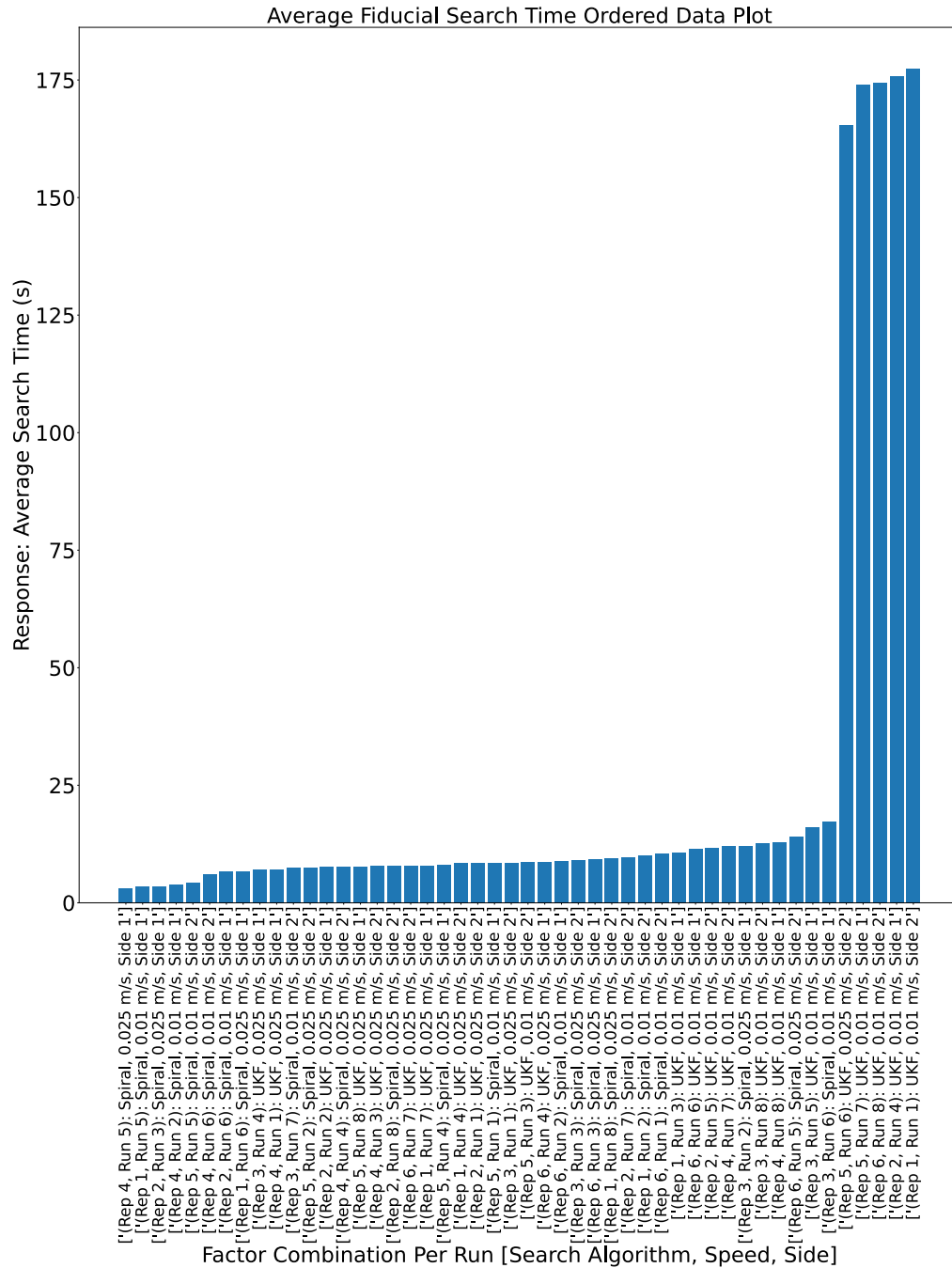


Fig. 40. Ordered data bar graph comparing the raw response (i.e., the average fiducial search time) against the factor combination used for each run.

ments. Considering the runs with Anomaly 1, the most important main factor impacting the fiducial interception rate was the search algorithm since this had the largest observed location shift in the data. The second most important main factor was indistinguishable between the remaining two factors. The best setting for the search algorithm was the spiral search, since the body of the data was closer to the optimal response value of 1. Similarly, the best settings for the AMR-CT speed and measured CMMA side were the slow speed of 0.01 m/s and side 1, respectively. Interestingly, no large difference in variation was detected amongst the settings any of the main factors.

For the average fiducial search time, Fig. 41 (bottom) was interpreted as follows. Again, the five runs where Anomaly 1 occurred stood out the most as outliers. The outliers again occurred across all settings of all the main factors except for the spiral search algorithm. Considering these outliers, the most important factor impacting the fiducial search time was the search algorithm, since this exhibited the largest location shift in the data. At a distant second was the AMR-CT speed, while the measured CMMA side exhibited too much data overlap for it to have been considered important. For the search algorithm, the spiral search performed best since the body of the data was closest to the optimal search time, (i.e., approaching zero). For the AMR-CT speed, the faster speed of 0.025 m/s was best. Finally, for the measured CMMA side, it could not be determined which setting performed best.

5.4.4. Interaction Effects Matrix Plots

The interaction effects matrix plot was used to produce 1) a list of factors, with two-factor interactions, ranked by importance to the response variable and 2) the best settings on average for each of the main factors. Note that, for the best settings, interactions are not considered because the experimenter has no direct control over interactions. This type of plot consists of multiple mean subplots arranged in an upper-right triangular formation. The diagonal subplots correspond to the three main effects while the off-diagonal plots correspond to the seven two-factor interactions. Each subplot relies on the alternate variable coding presented in Sec. 5.4.1 and shows the mean response between the high and low levels of each respective main factor or interaction. Each subplot also shows the grand mean as a horizontal line and a legend labeled with the effect size. The effect size is simply the difference in average response between the high and low levels, where larger differences indicate a more significant effect on the response. The interaction effects matrix plots for the fiducial interception rate and average fiducial search time are given in Fig. 42 and 43, respectively [165].

Therefore, based on Fig. 42, the list of factors that had the most impact on the fiducial interception rate were ranked based on which of the subplots had the lines with the steepest slope, which estimated the corresponding effect size. Specifically, the ranking was:

1. Search algorithm (estimated effect 0.354).

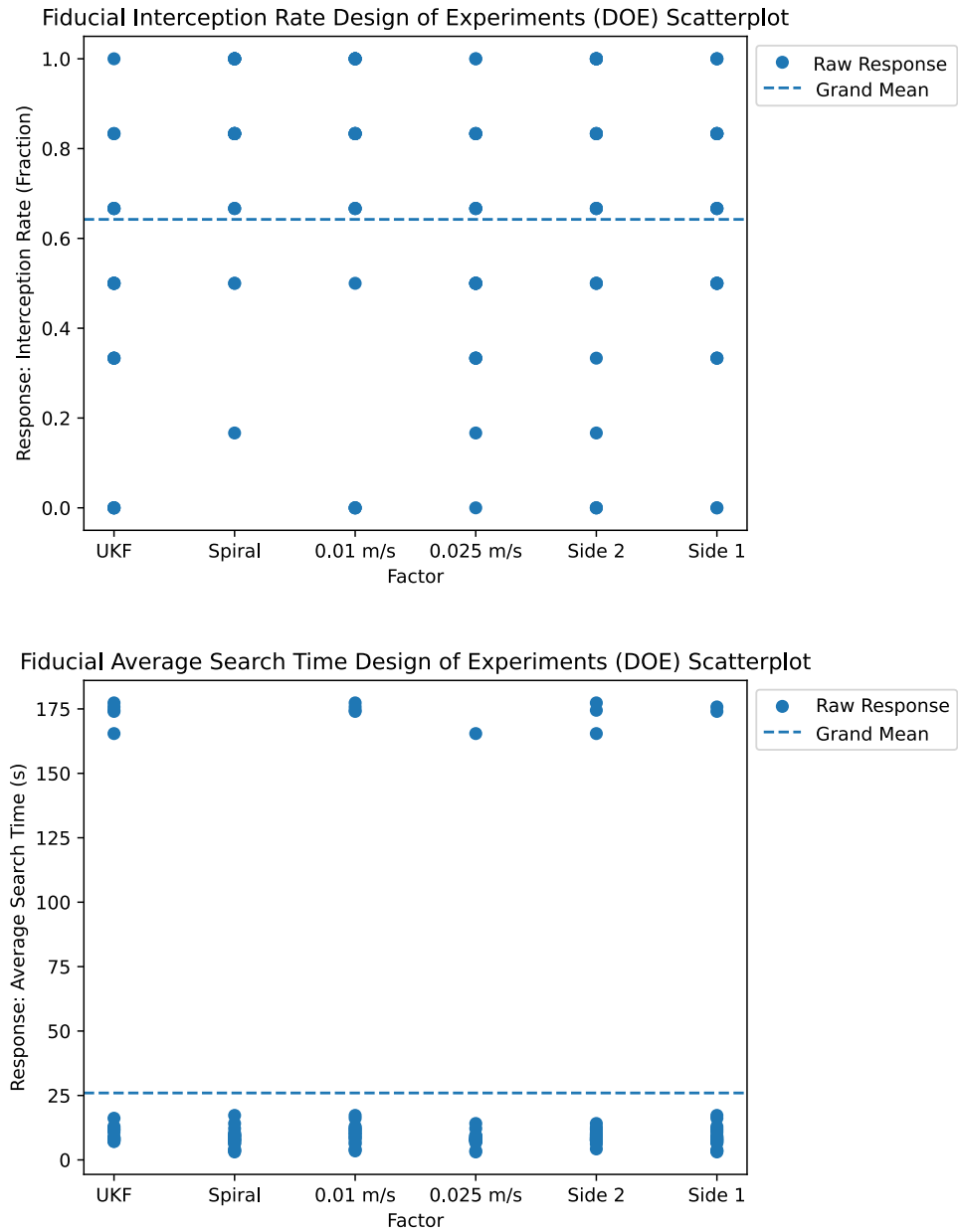


Fig. 41. DOE scatter plot comparing the raw fiducial interception rate (top) and average fiducial search time (bottom) across the three main factor levels. The center line plots the grand mean for the respective response variable.

2. AMR-CT speed (estimated effect: -0.132).
3. The interaction between measured CMMA side and AMR-CT speed (estimated effect: -0.076).
4. The interaction between search algorithm and AMR-CT speed (estimated effect: -0.062).
5. Measured CMMA side and the interaction between search algorithm and side (estimated effect ± 0.035).

For the three main effects, the best settings on average were the spiral search, the slow AMR-CT speed, and side 2 of the CMMA.

Similarly, for Fig. 43, the list of factors that had the most impact on the average fiducial interception rate were:

1. Search algorithm (estimated effect: -35.625).
2. The interaction between search algorithm and AMR-CT speed (estimated effect: 22.799).
3. AMR-CT speed (estimated effect: -22.625).
4. The interaction between measured CMMA side and AMR-CT speed (estimated effect: 7.406).
5. Measured CMMA side (estimated effect: 7.013).
6. The interaction between search algorithm and side (estimated effect: -5.972).

Additionally, for the three main effects, the best settings on average were the spiral search, the fast AMR-CT speed, and side 1 of the CMMA,

5.4.5. Block Plots

Block plots consist of three subplots corresponding to each of the three main factors of interest. Each subplot was formed by plotting the average response for each setting of a given main factor setting against all possible combinations of the other main factor settings (referred to as “robustness factors”). For each average response, the high and low setting of the factor of interest is labeled accordingly. The block plot was used to determine 1) the most important factors including interactions, and 2) the best settings for the factors. The block plots for the fiducial interception rate and average fiducial search time are given in Fig. 44 and 45, respectively [165]. Note that the length of each block is annotated.

The block plot for the fiducial interception rate in Fig. 44 was interpreted as follows. The important factors were determined by examining the consistency of the block heights and the arrangement of the high and low settings of a given factor [166]. First, the search algorithm was the only factor in which one setting (i.e., the spiral search) consistently

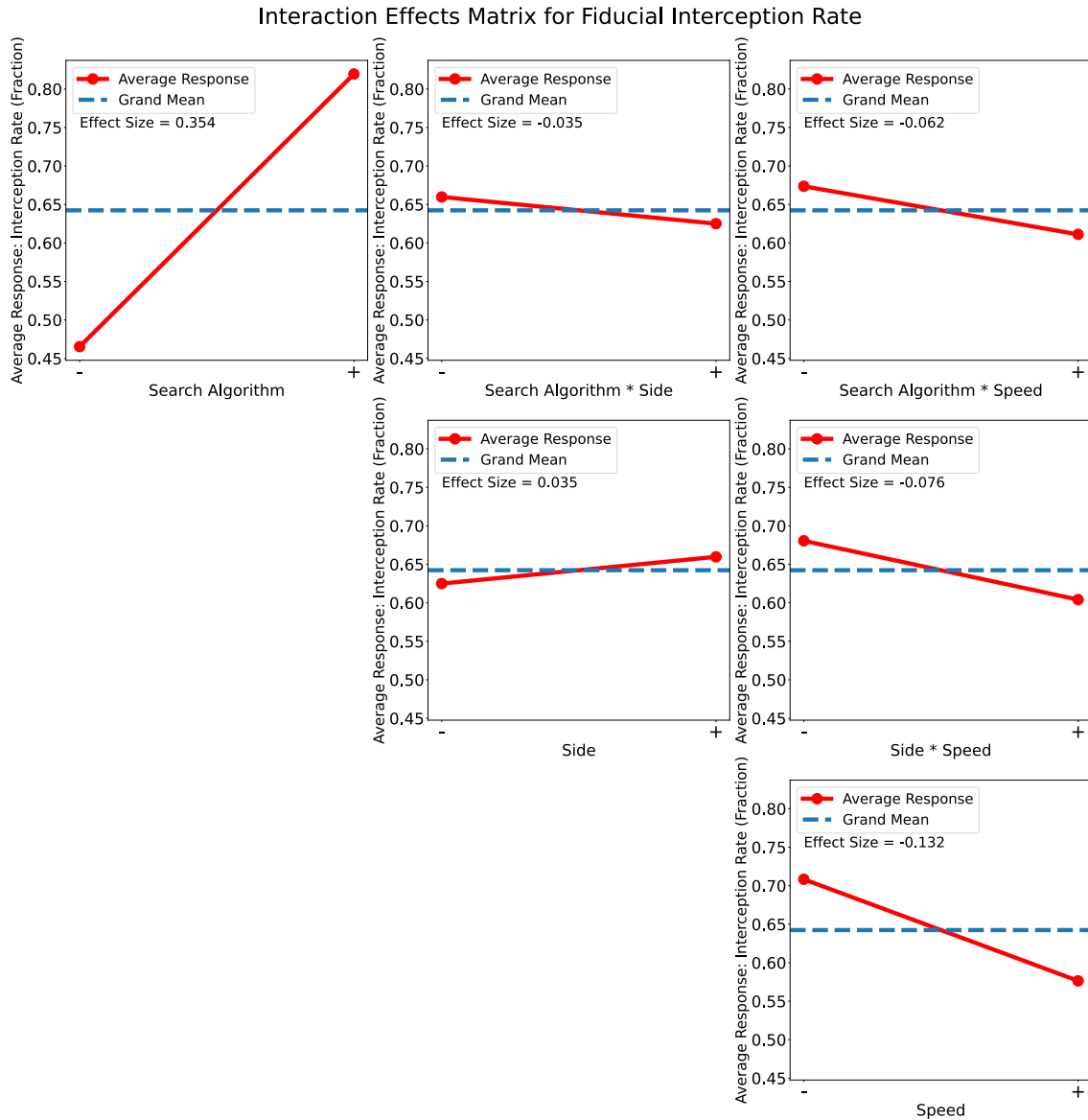


Fig. 42. Interaction effects matrix plot for the average response (i.e., the fiducial interception rate). Note that the on-diagonal subplots show the change in average response across factor levels for each of the three main factors while the off-diagonal plots show the change in average response for 2-factor interactions. The horizontal line of each plot shows the grand mean of the fiducial interception rates.

