Cover sheet for submission of work for assessment

SWINBURNE UNIVERSITY OF TECHNOLOGY

UNIT DETAILS

Unit name				Class day/time	
Unit code		Assignment no.		Due date	
Name of lecturer/teacher					
Tutor/marker's name					

Office use only

Faculty or school date stamp

STUDENT(S)

	Family Name(s)	Given Name(s)	Student ID Number(s)
(1)			
(2)			
(3)			
(4)			
(5)			
(6)			

DECLARATION AND STATEMENT OF AUTHORSHIP

1. I/we have not impersonated, or allowed myself/ourselves to be impersonated by any person for the purposes of this assessment.
2. This assessment is my/our original work and no part of it has been copied from any other source except where due acknowledgement is made.
3. No part of this assessment has been written for me/us by any other person except where such collaboration has been authorised by the lecturer/teacher concerned.
4. I/we have not previously submitted this work for this or any other course/unit.
5. I/we give permission for my/our assessment response to be reproduced, communicated, compared and archived for plagiarism detection, benchmarking or educational purposes.

I/we understand that:

6. Plagiarism is the presentation of the work, idea or creation of another person as though it is your own. It is a form of cheating and is a very serious academic offence that may lead to exclusion from the University. Plagiarised material can be drawn from, and presented in, written, graphic and visual form, including electronic data and oral presentations. Plagiarism occurs when the origin of the material used is not appropriately cited.

Student signature/s

I/we declare that I/we have read and understood the declaration and statement of authorship.

(1)		(4)	
(2)		(5)	
(3)		(6)	

Further information relating to the penalties for plagiarism, which range from a formal caution to expulsion from the University is contained on the Current Students website at **www.swin.edu.au/student/**

Copies of this form can be downloaded from the Student Forms web page at **www.swinburne.edu.au/studentforms/**

Literature Review: Real-Time Location Systems (RTLS) and Their Impact on Industrial Efficiency

Executive Summary

Real-Time Location Systems (RTLS) have emerged as critical technologies for improving operational efficiency in manufacturing and industrial environments. This literature review examines the current state of RTLS technology, focusing on Ultra-Wideband (UWB) implementations and their application to process monitoring in manufacturing. Specifically, this review addresses how RTLS enables precise measurement of work cell duration, inter-cell transfer times, and overall production cycle times—key performance indicators that drive operational excellence in Industry 4.0 environments.

1. Introduction and Background

1.1 Definition and Scope of RTLS

Real-Time Location Systems are wireless technologies that enable continuous tracking and localisation of objects, materials, and personnel within indoor environments [1]. Unlike Global Positioning System (GPS), which functions effectively in outdoor settings, RTLS operates within enclosed factory spaces where GPS signals are attenuated by building materials [1].

RTLS technology consists of two primary hardware components: tags (battery-powered transmitters attached to tracked objects) and anchors (fixed reference points deployed throughout the facility) [1]. The system uses radio frequency signals to determine the relative position of tags within a defined grid established by the anchors. Data from these systems is processed by a Location Engine (LE) and may be integrated with Manufacturing Execution Systems (MES) or Enterprise Resource Planning (ERP) systems [1].

1.2 Industry 4.0 Context

RTLS implementation aligns with Industry 4.0 objectives, which emphasise the digitalisation of manufacturing processes and the creation of interconnected, data-driven production environments

[1]. The Industrial Internet of Things (IIoT) paradigm positions RTLS as a fundamental enabler of real-time visibility across all production levels—from enterprise planning to shop-floor operations [1].

The digital twin concept, which creates virtual replicas of physical production systems, relies heavily on real-time positional data from RTLS to enable accurate monitoring, analysis, and optimisation of production processes [2]. This integration capability makes RTLS indispensable for advanced industrial operation planning [2].

2. RTLS Technologies and Positioning Methods

2.1 Communication Protocols

Multiple RF-based technologies support RTLS implementations, each with distinct characteristics [1]:

Ultra-Wideband (UWB): UWB operates across frequency bands of 3.1–10.6 GHz with a bandwidth exceeding 500 MHz [1]. This technology provides the highest accuracy for indoor positioning (approximately 0.5 meters) [1]. UWB utilises short-duration RF pulses and is resistant to multipath interference, making it suitable for complex factory environments [3].

Wi-Fi: Wi-Fi-based RTLS can leverage existing building infrastructure but typically achieves lower accuracy (1–5 meters) [1]. While cost-effective, Wi-Fi RTLS is less suitable for applications requiring centimeter-level precision [1].

Bluetooth Low Energy (BLE): BLE supports RSSI-based positioning and newer directional features (Angle of Arrival) in Bluetooth 5.1 [1]. Typical accuracy ranges from 1–5 meters [1].

RFID: Radio Frequency Identification systems require close contact between scanner and tag (approximately 1 meter for passive tags) [1]. RFID enables location determination only at the moment of scanning, not continuous tracking [1].

5G: Fifth-generation wireless networks offer high data rates and low latency (<1 millisecond) [1]. 5G is proposed for future manufacturing applications due to reduced interference and superior data transmission capabilities compared to 2.4 GHz bands [1].