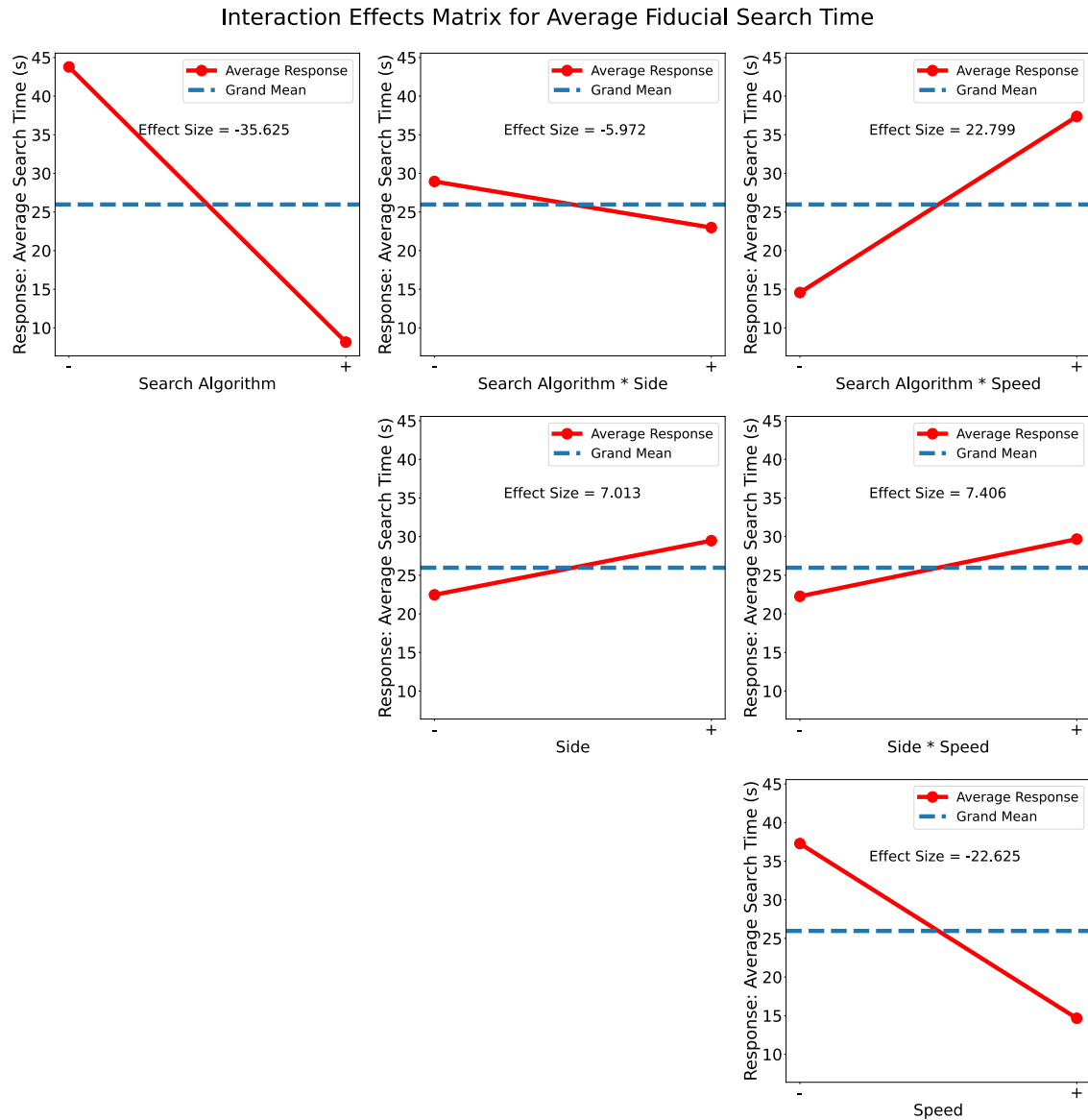


Fig. 43. Interaction effects matrix plot for the average response (i.e., the average search time). Note that the on-diagonal subplots show the change in average response across factor levels for each of the three main factor while the off-diagonal plots show the change in average response for 2-factor interactions. The horizontal line of each plot shows the grand mean of the average search times.

outperformed the other (i.e., the UKF search) across all settings of the other robustness factors. The blocks for the search algorithm were large, but inconsistently sized across the AMR-CT speed factor (i.e., the blocks corresponding to the slow speed were larger than the blocks for the fast speed), which suggested an interaction between search algorithm and AMR-CT speed was also important. Since the difference in block size across the settings of the AMR-CT speed was no more than 0.1950 and less than the estimated effect size of the search algorithm (0.3542), the effect of the interaction was considered to be less important than the effect of the search algorithm. The remaining block plots did not show consistency in either the setting that performed best nor the size of the block. In summary, the search algorithm and the interaction between search algorithm and speed were deemed important and the best setting for the search algorithm was the spiral search. The best settings for the AMR-CT speed and measured CMMA side could not easily be determined.

The block plot for the fiducial search time in Fig. 45 was interpreted as follows. Since none of the factor levels demonstrated consistently better performance over the other across all settings of the other robustness factors and none of the corresponding blocks were consistently large, important factors could not be identified. Therefore, the best settings for each of the main factors could also not be easily determined.

5.4.6. DOE Youden Plots

The DOE Youden plot was used to determine which of the factors, including interactions had the most impact on the response on average. Specifically, the plot was used to derive a ranked list of factors or interaction from most important to least important and to classify each factor or interaction as important or unimportant. The plots were formed as a scatter plot of the mean response for the high setting of each factor or interaction against the low setting for each factor or interaction. The location of the grand mean is included as vertical and horizontal lines. The DOE Youden plots for the fiducial interception rate and average fiducial search time are given in Fig. 46 [167].

The DOE Youden plot for the fiducial interception rate Fig. 46 (top) was interpreted as follows. The ranked list of factors was derived by examining which factors were furthest from the center of the plot. The factors that clustered towards the center of plot were deemed unimportant. Therefore, the ranked list of factors (with importance) was:

1. Search algorithm (important).
2. AMR-CT speed (important).
3. The interaction between measured CMMA side and AMR-CT speed (unimportant).
4. The interaction between search algorithm and AMR-CT speed (unimportant).
5. The interaction between search algorithm and measured CMMA side and the interaction between search algorithm, AMR-CT speed, and measured CMMA side, and measured CMMA side (unimportant).

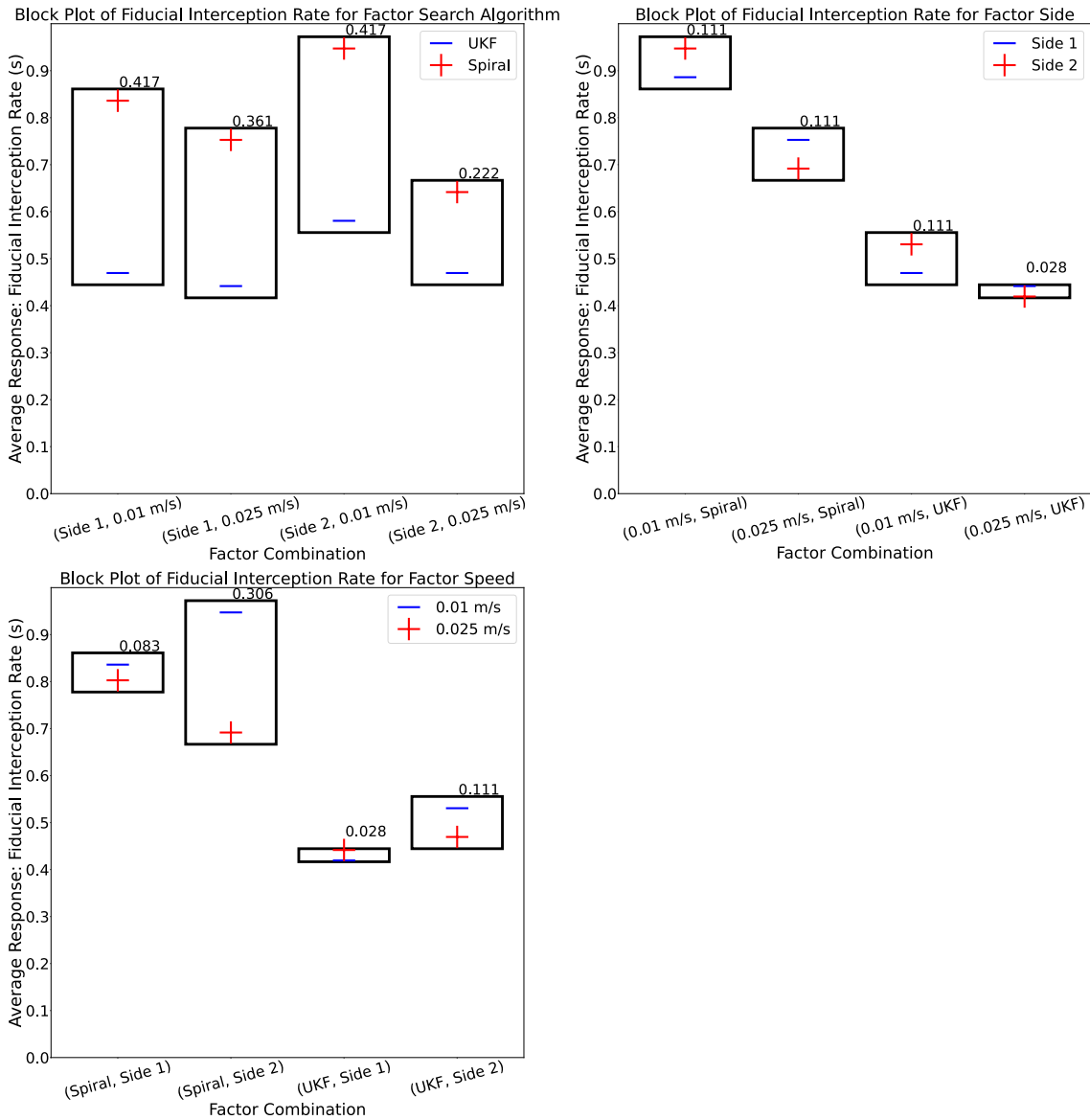


Fig. 44. Block plot for the fiducial interception rate. Note that each block is annotated with its respective length, as well as the corresponding high and low settings of the factor of interest.

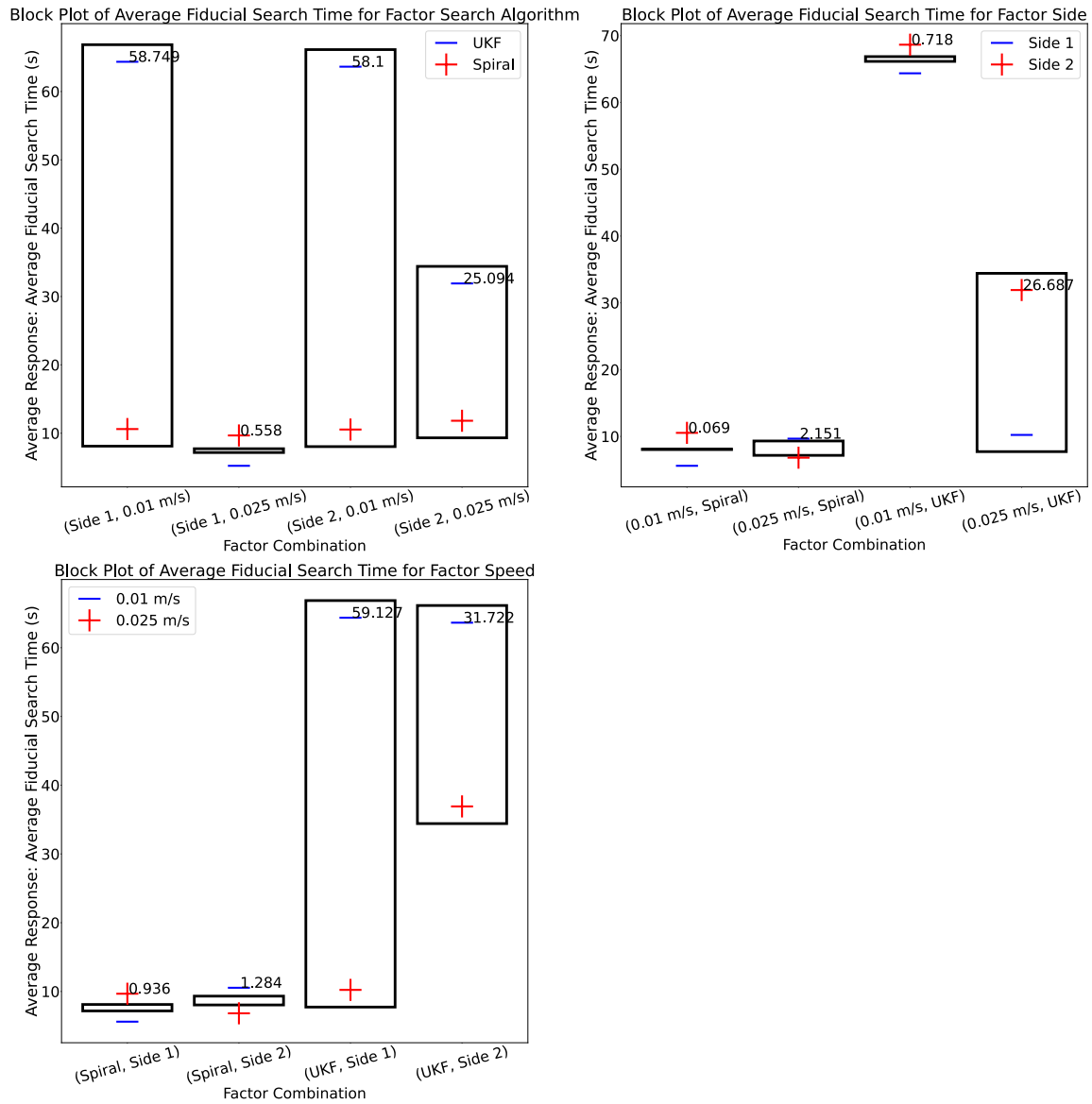


Fig. 45. Block plot for the average fiducial search time. Note that each block is annotated with its respective length, as well as the corresponding high and low settings of the factor of interest.

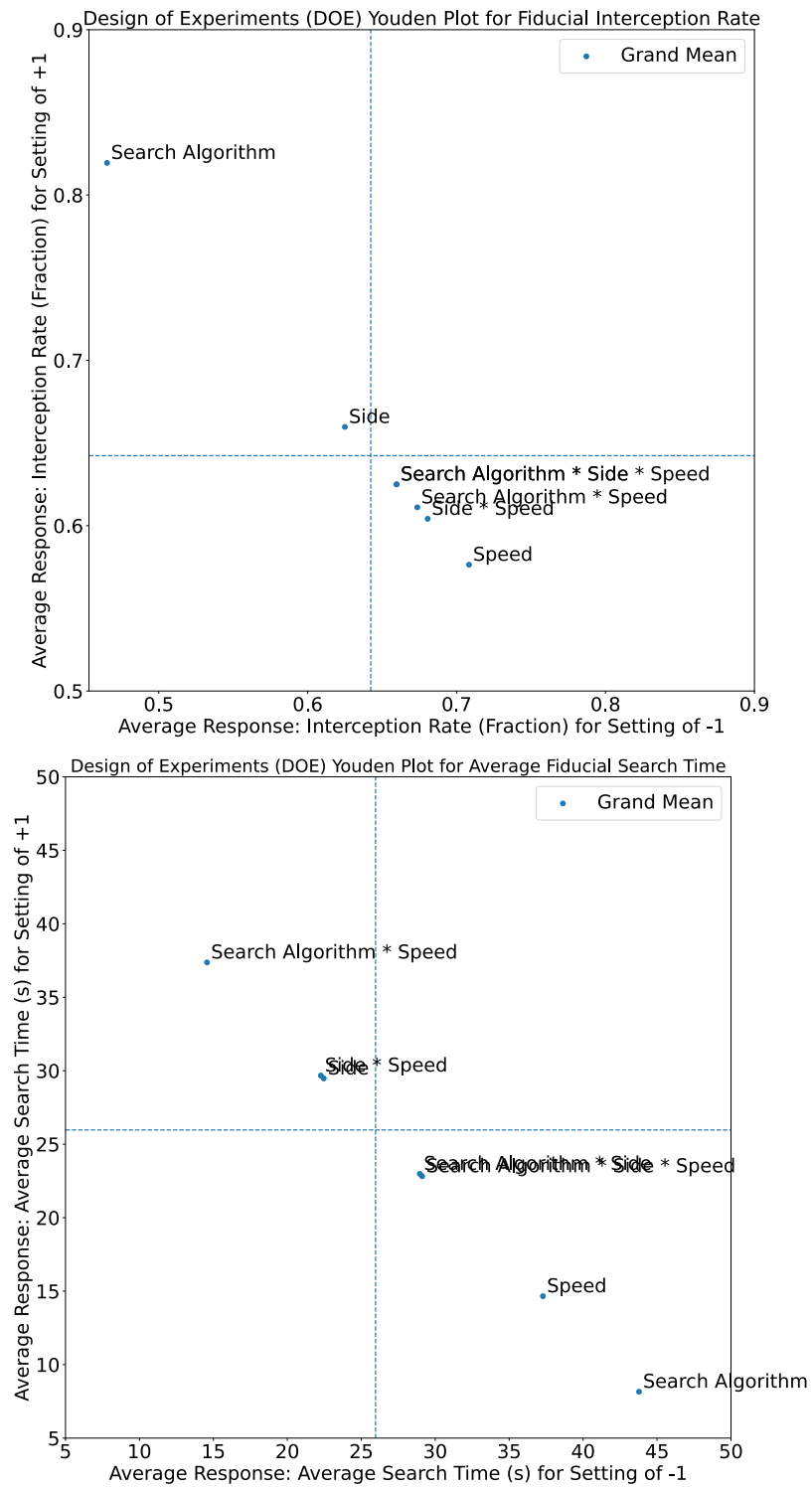


Fig. 46. The DOE Youden plot for the fiducial interception rate (top) and average fiducial search time (bottom).

Similarly, for the DOE Youden plot for the average fiducial search times (Fig. 46 (bottom)), the ranked list of factors was:

1. Search algorithm (important).
2. The interaction between search algorithm and AMR-CT speed (important) and AMR-CT speed (important).
3. The measured CMMA side and the interaction between measured CMMA side and AMR-CT speed (important).
4. The interaction between search algorithm, AMR-CT speed and measured CMMA side and the interaction between search algorithm and measured CMMA side (important).

All factors and interactions were deemed important to the average fiducial search time because they did not cluster towards the center of the Youden plot.

5.4.7. |Effects| Plots

Plots of |effects| were used to derive a list of factors, including interactions, ranked by importance. Additionally, the plot was used to identify the factors, including interactions, that were important by comparing the estimated effect size against an engineering significance value. The engineering significance (see Ref. [170]) was favored over statistical significance due to the results of the model adequacy check in Sec. 5.3. The plots were formed by plotting the ordered (i.e., largest to smallest) absolute value of the estimated effects for each of the main factors and interactions. Additionally, each plot includes a bifurcation line to determine important effects based on the selected engineering significance value for each response. The |effects| plots for the fiducial interception rate and average fiducial search time are given in Fig. 47. [171].

The |effects| plot for the fiducial interception rate (Fig. 47 (top)) was interpreted as follows. The engineering significance value, intuitively, was selected to correspond to a difference of one additional detected fiducial of six. Therefore, the bifurcation line was drawn at the fractional value of 0.167. From this, the search algorithm was considered the only important factor.

Alternatively, the |effects| plot for the average fiducial search time (Fig. 47 (bottom)) was interpreted as follows. The bifurcation line was instead drawn at a value of 1 s, which was slightly less than a rough engineering cutoff of 5% of grand mean of the average fiducial search times (i.e., $0.05 * 25.969 = 1.298$) [170]. From this, all of the factors and interactions were considered important and the ranking of factors was:

1. Search algorithm (important).
2. The interaction between search algorithm and AMR-CT speed and AMR-CT speed (important).

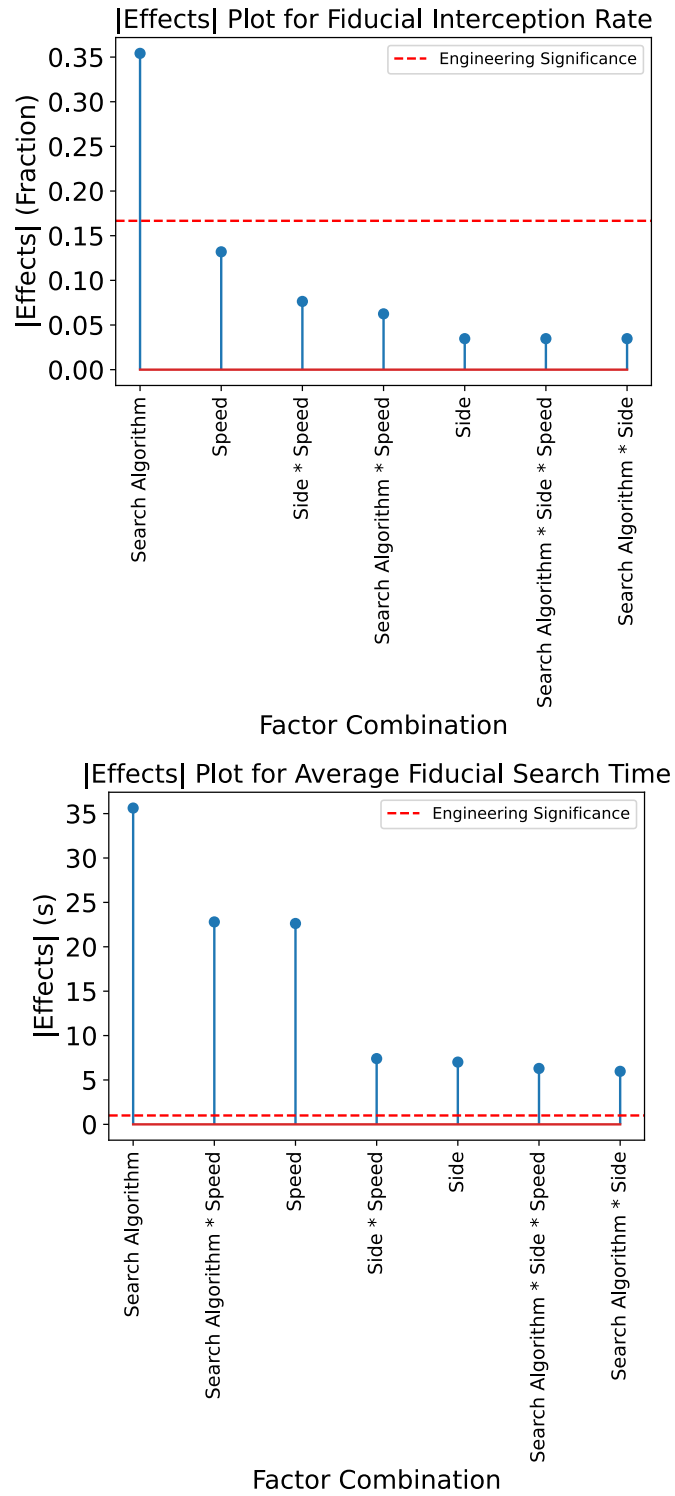


Fig. 47. |Effects| plots for the fiducial interception rate (top) and average fiducial search times (bottom).

3. Measured CMMA side and the interaction between measured CMMA side and AMR-CT speed.
4. The interaction between search algorithm, AMR-CT speed and measured CMMA side (important).
5. The interaction between search algorithm and measured CMMA side (important).

5.4.8. Half Normal Probability Plots

The half-normal probability plot was used to determine which factors, including interactions were important. It was formed by plotting the absolute values of the estimated effects, in order, against the theoretical quantiles of the half-normal probability distribution (i.e., the absolute value of a normally distributed random variable). To interpret the plots, the unimportant factors and interactions should have had estimated effects that were normally distributed at or near zero, whereas the estimated effects of important factors and interactions should not. Therefore, if all of the factors were unimportant, then the half-normal probability plot would show a linear relationship. If a factor was important, then it should have deviated from the line fitted to the other unimportant factors that appear near the origin of the plot. The half-normal probability plots for the fiducial interception rate and average fiducial search times are shown in Fig. 48 [172].

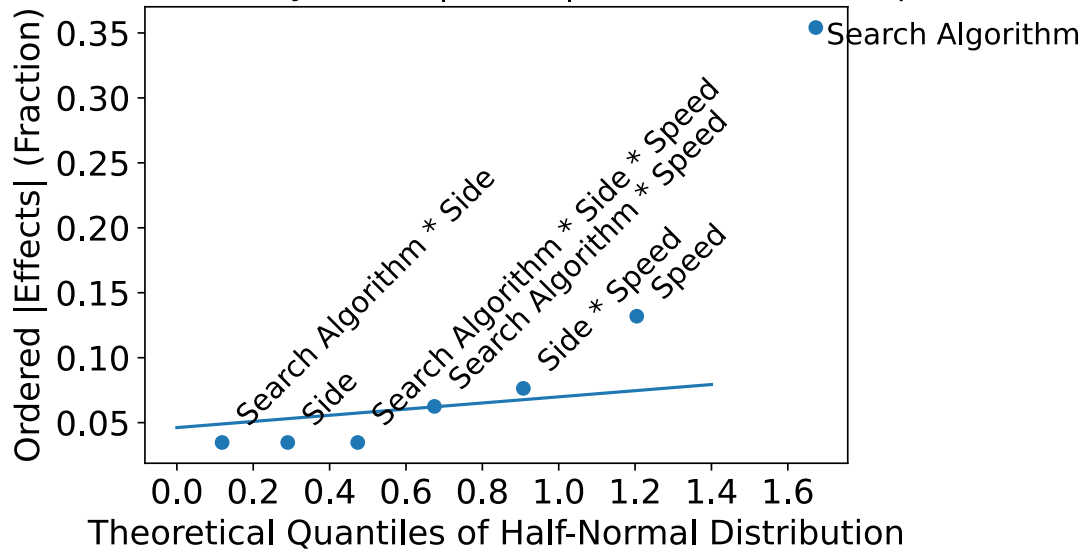
The half-normal probability plot for the fiducial interception rate (Fig. 48 (top)) was interpreted as follows. The factors, search algorithm and AMR-CT speed, substantially deviated from the line fitted to the other four near-zero factor and interaction estimated effects. Therefore, these two factors were deemed important and the remaining factors were deemed unimportant. The search algorithm was deemed more important than AMR-CT speed because the estimated effect for the former deviated more from the fitted line.

Similarly, the half-normal probability plot for the average fiducial search time (Fig. 48 (bottom)) was interpreted as follows. The estimated effects for the search algorithm, the interaction between search algorithm and AMR-CT speed, and the AMR-CT speed all deviated substantially from the line fitted to the other four near-zero factor and interaction estimated effects. Therefore, these three factors were deemed important and the remaining factors were deemed unimportant. The search algorithm effect was deemed more important than the effect of the AMR-CT speed and the interaction between the search algorithm and the AMR-CT speed because the estimated effect for the search algorithm deviated more from the fitted line.

5.4.9. Summary of Outcomes and Discussion

Summaries of the outcomes for each plot type of the EDA are presented for the fiducial interception rate and average fiducial search time in Table 12 and 13, respectively. For each plot type, where applicable, the important factors, with interactions, are provided along with the best settings for each of the main factors. For the best settings, a value of “N/A”

Half-Normal Probability Plot of |Effects| for Fiducial Interception Rate



Half-Normal Probability Plot of |Effects| for Average Fiducial Search Time

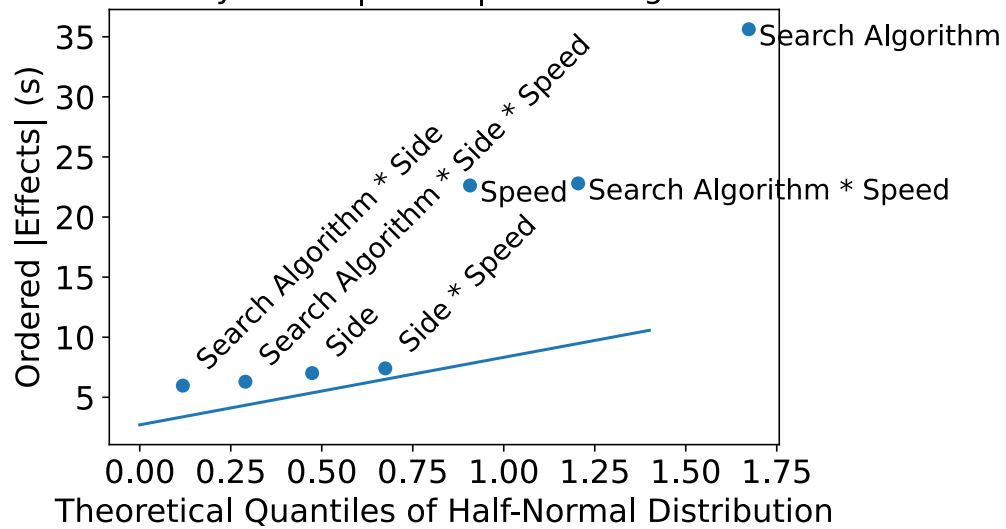


Fig. 48. Half normal probability plot for the fiducial interception rate (top) and average fiducial search time (bottom).

is used for plot types that are not designed to determine the best settings. Additionally, a value of “?” was used to indicate inconclusiveness when interpreting the best settings of a given plot.

First, the outcomes of the EDA for the fiducial interception rate are summarized as per Table 12. All plot types indicated that the search algorithm was the most important factor impacting the fiducial interception rate. Additionally, of the plot types that resulted in a ranking of important factors, the AMR-CT speed was consistently flagged as the second most important factor. For the plots that were used to interpret interactions, the interaction between AMR-CT speed and search algorithm and the interaction between AMR-CT speed and measured CMMA side were identified as important, although less so than the search algorithm alone. The measured CMMA side was generally considered to not be of importance. One possible explanation for this result was that the symmetric, identical fiducial layout between the two sides of the CMMA adequately controlled for differences that could be attributed to the measured CMMA side. Given the results in Ref. [10], testing different fiducial layouts for each CMMA side may yield different results. For example, a more significant interaction between search algorithm and measured CMMA side may be possible, given that the UKF hyper-parameters have been shown to be sensitive to the fiducial arrangement. The best settings for each factor were consistently determined to be the spiral search, the slow AMR-CT speed of 0.01 m/s, and side 2 of the CMMA.

Table 12. Summary of EDA Step Outcomes for the Fiducial Interception Rate

Plot Type	Important Factors and/or Ranking ^a	Best Settings ^b [Search Algorithm, AMR-CT Speed, CMMA Side]
Order Data	Search Algorithm	[Spiral, 0.01 m/s, Side 2]
DOE Scatter	Search Algorithm	[Spiral, 0.01 m/s, Side 2]
Interaction Effects Matrix	1. Search Algorithm 2. AMR-CT Speed 3. CMMA Side * AMR-CT Speed 4. Search Algorithm * AMR-CT Speed 5. CMMA Side 5. Search Algorithm * CMMA Side	[Spiral, 0.01 m/s, Side 2]
Block	Search Algorithm Search Algorithm * AMR-CT Speed	[Spiral Search, ?, ?]
DOE Youden	1. Search Algorithm 2. AMR-CT Speed	[N/A, N/A, N/A]
Effects	Search Algorithm	[N/A, N/A, N/A]
Half-normal Probability	1. Search Algorithm 2. AMR-CT Speed	[N/A, N/A, N/A]

^a Numbers indicate ranking where applicable; factors with same numbers have the same ranking of importance. ^b Value of “?” indicates inconclusiveness; value of “N/A” indicates plot not used to derive best settings.

Next, the outcomes of the EDA for the average fiducial search times are presented as per Table 12. First, it should be noted that the block plot was inconclusive with regards to the most important factor and the best settings. That aside, all plot types again flagged the search algorithm as being the most important factor impacting the fiducial search time. Of the plots that did not examine interactions, the AMR-CT speed was also identified as an important factor, albeit less so than the search algorithm. Of the plots that produced a ranking of important factors, the interaction between AMR-CT and search algorithm was considered more important than AMR-CT speed alone, but less important than the search algorithm alone. Note that, unlike the fiducial interception rate, the DOE Youden plot and |effects| plot flagged all of the factors and interactions as potentially important. The settings that resulted in the best average fiducial search time were the spiral search, the fast AMR-CT speed, and side 1 of the CMMA. Note that these best settings were consistent with the those for the fiducial interception rate, except for the best setting of the AMR-CT speed, which intuitively makes sense because a higher AMR-CT speed inherently results in a smaller time windows for searches from fiducial to fiducial. This result also reinforced the importance of evaluating both the fiducial interception rate and the fiducial search time

to assess continuous mobile manipulator performance.

Table 13. Summary of EDA Step Outcomes for the Average Fiducial Search Time

Plot Type	Important Factors and/or Ranking	Best Settings ^b [Search Algorithm, AMR-CT Speed, CMMA Side]
Order Data	Search Algorithm AMR-CT Speed	[Spiral, 0.025 m/s, Side 1]
DOE Scatter	Search Algorithm AMR-CT Speed	[Spiral, 0.025 m/s, ?]
Interaction Effects Matrix	1. Search Algorithm 2. Search Algorithm * AMR-CT Speed 3. AMR-CT Speed 4. CMMA Side * AMR-CT Speed 5. CMMA Side 5. Search Algorithm * CMMA Side	[Spiral, 0.025 m/s, Side 1]
Block	?	[?, ?, ?]
DOE Youden	1. Search Algorithm 2. Search Algorithm * AMR-CT Speed 2. AMR-CT Speed 3. CMMA Side 3. CMMA Side * AMR-CT Speed 4. Search Algorithm * CMMA Side * AMR-CT Speed 4. Search Algorithm * CMMA Side	[N/A, N/A, N/A]
Effects	1. Search Algorithm 2. Search Algorithm * AMR-CT Speed 2. AMR-CT Speed 3. CMMA Side 3. CMMA Side * AMR-CT Speed 4. Search Algorithm * CMMA Side * AMR-CT Speed 5. Search Algorithm * CMMA Side	[N/A, N/A, N/A]
Half-normal Probability	1. Search Algorithm 2. Search Algorithm * AMR-CT Speed 2. AMR-CT Speed	[N/A, N/A, N/A]

^a Numbers indicate ranking where applicable; factors with same numbers have the same ranking of importance. ^b Value of “?” indicates inconclusiveness; value of “N/A” indicates plot not used to derive best settings.

Several possibilities were considered to explain the better performance of the spiral search over the UKF search. The occurrences of Anomaly 3 could have, on average, degraded the performance of the UKF when compared to the spiral search. Applying the suggested mitigations for Anomaly 3 from Sec. 5.1 (i.e., using an additional registration aid, such as

reflective tape, creating alternate initial hyper-parameters for the search of the first fiducial) could eliminate these outliers to improve on-average performance. As tuning of the UKF hyper-parameters was limited and an alternate fiducial layout was used in comparison to Ref. [10], further tuning of the hyper-parameters could improve UKF performance.

As discussed in [10], performance could be inversely related to the inter-fiducial distance, due to accumulated uncertainty. The gradual motion drift accumulated by the frequent AMR-CT heading corrections, observable during replicate 3 (runs 3-5), shown in Fig. 49, for example, could have further exacerbated this problem and degraded performance of UKF. Note that, in Fig. 50, the drastic increase in search area and decrease in search density between fiducials 3 and 4 in comparison to between fiducials 4 and 5. Although fiducial 3 was detected for run 4, for run 5, both fiducials 3 and 4 were missed. Yet, the search area for fiducial 3 was drastically larger than for fiducial 4, suggesting increases in UKF co-variance that may not just be related to missing a fiducial. Further tuning of the hyper-parameters may mitigate such issues as well.

The plots in Fig. 49 and 50 also displayed gaps and/or more sudden shifts in AMR-CT position, which suggests AMR-CT communication performance and/or localization performance may have degraded performance. The manufacturer software for the AMR-CT allows adjustment of several localization parameters, including the frequency of re-localization computations (specified by a threshold travel distance [173]). These localization updates could also have impacted the accuracy of the UKF predictions, however, the complexity of interactions between the many parameters makes adjustment of such parameters potentially impractical.

6. Conclusion

In this report, the transfer of algorithms and simulations for evaluating continuous mobile manipulator performance was presented. This included the presentation of a coordinate system transformation and registration measurement procedure needed to unify the AMR-CT map coordinate system with the OTS coordinate system used as ground truth. Additionally, the design of a 2^3 factorial experiment and accompanied dataset was thoroughly described. The designed experiment and EDA were conducted to evaluate which of the main factors (i.e., search algorithm, AMR-CT speed, and measured CMMA side), including interactions, had the most impact on the accuracy (as measured by the fiducial interception rate) and speed (as measured by the average fiducial search time) of the continuous mobile manipulator task.

From the EDA, it was determined that the search algorithm was the most important factor impacting both the fiducial interception rate and the average fiducial search time. Additionally, the AMR-CT speed was also considered an the second most important factor for the former response variable and third important for the latter response variable. For the average fiducial search time, the interaction between AMR-CT speed and search algorithm of the second most important factor, though it should be noted that the effect size for all

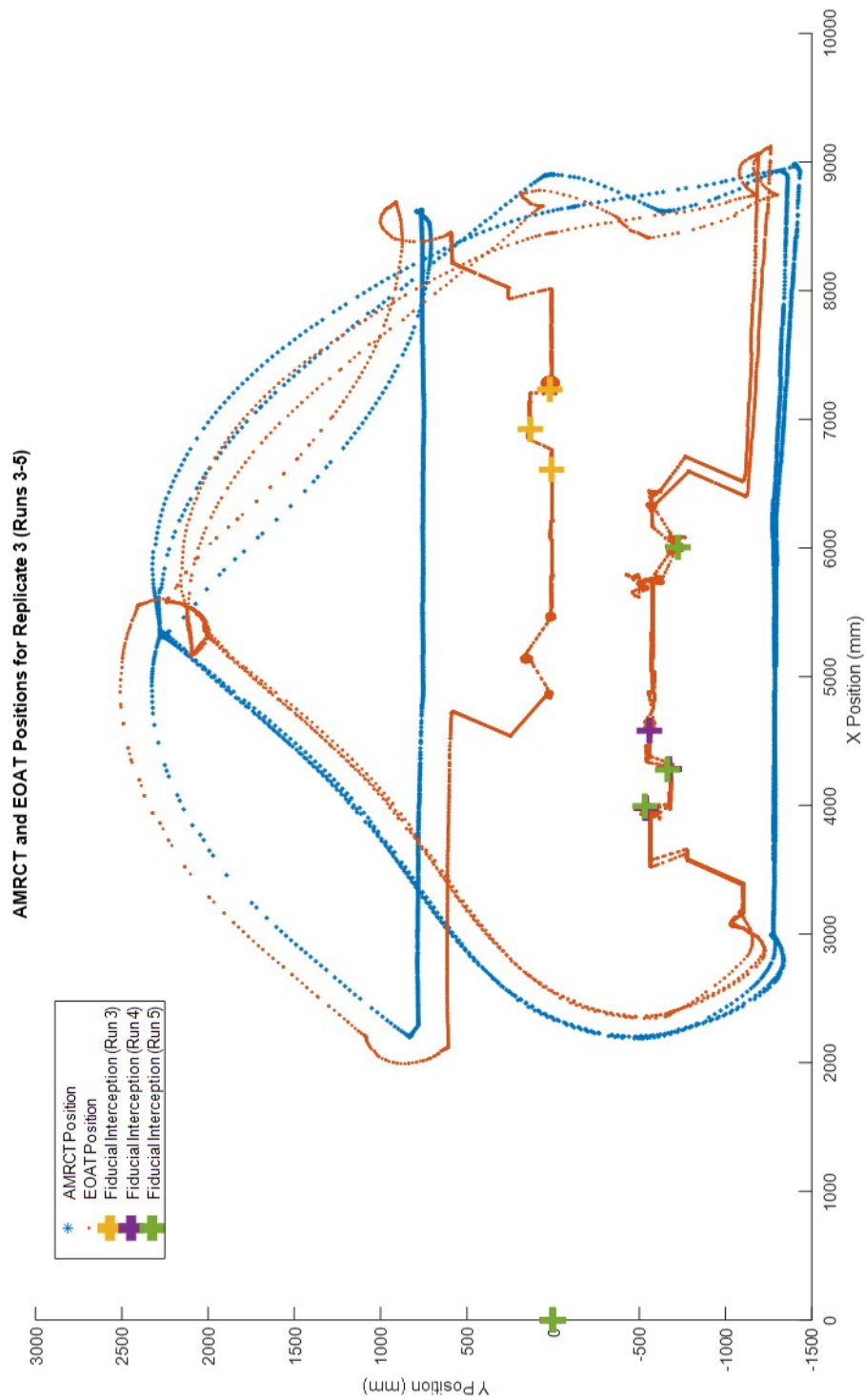


Fig. 49. Plot of the AMR-CT base and EOAT position during replicate 3 (runs 3-5). The intercepted fiducial positions are also plotted for reference, with the marks at the origin representing undetected fiducials.

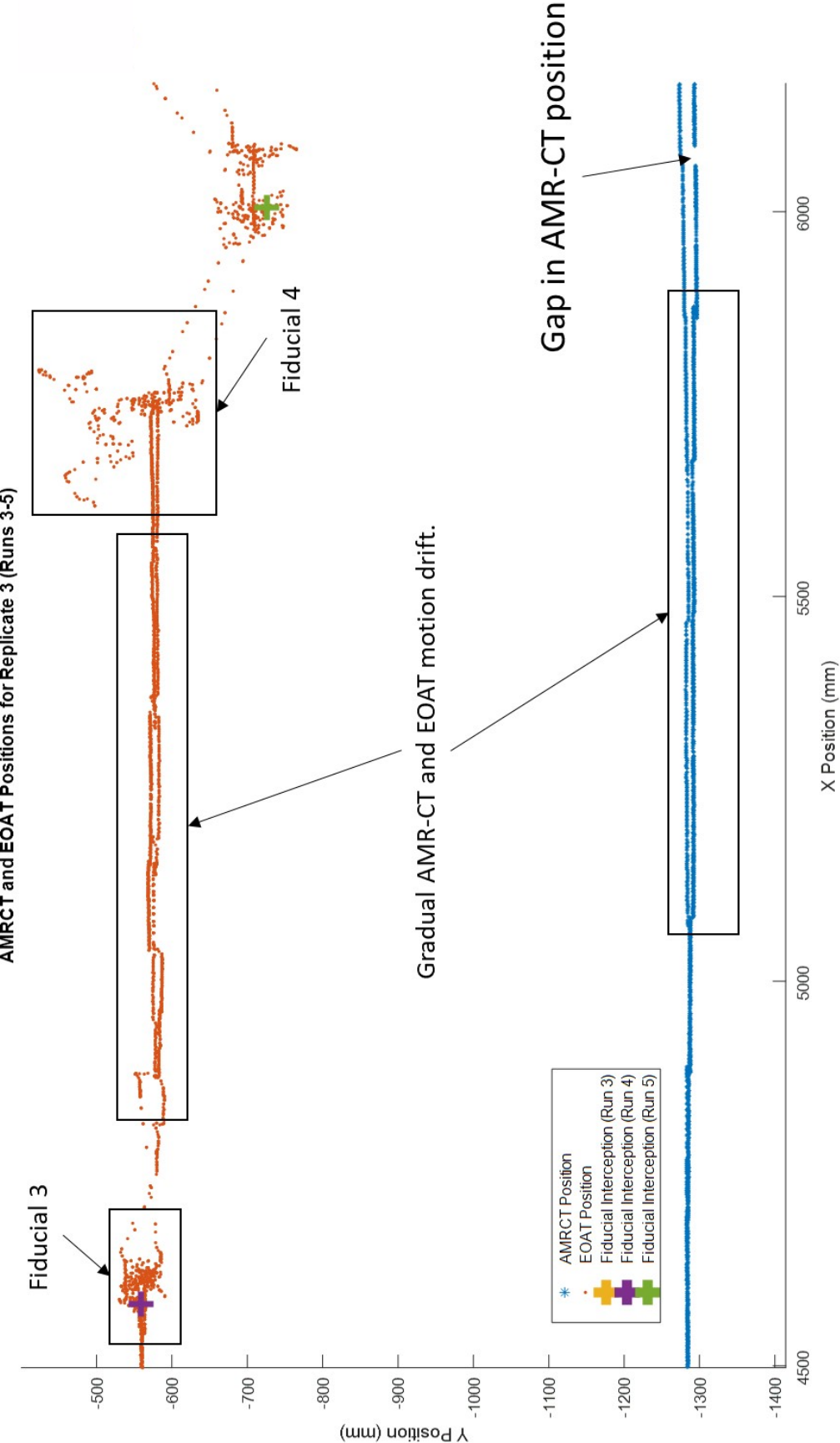


Fig. 50. Zoomed plot of the AMR-CT base and EOAT position during replicate 3 (runs 3-5). The intercepted fiducial positions are also plotted for reference, with the marks at the origin representing undetected fiducials. The zoomed in view shows Side 1 of the CMMA between fiducials 3 and 4.

of the main factors and interactions had engineering significance for the average fiducial search time. For the fiducial interception rate, only the search algorithm had engineering significance. The best settings for the fiducial interception rate were the spiral search, the slow AMR-CT speed of 0.01 m/s, and side 2 of the CMMA. For the average fiducial search time, the spiral search, the fast AMR-CT speed of 0.025 m/s, and side 1 of the CMMA.

Potential areas of improvement to the mobile manipulator initial registration and UKF were identified through the experiment and included the following. First was using an initial registration aid, such as reflective tape, and creating separately-tuned hyper-parameters when searching for the first fiducial. Such improvements should prevent occurrences of Anomaly 3 (i.e., when the first fiducial could not be found, the search for all subsequent fiducials would be aborted). Further tuning of the hyper-parameters might also improve the UKF performance in light of several possible sources of error, such as 1) different fiducial layout from the experiment conducted in Ref. [10], 2) the sparse search trajectory observed between pairs of fiducials that have large inter-fiducial distances, 3) motion drift introduced by frequent heading corrections (attributed to uneven flooring and unequal weight distribution on the mobile manipulator payload structure), and 4) AMR-CT localization performance. Additional improvements to the test implementation include upgrading the manipulator software drivers from `ur_modern_driver`, which was deprecated, to the latest supported manipulator drivers [86, 174].

With both the non-continuous and continuous cases of mobile manipulator performance capabilities now having completed example test implementations, future expansions to the NIST measurement methodology using the CMMA may now expand to evaluate performance scenarios that lie on a spectrum between or beyond these two capabilities. For example, future mobile manipulator performance test implementations can evaluate situations in which the mobile base and manipulator of a mobile manipulator do not simultaneously move, but the workpiece being operated on might be disturbed [175]. Furthermore, the performance evaluation of continuous performance capabilities can be expanded to include scenarios where the mobile base, manipulator, and workpiece are all simultaneously in motion, as has been observed in some recent works [176–178]. The evaluation of both capabilities holds value in adopting mobile manipulators in unstructured manufacturing environments and for heterogeneous robot collaboration, respectively.

References

- [1] Bostelman RV, Hong T, Marvel J (2016) Survey of research for performance measurement of mobile manipulators. *Journal of Research (NIST JRES)* 121:342–366. <https://doi.org/10.6028/jres.121.015>
- [2] Li G, Huang Y, Zhang X, Liu C, Shao W, Jiang L, Meng J (2020) Hybrid maps enhanced localization system for mobile manipulator in harsh manufacturing workshop. *IEEE Access* 8:10782–10795. <https://doi.org/10.1109/ACCESS.2020.2965300>

- [3] Gmerek AJ, Plastropoulos A, Collins P, Kimball M, Amit AW, Liu J, Karfakis PT, K Shah J, Carroll, Virk GS (2018) A novel holonomic mobile manipulator robot for construction sites. *Robotics Transforming the Future – Proceedings of the 21st International Conference on Climbing and Walking Robots and the support Technologies for Mobile Machines (CLAWAR 2018)*, pp 375 – 384.
- [4] Bo T, XingWei Z, Han D (2019) Mobile-robotic machining for large complex components: A review study. *SCIENCE CHINA-TECHNOLOGICAL SCIENCES* 62(8, SI):1388–1400. <https://doi.org/10.1007/s11431-019-9510-1>
- [5] Wang J, Tao B, Gong Z, Yu S, Yin Z (2021) A mobile robotic measurement system for large-scale complex components based on optical scanning and visual tracking. *Robotics and Computer-Integrated Manufacturing* 67:102010. <https://doi.org/https://doi.org/10.1016/j.rcim.2020.102010>. Available at <https://www.sciencedirect.com/science/article/pii/S0736584520302210> (Accessed August 22, 2023)
- [6] Li F, Jiang Y, Li T (2021) A laser-guided solution to manipulate mobile robot arm terminals within a large workspace. *IEEE/ASME Transactions on Mechatronics* 26(5):2676–2687. <https://doi.org/10.1109/TMECH.2020.3044461>
- [7] Zhou K, Ebenhofer G, Eitzinger C, Zimmermann U, Walter C, Saenz J, Castaño LP, Hernández MAF, Oriol JN (2014) Mobile manipulator is coming to aerospace manufacturing industry. *2014 IEEE International Symposium on Robot and Sensors Environments (ROSE) Proceedings*, pp 94–99. <https://doi.org/10.1109/ROSE.2014.6952990>
- [8] Dhanaraj N, Yoon YJ, Malhan R, Bhatt PM, Thakar S, Gupta SK (2022) A mobile manipulator system for accurate and efficient spraying on large surfaces. *Procedia Computer Science* 200:1528–1539. <https://doi.org/https://doi.org/10.1016/j.procs.2022.01.354>. Available at <https://www.sciencedirect.com/science/article/pii/S1877050922003635> 3rd International Conference on Industry 4.0 and Smart Manufacturing (Accessed August 22, 2023)
- [9] Amoako-Frimpong SY, Messina M, Medeiros H, Marvel J, Bostelman R (2018) Stochastic search methods for mobile manipulators. *Procedia Manufacturing* 17:976–984. <https://doi.org/10.1016/j.promfg.2018.10.106>. Available at <https://www.sciencedirect.com/science/article/pii/S235197891831223X> 28th International Conference on Flexible Automation and Intelligent Manufacturing (FAIM2018), June 11-14, 2018, Columbus, OH, USA Global Integration of Intelligent Manufacturing and Smart Industry for Good of Humanity (Accessed August 22, 2023)
- [10] Jiayu X (2021) *Stochastic Modeling for Mobile manipulators* Master’s thesis Marquette University 1515 W. Wisconsin Ave. Milwaukee, WI 53233. Available at https://epublications.marquette.edu/theses_open/687/ (Accessed August 22, 2023).
- [11] Hamner B, Koterba S, Shi J, Simmons R, Singh S (2009) Mobile robotic dynamic tracking for assembly tasks. *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp 2489–2495. <https://doi.org/10.1109/IROS.2009.5354237>
- [12] DeSouza G, Kak A (2004) A subsumptive, hierarchical, and distributed vision-based architecture for smart robotics. *IEEE Transactions on Systems, Man, and Cybernet-*

- ics, *Part B (Cybernetics)* 34(5):1988–2002. <https://doi.org/10.1109/TSMCB.2004.831768>
- [13] Yoon Y, Kosaka A, Kak AC (2008) A new Kalman-filter-based framework for fast and accurate visual tracking of rigid objects. *IEEE Transactions on Robotics* 24(5):1238–1251. <https://doi.org/10.1109/TRO.2008.2003281>
 - [14] Bostelman RV, Hong T, Marvel JA (2015) Performance measurement of mobile manipulators. *Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications 2015*, ed Braun JJ International Society for Optics and Photonics (SPIE), Vol. 9498, pp 97 – 106. <https://doi.org/10.1117/12.2177344>. Available at https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=918268 (Accessed July 17, 2023)
 - [15] Bostelman RV, Aboul-Enein O, Yoon S, Li-Baboud YS (2022) Design and application of the reconfigurable mobile manipulator artifact (RMMA), . <https://doi.org/https://doi.org/10.6028/NIST.AMS.100-46>. Available at https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=933475 (Accessed August 22, 2023)
 - [16] Pankert J, Hutter M (2020) Perceptive model predictive control for continuous mobile manipulation. *IEEE Robotics and Automation Letters* 5(4):6177–6184. <https://doi.org/10.1109/LRA.2020.3010721>
 - [17] Sprovieri J (2016) New technology for robotic welding. *Assembly Magazine* Available at <https://www.assemblymag.com/articles/93555-new-technology-for-robotic-welding> (Accessed August 16th, 2023).
 - [18] Sustarevas J, Butters D, Hammid M, Dwyer G, Stuart-Smith R, Pawar VM (2018) MAP - a mobile agile printer robot for on-site construction. *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp 2441–2448. <https://doi.org/10.1109/IROS.2018.8593815>
 - [19] Zhang X, Li M, Lim JH, Weng Y, Tay YWD, Pham H, Pham QC (2018) Large-scale 3D printing by a team of mobile robots. *Automation in Construction* 95:98–106. <https://doi.org/https://doi.org/10.1016/j.autcon.2018.08.004>. Available at <https://www.sciencedirect.com/science/article/pii/S0926580518304011> (Accessed August 22, 2023)
 - [20] Tiriyaki ME, Zhang X, Pham QC (2019) Printing-while-moving: a new paradigm for large-scale robotic 3D printing. *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp 2286–2291. <https://doi.org/10.1109/IROS40897.2019.8967524>
 - [21] Xu X, Wang Z, Feng C (2021) Projector-guided non-holonomic mobile 3D printing. *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp 8039–8045. <https://doi.org/10.1109/ICRA48506.2021.9561719>
 - [22] Sustarevas J, Kanoulas D, Julier S (2021) Task-consistent path planning for mobile 3D printing. *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp 2143–2150. <https://doi.org/10.1109/IROS51168.2021.9635916>
 - [23] Sustarevas J, Kanoulas D, Julier S (2022) Autonomous mobile 3D printing of large-scale trajectories. *2022 IEEE/RSJ International Conference on Intelligent*

- Robots and Systems (IROS)*, pp 6561–6568. <https://doi.org/10.1109/IROS47612.2022.9982274>
- [24] Bhatt PM, Malhan RK, Shembekar AV, Yoon YJ, Gupta SK (2020) Expanding capabilities of additive manufacturing through use of robotics technologies: A survey. *Additive Manufacturing* 31:100933. <https://doi.org/10.1016/j.addma.2019.100933>. Available at <http://www.sciencedirect.com/science/article/pii/S2214860419312266> (August 22, 2023)
- [25] (2018) National Institute of Standards and Technology (NIST) robotic systems for smart manufacturing. Available at www.nist.gov/programs-projects/robotic-systems-smart-manufacturing-program (Accessed August 9th, 2023).
- [26] (2018) National Institute of Standards and Technology (NIST) measurement science for manufacturing robotics. Available at <https://www.nist.gov/programs-projects/measurement-science-manufacturing-robotics> (Accessed August 9th, 2023).
- [27] (2018) National Institute of Standards and Technology (NIST) mobility performance of robotic systems. Available at <https://www.nist.gov/programs-projects/mobility-performance-robotic-systems> (Accessed August 9th, 2023).
- [28] Aboul-Enein O, Jing Y, Bostelman R (2020) Formalizing performance evaluation of mobile manipulator robots using CTML. *Proceedings of the ASME 2020 International Mechanical Engineering Congress and Exposition (IMECE 2020)* (ASME), Vol. Volume 2B: Advanced Manufacturing. <https://doi.org/10.1115/IMECE2020-23234>. Available at https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=928432 (Accessed July 17, 2023) V02BT02A004.
- [29] Bostelman RV, Foufou S, Legowik SA, Hong TH (2016) Mobile manipulator performance measurement towards manufacturing assembly tasks. *Proceedings of the 13th IFIP International Conference on Product Lifecycle Management (PLM16)*, eds Harik R, Rivest L, Bernard A, Eynard B, Bouras A (Springer International Publishing), Vol. 492, pp 411–420. https://doi.org/10.1007/978-3-319-54660-5_37. Available at https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=920470 (Accessed August 22, 2023)
- [30] Bostelman RV, Hong T, Legowik S (2016) Mobile robot and mobile manipulator research towards ASTM standards development. *Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications 2016*, ed Braun JJ International Society for Optics and Photonics (SPIE), Vol. 9872, pp 111 – 120. <https://doi.org/10.1117/12.2228464>
- [31] Bostelman RV, Eastman R, Hong TH, Aboul-Enein O, Legowik SA, Foufou S (2016) Comparison of registration methods for mobile manipulators. *Advances in Cooperative Robots*, eds Tokhi MO, Virk GS (World Scientific), pp 205–213. https://doi.org/10.1142/9789813149137_0026. Available at https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=921002 (Accessed July 17, 2023)
- [32] Bostelman RV, Li-Baboud YS, Legowik SA, Foufou S (2017) Mobile manipulator performance measurement data (National Institute of Standards and Technology (NIST), 100 Bureau Drive Gaithersburg, MD 20899), <https://doi.org/10.6028/NIST>.

[TN.1965](#)

- [33] Bostelman RV, Li-Baboud Y, Yoon S, Shah M, Aboul-Enein O (2020) Towards measurement of advanced mobile manipulator performance for assembly applications (100 Bureau Drive Gaithersburg, MD 20899), <https://doi.org/10.6028/NIST.TN.2108>
- [34] Aboul-Enein O, Bostelman R, Li-Baboud YS, Shah M (2022) Performance measurement of a mobile manipulator-on-a-cart and coordinate registration methods for manufacturing applications (100 Bureau Drive Gaithersburg, MD 20899), <https://doi.org/https://doi.org/10.6028/NIST.AMS.100-45r1>. Available at https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=934492 (Accessed July 17, 2023)
- [35] Aboul-Enein O, Medeiros H, Shah M, Li-Baboud YS, Bostelman R, Virts A (2023) Continuous mobile manipulator performance experiment 02-01-2022. <https://doi.org/10.18434/mds2-3061>.
- [36] (2018) Intel Core i7-8850H processor specifications. Available at <https://ark.intel.com/content/www/us/en/ark/products/134899/intel-core-i78850h-processor-9m-cache-up-to-4-30-ghz.html> (Accessed August 16th, 2023).
- [37] (2018) NVIDIA QUADRO P600 datasheet. Available at <https://www.nvidia.com/content/dam/en-zz/Solutions/design-visualization/productspage/quadro/quadro-desktop/quadro-pascal-p600-data-sheet-us-nv-704532-r1.pdf> (Accessed August 16th, 2023).
- [38] (2023) Install linux on windows with wsl. Available at <https://learn.microsoft.com/en-us/windows/wsl/install> (Accessed August 16th, 2023).
- [39] (2023) ROS wiki: melodic. Available at <https://wiki.ros.org/melodic> (Accessed August 16th, 2023).
- [40] (2023) ROS wiki: noetic/installation/Ubuntu. Available at <https://wiki.ros.org/noetic/Installation/Ubuntu> (Accessed August 16th, 2023).
- [41] (2017) Ubuntu documentation: Repositories. Available at <https://help.ubuntu.com/community/Repositories/Ubuntu> (Accessed August 16th, 2023).
- [42] (2015) Ubuntu documentation: Repositories command line. Available at <https://help.ubuntu.com/community/Repositories/CommandLine> (Accessed August 16th, 2023).
- [43] (2014) Installing catkin_tools. Available at <https://catkin-tools.readthedocs.io/en/latest/installing.html> (Accessed August 16th, 2023).
- [44] (2015) 'catkin build' gives 'command not found'. Available at <https://answers.ros.org/question/207433/catkin-build-gives-command-not-found/> (Accessed August 16th, 2023).
- [45] Purvis M (2020) Python 3 release - will be 0.7.0. Available at https://github.com/catkin/catkin_tools/issues/594 (Accessed August 16th, 2023).
- [46] (2014) osrf_pycommon. Available at <https://osrf-pycommon.readthedocs.io/en/latest/> (Accessed August 16th, 2023).
- [47] (2022) Installing Gazebo with ROS. Available at <https://gazebo.org/docs/latest/>

- [ros.installation](#) (Accessed August 16th, 2023).
- [48] (2023) Gazebo Classic: Source code. Available at <https://github.com/gazebo/gazebo-classic> (Accessed August 16th, 2023).
- [49] (2023) gazebo_ros_pkg: Source code. Available at https://github.com/ros-simulation/gazebo_ros_pkgs (Accessed August 16th, 2023).
- [50] (2022) VcXsrv Download. Available at <https://sourceforge.net/projects/vcxsrv/> (Accessed August 16th, 2023).
- [51] Bernlöhner J (2017) Running ROS on Windows 10. Available at <https://janbernloehr.de/2017/06/10/ros-windows> (Accessed August 16th, 2023).
- [52] RickMakes (2020) Windows subsystem for linux 1: Installing vcxrv x server. Available at https://www.youtube.com/watch?v=68_FgUGQGhc YouTube video. (Accessed August 16th, 2023).
- [53] Yates J (2020) The complete WSL2 + GUI setup. Available at <https://medium.com/@japheth.yates/the-complete-wsl2-gui-setup-2582828f4577> (Accessed August 16th, 2023).
- [54] (2018) Ubuntu 18.4 libQt5Core.so.5: cannot open shared object file: No such file or directory. Available at <https://askubuntu.com/questions/1034313/ubuntu-18-4-libqt5core-so-5-cannot-open-shared-object-file-no-such-file-or-dir> (Accessed August 16th, 2023).
- [55] (2022) ROS Wiki: rosdep. Available at <https://wiki.ros.org/rosdep> (Accessed August 16th, 2023).
- [56] (2019) LD series mobile robot datasheet. Available at https://assets.omron.eu/downloads/datasheet/en/v2/ld-series_mobile_robot_datasheet_en.pdf (Accessed September 6, 2023).
- [57] (2021) Mobile planner software download. Available at <https://automation.omron.com/en/us/products/family/Mobile%20Planner> (Accessed August 9th, 2023).
- [58] Omron Adept Technologies 4225 Hacienda Drive, Pleasanton, CA 94588 (2016) *Mobile Robot Software Suite User's Guide*, I614-E-01 Ed. Available at <https://assets.omron.com/m/0c55a28223f13734/original/Mobile-Robot-Software-Suite-User-s-Guide.pdf> (Accessed July 17, 2023).
- [59] (2013) Pioneer LX mobile research platform datasheet. Available at <https://www.generationrobots.com/media/Pioneer%20LX%20datasheet.pdf> (Accessed August 9th, 2023).
- [60] (2019) Pioneer LX. Available at <https://robots.ros.org/pioneer-lx/> (Accessed August 9th, 2023).
- [61] (2019) Adept MobileRobots Pioneer and Pioneer-compatible platforms. Available at http://wiki.ros.org/Robots/AMR_Pioneer_Compatible (Accessed August 9th, 2023).
- [62] Omron Adept Technologies 4225 Hacienda Drive, Pleasanton, CA 94588 (2016) *Advanced Robotics Command Language Reference Guide*, I617-E-01 Ed. Available at <https://assets.omron.com/m/6f13d7c70932964d/original/Advanced-Robotics-Command-Language-Reference-Guide.pdf> (Accessed July 17, 2023).

- [63] Proctor F (2019) ULAPI download. Available at <https://github.com/frederickproctor/ulapi> (Accessed July 17, 2023).
- [64] Proctor F (2018) Ulapl readme. Available at <https://github.com/frederickproctor/ulapi/blob/master/README.Linux> (Accessed August 9th, 2023).
- [65] (2010) Making configure scripts. Available at https://www.gnu.org/software/autoconf/manual/autoconf-2.67/html_node/Making-configure-Scripts.html (Accessed August 9th, 2023).
- [66] (2021) Twist message. Available at https://docs.ros.org/en/noetic/api/geometry_msgs/html/msg/Twist.html (Accessed August 9th, 2023).
- [67] (2018) Publishers and subscribers. Available at <https://wiki.ros.org/roscpp/Overview/Publishers%20and%20Subscribers> (Accessed August 9th, 2023).
- [68] (2019) Writing a simple publisher and subscriber (C++). Available at <https://wiki.ros.org/ROS/Tutorials/WritingPublisherSubscriber%28c%2B%2B%29> (Accessed August 9th, 2023).
- [69] Corke P (2017) *Robotics, Vision and Control: Fundamental Algorithms in MATLAB* (Springer International Publishing AG), Chapter 6, 2nd Ed., pp 151–154. <https://doi.org/10.1007/978-3-319-54413-7>
- [70] (2012) Services. Available at <http://wiki.ros.org/roscpp/Overview/Services> (Accessed August 9th, 2023).
- [71] (2019) Writing a simple service and client (C++). Available at <http://wiki.ros.org/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29> (Accessed August 9th, 2023).
- [72] (2020) Creating a ros msg and srv. Available at <http://wiki.ros.org/ROS/Tutorials/CreatingMsgAndSrv> (Accessed August 9th, 2023).
- [73] (2015) Overview of tf datatypes. Available at <http://wiki.ros.org/tf/Overview/Data%20Types> (Accessed August 9th, 2023).
- [74] (2021) Writing a tf broadcaster (C++). Available at <http://wiki.ros.org/tf/Tutorials/Writing%20a%20tf%20broadcaster%20%28C%2B%2B%29> (Accessed August 9th, 2023).
- [75] (2016) Pro 3600 owner’s manual. Available at https://www.leveldevelopments.com/wp/wp-content/uploads/documents/PRO3600_Instructions.pdf (Accessed August 9th, 2023).
- [76] (2011) XE and EP batteries range summary. Available at https://www.enersys.com/493c0d/globalassets/documents/product-documentation/genesis/pure-lead/apac/en-xe-ep-rs-005_0911.pdf (Accessed July 17, 2023).
- [77] (2010) ML900-24 specification sheet. Available at http://powerbright.com/manuals/specs/ML900_spec_sheet.pdf (Accessed July 17, 2023).
- [78] (2006) UL(AC), DL(DC) series datasheet. Available at <https://assets.alliedelec.com/v1620727753/Datasheets/b181f9d764d2f836322247b876112d39.pdf> (Accessed July 17, 2023).
- [79] (2016) Universal Robots UR5 technical specifications. Available at https://www.universal-robots.com/media/50588/ur5_en.pdf (Accessed July 17, 2023).

- [80] (2022) WORLD-BEAM® QS18LLP series datasheet. Available at <http://info.bannerengineering.com/cs/groups/public/documents/literature/118900.pdf> (Accessed July 17, 2023).
- [81] (2020) Universal robots legacy download center. Available at <https://www.universal-robots.com/articles/ur/documentation/legacy-download-center/> (Accessed August 9th, 2023).
- [82] Universal Robots A/S Energivej 25, DK-5260 Odense, Denmark (2019) *Universal Robots Service Manual*, UR5_en.3.2.7. Ed. Available at https://s3-eu-west-1.amazonaws.com/ur-support-site/15737/UR5_Service_Manual_en.pdf (Accessed July 17, 2023).
- [83] (2019) Real-Time Data Exchange (RTDE) guide. Available at <https://www.universal-robots.com/articles/ur/interface-communication/real-time-data-exchange-rtde-guide/>.
- [84] Coleman D, Şucan IA, Chitta S, Correll N (2014) Reducing the barrier to entry of complex robotic software: a MoveIt! case study. *Journal of Software Engineering for Robotics* 5(1):3–16. https://doi.org/10.6092/JOSER_2014_05_01_p3
- [85] Sucan IA, Chitta S MoveIt. Available at movit.ros.org (Accessed August 16th, 2023).
- [86] Andersen TT (2015) Optimizing the universal robots ros driver. (Technical University of Denmark, Department of Electrical Engineering), Available at [http://orbit.dtu.dk/en/publications/optimizing-the-universal-robots-ros-driver\(20dde139-7e87-4552-8658-dbf2cdaab24b\).html](http://orbit.dtu.dk/en/publications/optimizing-the-universal-robots-ros-driver(20dde139-7e87-4552-8658-dbf2cdaab24b).html).
- [87] Chitta S, Marder-Eppstein E, Meeussen W, Pradeep V, Rodríguez Tsouroukdissian A, Bohren J, Coleman D, Magyar B, Raiola G, Lüdtke M, Fernández Perdomo E (2017) ros_control: A generic and simple control framework for ros. *The Journal of Open Source Software* <https://doi.org/10.21105/joss.00456>. Available at <http://www.theoj.org/joss-papers/joss.00456/10.21105.joss.00456.pdf>
- [88] (2019) "Low Bandwidth Trajectory Follower" cannot connect to robot. Available at https://github.com/ros-industrial/ur_modern_driver/issues/336 (Accessed August 16th, 2023).
- [89] (2022) ur_modern_driver README.md. Available at https://github.com/ros-industrial/ur_modern_driver/blob/kinetic-devel/README.md (Accessed August 16th, 2023).
- [90] Bostelman RV (2018) *Performance measurement of mobile manipulators*. Ph.D. thesis. Université Bourgogne Franche-Comté, . Available at <https://tel.archives-ouvertes.fr/tel-01803721>.
- [91] (2022) Intel Xeon W-2145 processor specifications. Available at <https://www.intel.com/content/www/us/en/products/sku/126707/intel-xeon-w2145-processor-11m-cache-3-70-ghz/specifications.html> (Accessed August 9th, 2023).
- [92] (2019) NVIDIA Titan RTX product overview. Available at

- <https://www.nvidia.com/content/dam/en-zz/Solutions/titan/documents/titan-rtx-for-creators-us-nvidia-1011126-r6-web.pdf> (Accessed August 9th, 2023).
- [93] (2023) Motive documentation calibration. Available at <https://docs.optitrack.com/motive/calibration> (Accessed August 9th, 2023).
- [94] (2015) Cw-500 technical drawing. Available at <https://d111srqycjesc9.cloudfront.net/CW-500TechnicalDrawing.pdf>.
- [95] Bostelman RV, Falco JA, Shah M, Hong TH (2016) *Autonomous Industrial Vehicles: From the Laboratory to the Factory Floor* (ASTM International), Chapter Dynamic Metrology Performance Measurement of a Six Degree-Of-Freedom Tracking System used in Smart Manufacturing, pp 91–105. <https://doi.org/10.1520/STP1594-EB>
- [96] (2023) Motive documentation rigid body tracking. Available at <https://docs.optitrack.com/motive/rigid-body-tracking> (Accessed August 9th, 2023).
- [97] Odom W, Knott T (2006) *Networking Basics: CCNA 1 Companion Guide*. Cisco Networking Academy Program series (Cisco Press).
- [98] (2022) National institute of standards and technology time servers. Available at <https://www.time.gov/> (Accessed August 9th, 2023).
- [99] (2020) W32tm. Available at [https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-r2-and-2012/ff799054\(v=ws.11\)](https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-r2-and-2012/ff799054(v=ws.11)) (Accessed August 9th, 2023).
- [100] (2020) Wsl2 date incorrect after waking from sleep. Available at <https://github.com/microsoft/WSL/issues/5324> (Accessed August 9th, 2023).
- [101] Shah M (2011) Comparing two sets of corresponding six degree of freedom data. *Computer Vision and Image Understanding* 115(10):1355 – 1362. <https://doi.org/10.1016/j.cviu.2011.05.007>. Available at <https://www.sciencedirect.com/science/article/pii/S1077314211001378> (Accessed August 22, 2023)
- [102] Shah M, Eastman RD, Hong T (2012) An overview of robot-sensor calibration methods for evaluation of perception systems. *Proceedings of the Workshop on Performance Metrics for Intelligent Systems* PerMIS '12 (Association for Computing Machinery, New York, NY, USA), p 15–20. <https://doi.org/10.1145/2393091.2393095>
- [103] Shah M (2014) Calibration and registration techniques for robotics. Available at <http://faculty.cooper.edu/mili/Calibration/index.html> (Accessed August 9th, 2023).
- [104] Mount G (2017) Generate unique, non-repeating list of numbers in excel with rand and rank. Available at <https://www.youtube.com/watch?v=gmATWzSVeKM> youtube Video (Accessed August 9th, 2023).
- [105] Omron Adept Technologies 4225 Hacienda Drive, Pleasanton, CA 94588 (2019) *LD Cart Transporter User's Guide*, I612-E-05 Ed. Available at <https://assets.omron.com/m/65b76dca5a05032a/original/LD-Cart-Transporter-User-s-Guide.pdf> (Accessed July 17, 2023).
- [106] (2021) Anaconda individual edition download. Available at <https://www.anaconda.com/products/individual> (Accessed August 9th, 2023).
- [107] (2021) Pandas documentation. Available at <https://pandas.pydata.org/pandas-docs/>

- [stable/index.html](#) (Accessed August 9th, 2023).
- [108] McIntire G, Martin B, Washington L (2021) Python pandas tutorial: A complete introduction for beginners. Available at <https://www.learndatasci.com/tutorials/python-pandas-tutorial-complete-introduction-for-beginners/> (Accessed August 9th, 2023).
- [109] (2014) How to count the nan values in a column in pandas dataframe. Available at <https://stackoverflow.com/questions/26266362/how-to-count-the-nan-values-in-a-column-in-pandas-dataframe> (Accessed August 9th, 2023).
- [110] Hunter J, Dale D, Firing E, Droettboom M (2021) Matplotlib documentation. Available at <https://matplotlib.org/stable/index.html> (Accessed August 9th, 2023).
- [111] T B (2020) Clearing the confusion once and for all: fig, ax = plt.subplots(). Available at <https://towardsdatascience.com/clearing-the-confusion-once-and-for-all-fig-ax-plt-subplots-b122bb7783ca> (Accessed August 9th, 2023).
- [112] (2018) plt.show() does nothing when used for the second time. Available at <https://stackoverflow.com/questions/50452455/plt-show-does-nothing-when-used-for-the-second-time> (Accessed August 9th, 2023).
- [113] (2021) scipy.spatial.transform.rotation. Available at <https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.transform.Rotation.html> (Accessed August 9th, 2023).
- [114] (2021) Numpy documentation. Available at <https://numpy.org/doc/stable/index.html> (Accessed August 9th, 2023).
- [115] Corke P (2017) *Robotics, Vision and Control: Fundamental Algorithms in MATLAB* (Springer International Publishing AG), Chapter 2, 2nd Ed., pp 17–62. <https://doi.org/10.1007/978-3-319-54413-7>
- [116] Shah M (2014) Arun.m source code. Available at <http://faculty.cooper.edu/mili/Calibration/RT/arun.m> (Accessed August 9th, 2023).
- [117] Shah M (2014) Shahone.m source code. Available at <http://faculty.cooper.edu/mili/Calibration/AXYB/shahOne.m> (Accessed August 9th, 2023).
- [118] (2014) How do i put a title for my entire subplot?. Available at <https://www.mathworks.com/matlabcentral/answers/165828-how-do-i-put-a-title-for-my-entire-subplot> (Accessed August 9th, 2023).
- [119] (2015) How can i change the font size of plot tick labels?. Available at <https://www.mathworks.com/matlabcentral/answers/173658-how-can-i-change-the-font-size-of-plot-tick-labels> (Accessed August 9th, 2023).
- [120] (2016) scatter3 makes a 2d plot. Available at <https://www.mathworks.com/matlabcentral/answers/300221-scatter3-makes-a-2d-plot> (Accessed August 9th, 2023).
- [121] (2017) How to plot multiple data sets on the same graph.

- Available at <https://www.mathworks.com/matlabcentral/answers/357557-how-to-plot-multiple-data-sets-on-the-same-graph> (Accessed August 9th, 2023).
- [122] (2019) MATLAB R2019b documentation. Available at <https://www.mathworks.com/matlabcentral/answers/173658-how-can-i-change-the-font-size-of-plot-tick-labels> (Accessed August 9th, 2023).
- [123] Gaschler A, Brudnak M, Plesch A, Herbert J (2022) 3D rotation converter. Available at <https://www.andre-gaschler.com/rotationconverter/> (Accessed August 9th, 2023).
- [124] Gaschler A, Brudnak M, Plesch A, Herbert J (2019) 3D rotation converter source code. Available at <https://github.com/gaschler/rotationconverter> (Accessed August 9th, 2023).
- [125] (2023) ROS Wiki: urdf. Available at <https://wiki.ros.org/urdf> (Accessed August 9th, 2023).
- [126] (2017) ROS Wiki: tf debugging tools. Available at <https://wiki.ros.org/tf/Debugging%20tools> (Accessed August 9th, 2023).
- [127] Meeussen W (2016) ROS documentation: joint_state_controller. Available at https://docs.ros.org/en/jade/api/joint_state_controller/html/c++/classjoint__state__controller_1_1JointStateController.html (Accessed August 9th, 2023).
- [128] (2020) ROS Wiki: robot_state_publisher. Available at https://wiki.ros.org/robot_state_publisher (Accessed August 9th, 2023).
- [129] (2015) Remote control via TCP/IP. Available at <https://www.universal-robots.com/articles/ur/interface-communication/remote-control-via-tcpip/> (Accessed August 9th, 2023).
- [130] Montgomery DC (2013) *Design and Analysis of Experiments* (John Wiley and Sons), Chapter 2, eighth Ed., p 14.
- [131] Drummond GB, Vowler SL (2012) Type i: families, planning and errors. *The Journal of Physiology* 590(20):4971–4974. <https://doi.org/10.1113/jphysiol.2012.244061>
- [132] Montgomery DC (2013) *Design and Analysis of Experiments* (John Wiley and Sons), Chapter 3, eighth Ed., p 98.
- [133] McClave J, Sincich T (2017) *Statistics* (Pearson Education), thirteenth Ed.
- [134] Montgomery DC (2013) *Design and Analysis of Experiments* (John Wiley and Sons), Chapter 6, eighth Ed., pp 233–234.
- [135] Montgomery DC (2013) *Design and Analysis of Experiments* (John Wiley and Sons), Chapter 6, eighth Ed., p 285.
- [136] Montgomery DC (2013) *Design and Analysis of Experiments* (John Wiley and Sons), Chapter 5, eighth Ed., p 184.
- [137] Montgomery DC (2013) *Design and Analysis of Experiments* (John Wiley and Sons), Chapter 6, eighth Ed., pp 241–253.
- [138] Montgomery DC (2013) *Design and Analysis of Experiments* (John Wiley and Sons), Chapter 5, eighth Ed., pp 206–211.
- [139] Montgomery DC (2013) *Design and Analysis of Experiments* (John Wiley and Sons),

- Chapter 3, eighth Ed., pp 80–88.
- [140] Montgomery DC (2013) *Design and Analysis of Experiments* (John Wiley and Sons), Chapter 5, eighth Ed., p 198.
 - [141] (2017) Catkin tutorials creating a workspace. Available at https://wiki.ros.org/catkin/Tutorials/create_a_workspace (Accessed August 9th, 2023).
 - [142] (2022) MATLAB. Available at <https://www.mathworks.com/products/matlab.html> (Accessed August 9th, 2023).
 - [143] Eaton JW (2022) Octave download. Available at <https://octave.org/download> (Accessed August 9th, 2023).
 - [144] (2022) roscpp overview time. Available at <https://wiki.ros.org/roscpp/Overview/Time> (Accessed August 9th, 2023).
 - [145] (2009) How to redirect output to file and stdout. Available at <https://stackoverflow.com/questions/418896/how-to-redirect-output-to-a-file-and-stdout> (Accessed August 9th, 2023).
 - [146] (2016) How do you put date and time in a filename. Available at <https://unix.stackexchange.com/questions/278939/how-do-you-put-date-and-time-in-a-file-name> (Accessed August 9th, 2023).
 - [147] (2020) Rosbag. Available at <https://wiki.ros.org/rosbag> (Accessed August 9th, 2023).
 - [148] Wagenmakers EJ (2022) Jasp 0.16.1 download. Available at <https://jasp-stats.org/download/> (Accessed August 9th, 2023).
 - [149] Goss-Sampson MA (2022) *Statistical Analysis in JASP 0.16.1: A Guide for Students*, 5th Ed. Available at <https://jasp-stats.org/wp-content/uploads/2022/04/Statistical-Analysis-in-JASP-A-Students-Guide-v16.pdf>.
 - [150] Montgomery DC (2013) *Design and Analysis of Experiments* (John Wiley and Sons), Chapter 3, eighth Ed., p 85.
 - [151] Trutna L, Spagon P, del Castillo E, Moore T, Hartley S, Hurwitz A (2012) *NIST/SEMATECH e-Handbook of Statistical Methods: An EDA Approach to Experimental Design* (National Institute of Standards and Technology (NIST)), Chapter 5, p 5.5.9. <https://doi.org/https://doi.org/10.18434/M32189>
 - [152] (2019) Attributeerror: module 'matplotlib' has no attribute 'scatter'. Available at <https://stackoverflow.com/questions/58540629/attributeerror-module-matplotlib-has-no-attribute-scatter> (Accessed August 9th, 2023).
 - [153] (2013) Combine two columns of text in pandas dataframe. Available at <https://stackoverflow.com/questions/19377969/combine-two-columns-of-text-in-pandas-dataframe> (Accessed August 9th, 2023).
 - [154] (2017) How to change spacing between ticks. Available at <https://stackoverflow.com/questions/44863375/how-to-change-spacing-between-ticks> (Accessed August 9th, 2023).
 - [155] (2016) Plotting lines connecting points. Available at <https://stackoverflow.com/questions/35363444/plotting-lines-connecting-points> (Accessed August 9th, 2023).

- [156] (2016) Pandas: calculating the mean values of duplicate entries in a dataframe. Available at <https://stackoverflow.com/questions/39919570/pandas-calculating-the-mean-values-of-duplicate-entries-in-a-dataframe> (Accessed August 9th, 2023).
- [157] (2019) How to show points in the scatter plot as plus and minus signs?. Available at <https://stackoverflow.com/questions/59443190/how-to-show-points-in-the-scatter-plot-as-plus-and-minus-signs> (Accessed August 9th, 2023).
- [158] (2017) How to hide lines in matplotlib? [duplicate]. Available at <https://stackoverflow.com/questions/44736034/how-to-hide-lines-in-matplotlib> (Accessed August 9th, 2023).
- [159] (2018) Matplotlib y axis values are not ordered [duplicate]. Available at <https://stackoverflow.com/questions/48062499/matplotlib-y-axis-values-are-not-ordered> (Accessed August 9th, 2023).
- [160] (2011) How to put the legend outside the plot. Available at <https://stackoverflow.com/questions/4700614/how-to-put-the-legend-outside-the-plot> (Accessed August 9th, 2023).
- [161] (2021) probplot draws fit line even when fit=false. Available at <https://github.com/scipy/scipy/issues/13769> (Accessed August 9th, 2023).
- [162] (2022) How to add an average line to plot in matplotlib. Available at <https://www.statology.org/matplotlib-average-line/> (Accessed August 9th, 2023).
- [163] (2022) How to calculate the mean by group in pandas (with examples). Available at <https://www.statology.org/pandas-mean-by-group/> (Accessed August 9th, 2023).
- [164] Matplotlib.figure.figure.set_size_inches() in python. Available at https://www.geeksforgeeks.org/matplotlib-figure-figure-set_size_inches-in-python/ (Accessed August 9th, 2023).
- [165] Trutna L, Spagon P, del Castillo E, Moore T, Hartley S, Hurwitz A (2012) *NIST/SEMATECH e-Handbook of Statistical Methods: Interaction effects matrix plot* (National Institute of Standards and Technology (NIST)), Chapter 5, p 5.5.9.4. <https://doi.org/https://doi.org/10.18434/M32189>. Available at <https://www.itl.nist.gov/div898/handbook/pri/section5/pri594.htm> accessed July 27, 2023
- [166] Trutna L, Spagon P, del Castillo E, Moore T, Hartley S, Hurwitz A (2012) *NIST/SEMATECH e-Handbook of Statistical Methods: Block plot* (National Institute of Standards and Technology (NIST)), Chapter 5, p 5.5.9.5. <https://doi.org/https://doi.org/10.18434/M32189>. Available at <https://www.itl.nist.gov/div898/handbook/pri/section5/pri595.htm> accessed July 27, 2023
- [167] Trutna L, Spagon P, del Castillo E, Moore T, Hartley S, Hurwitz A (2012) *NIST/SEMATECH e-Handbook of Statistical Methods: DOE Youden plot* (National Institute of Standards and Technology (NIST)), Chapter 5, p 5.5.9.6. <https://doi.org/https://doi.org/10.18434/M32189>. Available at <https://www.itl.nist.gov/div898/handbook/pri/section5/pri596.htm> accessed July 27, 2023
- [168] Trutna L, Spagon P, del Castillo E, Moore T, Hartley S, Hurwitz A (2012)

- NIST/SEMATECH e-Handbook of Statistical Methods: Ordered data plot* (National Institute of Standards and Technology (NIST)), Chapter 5, p 5.5.9.1. <https://doi.org/https://doi.org/10.18434/M32189>. Available at <https://www.itl.nist.gov/div898/handbook/pri/section5/pri591.htm> accessed July 27, 2023
- [169] Trutna L, Spagon P, del Castillo E, Moore T, Hartley S, Hurwitz A (2012) *NIST/SEMATECH e-Handbook of Statistical Methods: DOE scatter plot* (National Institute of Standards and Technology (NIST)), Chapter 5, p 5.5.9.2. <https://doi.org/https://doi.org/10.18434/M32189>. Available at <https://www.itl.nist.gov/div898/handbook/pri/section5/pri592.htm> accessed July 27, 2023
- [170] Trutna L, Spagon P, del Castillo E, Moore T, Hartley S, Hurwitz A (2012) *NIST/SEMATECH e-Handbook of Statistical Methods: An EDA Approach to Experimental Design* (National Institute of Standards and Technology (NIST)), Chapter 5, p 5.5.9.7.2. <https://doi.org/https://doi.org/10.18434/M32189>. Available at <https://www.itl.nist.gov/div898/handbook/pri/section5/pri5972.htm> accessed July 27, 2023
- [171] Trutna L, Spagon P, del Castillo E, Moore T, Hartley S, Hurwitz A (2012) *NIST/SEMATECH e-Handbook of Statistical Methods: —Effects— plot* (National Institute of Standards and Technology (NIST)), Chapter 5, p 5.5.9.7. <https://doi.org/https://doi.org/10.18434/M32189>. Available at <https://www.itl.nist.gov/div898/handbook/pri/section5/pri597.htm> accessed July 27, 2023
- [172] Trutna L, Spagon P, del Castillo E, Moore T, Hartley S, Hurwitz A (2012) *NIST/SEMATECH e-Handbook of Statistical Methods: An EDA Approach to Experimental Design* (National Institute of Standards and Technology (NIST)), Chapter 5, p 5.5.9.8. <https://doi.org/https://doi.org/10.18434/M32189>. Available at <https://www.itl.nist.gov/div898/handbook/pri/section5/pri598.htm> accessed July 27, 2023
- [173] Omron Adept Technologies 4225 Hacienda Drive, Pleasanton, CA 94588 (2016) *Mobile Robot Software Suite User's Guide*, I614-E-01 Ed. Available at <https://assets.omron.com/m/0c55a28223f13734/original/Mobile-Robot-Software-Suite-User-s-Guide.pdf> (Accessed July 17, 2023).
- [174] (2021) Universal_Robots_ROS_Driver source code. Available at https://github.com/UniversalRobots/Universal_Robots_ROS_Driver (Accessed August 16th, 2023).
- [175] Aboul-Enein O, Li-Baboud YS, Bostelman R, Virts A (2023) Design and implementation of a closed-loop mobile manipulator control system, . <https://doi.org/https://doi.org/10.6028/NIST.TN.2258>. Available at https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=936827
- [176] Schierl A, Angerer A, Hoffmann A, Vistein M, Reif W (2016) On structure and distribution of software for mobile manipulators. *Informatics in Control, Automation and Robotics 12th International Conference, ICINCO 2015 Colmar, France, July 21-23, 2015 Revised Selected Papers*, eds Filipe J, Madani K, Gusikhin O, Sasiadek J (Springer International Publishing, Cham), pp 209–227.
- [177] Schierl A (2017) *Object-Oriented Modeling and Coordination of Mobile Robots*. Doctoral thesis Universität Augsburg, Universität Augsburg, Universitätsstr. 2, 86159 Augsburg. Available at <https://opus.bibliothek.uni-augsburg.de/opus4/>

[frontdoor/index/index/docId/3915](#) (Accessed July 17, 2023).

- [178] Obregón-Flores J, Arechavaleta G, Becerra HM, Morales-Díaz A (2021) Predefined-time robust hierarchical inverse dynamics on torque-controlled redundant manipulators. *IEEE Transactions on Robotics* 37(3):962–978. <https://doi.org/10.1109/TRO.2020.3042054>

Appendix A. List of Symbols, Abbreviations, and Acronyms

Acronyms

2D Two-Dimensional. 12, 19, 20, 52–54

3D Three-Dimensional. 12, 17, 20, 24, 27, 38

AGV Automatic Guided Vehicle. 3

AM Additive Manufacturing. 1

AMCL Adaptive Monte Carlo Localization. 9

AMR Autonomous Mobile Robot. 1, 3, 9, 12, 15, 19, 32, 43

AMR-CT Autonomous Mobile Robot Cart Transporter. iii–v, vii, viii, 3–9, 11–29, 31–38, 43, 48, 50–52, 54–57, 59, 67, 69, 72, 74, 77, 81, 83, 85–91, 107–109

ANOVA Analysis of Variance. 50, 67, 107, 108

API Application Programming Interface. 9, 12, 13, 51

ARIA Advanced Robotics Interface for Applications. 9

CMMA Configurable Mobile Manipulator Apparatus. iii, iv, vi, vii, 2–4, 8, 9, 12, 13, 15–18, 22, 43, 45, 46, 48–50, 52–57, 59–62, 67–69, 72, 74, 77, 81, 83, 85–88, 90, 91, 107–109

CNC Computer Numerical Control. 1

CSV Comma Separated Value. 17, 23, 24, 45, 52, 54

DOE design of experiments. vi, vii, 69, 73, 77, 80, 81, 85–87

DoF Degrees of Freedom. v, 14, 19, 20, 27, 29, 31–33

e-stop emergency stop. 12

EDA Exploratory Data Analysis. iii, viii, 67, 83, 85–88

EOAT End-of-Arm Tool. vi, vii, 2, 15, 45, 47, 52–54, 56, 89, 90

FOV field-of-view. 17

FPS Frames-per-Second. 17

GPU graphics processing unit. 5

JASP Jeffreys’s Amazing Statistics Program. 64, 107

MP Megapixel. 17

MPC Model-Predictive Control. 1

NIST National Institute of Standards and Technology. iv, vi, viii, 2, 4, 5, 8, 9, 11, 15, 16, 18–20, 32, 43, 55, 64, 91

NTP Network Time Protocol. vi, 18, 64

OS Operating System. 5, 15, 18, 56, 62, 63

OTS Optical Tracking System. iii–vi, 14, 15, 17–25, 27, 29, 31–34, 36–38, 43, 45, 46, 48, 49, 52, 54–57, 62, 63, 88

PC personal computer. 5, 15, 62, 63

pose position and orientation. viii, 12, 14, 20, 22–24, 27, 29

Q-Q quantile-quantile. vi, 64–66

RAM random access memory. 5, 17

RLS Retro-reflective Laser Sensor and Emitter. iv, 2, 13, 15, 16, 45, 52, 59

RMMA Re-configurable Mobile Manipulator Artifact. 2, 8, 54

ROS Robot Operating System. iv, vi, 5, 8–15, 18, 19, 32, 43, 44, 52–54, 56, 58

RPY roll-pitch-yaw. 12, 38

RTDE Real-Time Data Exchange. 15

SDK software development kit. 9

SRDF Semantic Robot Description Format. 43

TCP/IP Transmission Control Protocol / Internet Protocol. 12

UKF Unscented Kalman Filter. 3, 4, 18, 50, 53, 54, 57, 59–62, 68, 69, 77, 85, 87, 88, 91, 108, 109

ULAPI User-Level Application Programming Interface. 12

URDF Universal Robot Descriptor Format. 9, 32, 38, 43

xacro XML Macro. 43

Appendix B. Analysis of Variance

In this appendix, the results of conducting an ANOVA on the fiducial interception rates and fiducial search times are presented. Note that this analysis is included for completeness, but in light of the results of the model adequacy checks of Sec. 5.3, the conclusions derived from this analysis should be viewed cautiously. The ANOVA computations presented in Table 14 and 15 were computed using JASP (version 0.16.0.0) [148, 149].

Appendix B.1. Fiducial Interception Rate

The ANOVA testing hypotheses Eq. 5 - 11 on the fiducial interception rates are shown in Table 14. First, the second-order interaction effect of Eq. 5 is evaluated. Since the p-value was 0.645, which was greater than 0.05, the null hypothesis is not rejected and it would be concluded that there was not sufficient evidence to suggest that the interaction between search algorithm, AMR-CT speed, and measured CMMA side was significant. In this decision, it was possible that a Type II error was made, however the probability of this occurring was unknown.

Next, the first-order interaction effects of Eq. 6 - 8 were tested. Since the p-values were 0.408, 0.645, and 0.313, respectively, and all greater than 0.05, the null hypothesis was not rejected. Therefore it would be concluded that there was not sufficient evidence to suggest that any of the first-order interaction effects on the fiducial interception rate (i.e., the interaction between search algorithm and AMR-CT speed, the interaction between search algorithm and measured CMMA side, and the interaction between the AMR-CT speed and measured CMMA side) were significant. Again, in this decision, it was possible that a Type II error was made, however, the probability of this occurring was unknown.

Finally, we examine the effect of each individual factor. First, for the effect of search algorithm on fiducial interception rate, since the p-value was almost zero and less than 0.05, the null hypothesis was rejected. Therefore, it would be concluded that the effect of search algorithm on the fiducial interception rate was significant. In this conclusion, it is possible that a Type I error was made, and the probability of this occurring was almost 0. Next, for the effect of AMR-CT speed on fiducial interception time, since the p-value was 0.085 and greater than 0.05, the null hypothesis was not rejected. Therefore, it would be concluded that there was not sufficient evidence to suggest that the effect of AMR-CT

speed was significant. In this decision, it was possible that a Type II error was made, however, the probability of this occurring was unknown. Finally, for the effect of measured CMMA side on fiducial interception time, since the p-value was 0.645 and greater than 0.05, the null hypothesis was not rejected. Therefore, it would be concluded that there was not sufficient evidence to suggest that the effect of measured CMMA side was significant. In this decision, it was possible that a Type II error was made, however, the probability of this occurring was unknown.

Table 14. Analysis of Variance for comparing the fiducial interception rate.

Source of Variance	Sum of Squares ^{ab}	Degrees of Freedom	Mean Square ^a	\mathcal{F}_0	P-value ^c
Search Algorithm	1.505	1	1.505	22.461	2.707e-5
Side	0.014	1	0.014	0.216	0.645
Search Algorithm * Side	0.014	1	0.014	0.216	.645
Speed	0.209	1	0.209	3.117	0.085
Search Algorithm * Speed	0.047	1	0.047	0.699	0.408
Side * Speed	0.070	1	0.070	1.045	0.313
Search Algorithm * Side * Speed	0.014	1	0.014	0.216	0.645
Error	2.681	40	0.067		

^a Measured as fraction of detected fiducials square. ^b Type III Sum of Squares used.

^c A probability.

Since sufficient evidence was found to suggest that the effect of search algorithm was significant and the mean fiducial interception rate observed for the UKF was consistently lower than that of the deterministic spiral search, the deterministic spiral search was found to perform more accurately than the UKF search.

Appendix B.2. Fiducial Search Time

The ANOVA testing hypotheses Eq. 5 - 11 on the fiducial search times are shown in Table 15. First, the second-order interaction effect of Eq. 5 is evaluated. Since the p-value was 0.652, which was greater than 0.05, the null hypothesis is not rejected and it would be concluded that there was not sufficient evidence to suggest that the interaction between search algorithm, AMR-CT speed, and measured CMMA side was significant. In this decision, it was possible that a Type II error was made, however the probability of this occurring was unknown.

Next, the first-order interaction effects of Eq. 6 - 8 were tested. Since the p-values were 0.108, 0.669, and 0.596, respectively, and all greater than 0.05, the null hypothesis was not rejected. Therefore it would be concluded that there was not sufficient evidence to suggest that any of the first-order interaction effects on the fiducial search time (i.e., the interaction between search algorithm and AMR-CT speed, the interaction between search algorithm and measured CMMA side, and the interaction between the AMR-CT speed and measured

CMMA side) were significant. Again, in this decision, it was possible that a Type II error was made, however, the probability of this occurring was unknown.

Finally, we examine the effect of each individual factor. First, for the effect of search algorithm on fiducial search time, since the p-value was 0.014 and less than 0.05, the null hypothesis was rejected. Therefore, it would be concluded that the effect of search algorithm on the fiducial search time was significant. In this conclusion, it is possible that a Type I error was made, and the probability of this occurring was almost 0. Next, for the effect of AMR-CT speed on fiducial interception time, since the p-value was 0.111 and greater than 0.05, the null hypothesis was not rejected. Therefore, it would be concluded that there was not sufficient evidence to suggest that the effect of AMR-CT speed was significant. In this decision, it was possible that a Type II error was made, however, the probability of this occurring was unknown. Finally, for the effect of measured CMMA side on fiducial interception time, since the p-value was 0.615 and greater than 0.05, the null hypothesis was not rejected. Therefore, it would be concluded that there was not sufficient evidence to suggest that the effect of measured CMMA side was significant. In this decision, it was possible that a Type II error was made, however, the probability of this occurring was unknown.

Table 15. Analysis of Variance for comparing the fiducial search time.

Source of Variance	Sum of Squares ^{ab}	Degrees of Freedom	Mean Square ^a	\mathcal{F}_0	P-value ^c
Search Algorithm	15229.747	1	15229.747	6.617	0.014
Side	590.138	1	590.138	0.256	0.615
Speed	6142.768	1	6142.768	2.669	0.110
Search Algorithm * Side	427.954	1	427.954	0.186	0.669
Search Algorithm * Speed	6237.634	1	6237.634	2.710	0.108
Side * Speed	658.205	1	658.205	0.286	0.596
Search Algorithm * Side * Speed	475.701	1	475.701	0.207	0.652
Error	92058.235	40	2301.456		

^a Measured as seconds squared. ^b Type III Sum of Squares used. ^c A probability.

Since sufficient evidence was found to suggest that the effect of search algorithm was significant and the mean fiducial search time observed for the UKF was consistently higher than that of the deterministic spiral search, the deterministic spiral search was found to perform more accurately than the UKF search.