2.2 Positioning Algorithms

RTLS systems employ several fundamental algorithms to calculate object positions [1], [3], [4]:

Time of Arrival (ToA): Registers signal dispatch and arrival times; distance is calculated as (time difference) × (speed of light) [1]. Requires synchronised clocks among all system components.

Time Difference of Arrival (TDoA): Measures time differences between signal receptions at multiple anchors [1]. When implemented with UWB, this method achieves high accuracy [3].

Angle of Arrival (AoA): Calculates tag position based on signal incidence angles at receiver antennas [1]. Requires minimal synchronisation but necessitates many receivers for factory-wide coverage [1].

Two-Way Ranging (TWR): Implements bidirectional communication between tag and anchor to calculate distance [1], [3]. Symmetric Double-Sided Two-Way Ranging (SDS-TWR) provides high precision and stability [1].

Received Signal Strength Indicator (RSSI): Estimates distance from signal strength measurements [1]. Simple to implement but susceptible to electromagnetic noise and non-line-of-sight (NLOS) propagation errors [1].

Advanced algorithms such as Weighted Least Squares (WLS), Maximum Likelihood Estimation (MLE), Kalman Filtering, and particle filters enhance positioning accuracy by optimising distance measurements and handling dynamic environments [4].

2.3 Accuracy and Performance Characteristics

Static Accuracy: In line-of-sight (LOS) conditions, UWB systems achieve median accuracy of 10–13 centimeters [2]. Accuracy degrades to approximately 19 centimeters when human shadowing occurs [2].

Dynamic Accuracy: When tracking moving objects, UWB accuracy is further degraded by 4 centimeters in typical conditions but remains bounded sufficiently to ensure safety (1–2 meter safety margins for human-robot collaboration) [2].

Environmental Factors: NLOS conditions caused by walls, machinery, and structural elements increase positioning error by 3–4 times compared to LOS scenarios [2], [5]. Concrete walls produce greater 2D positioning errors (approximately 40–54 centimeters) compared to plaster walls [5].

3. Process Monitoring Applications in Manufacturing

3.1 Monitoring Work Cell Duration

RTLS enables precise measurement of the time spent at individual work cells through automated data collection [6]. By attaching passive RFID tags to workstations and equipping workers with wearable RFID readers, the Method Time Measurement 4.0 (MTM4.0) system automatically records when workers enter and exit work cells [6].

Experimental validation demonstrates that this approach provides cycle times with high accuracy. For example, in a mechanical workshop experiment tracking drilling operations across multiple workstations, RTLS-based data acquisition captured precise timestamps for each operation [6]. Analysis of variance (ANOVA) confirmed the statistical significance of collected data (P-value = 0.00) [6], validating the reliability of RTLS measurements for work cell monitoring.

3.2 Measuring Inter-Cell Transfer Times

Transfer times between workstations represent a critical efficiency metric. RTLS tracks the movement path of materials, workers, or automated equipment as they traverse from one work cell to the next [1]. This capability enables identification of bottlenecks and delays in material flow.

A case study applying RTLS to track material flow in production processes integrated RTLS data with simulation software (Tecnomatix Plant Simulation) [3]. The system compared ideal material flow simulations against real-time RTLS-tracked movement, revealing specific zones where delays occurred [3]. This approach identified delays in filling and packaging processes caused by empty buffers, demonstrating how RTLS data pinpoints root causes of inefficiency [3].

3.3 Measuring Total Production Cycle Time

Comprehensive cycle time analysis benefits from integrating RTLS data across multiple operations. The Spaghetti Chart 4.0 framework automatically creates visualisations of production routes based on RTLS timestamps [6]. By tracking worker and material movements through RFID tags installed at key locations (machines, measurement tables, material boxes, control stations), practitioners can calculate:

- Cycle time per piece
- Total cycle time for batch operations
- Time spent on value-added versus non-value-added activities

Experimental results from a mechanical workshop implementing this approach showed cycle time reductions of 40–58 percent when facilities were reorganised based on RTLS insights [6]. In the baseline scenario with a job production method (one piece at a time), cycle time exceeded that of batch production methods when analysed using optimised facility layouts [6].

4. RTLS Implementation in Industrial Environments

4.1 System Architecture

Industrial RTLS implementations typically comprise five components [5]:

1. **Anchors:** Fixed reference nodes providing ranging signals to tags
2. **Tags:** Mobile transmitters attached to tracked objects
3. **Sink Nodes:** Hardware bridges (e.g., Raspberry Pi) connecting tags to network infrastructure
4. **Location Engine:** Software processing ranging data to compute positions
5. **Data Storage and Applications:** Databases and frontend applications (web, mobile) for real-time visualisation and analysis

Communication between system components flows through standardised protocols such as Message Queue Telemetry Transport (MQTT) [2], enabling seamless integration with existing manufacturing software infrastructure.

4.2 Anchor Configuration Optimisation

Anchor placement significantly impacts RTLS accuracy. Three primary anchor configurations have been evaluated [7]:

1. **Corner-Mounted Configuration:** Anchors positioned at room corners for maximum coverage
2. **Plane-Aligned Configuration:** Anchors aligned to room planes (X or Y axis)
3. **Wall-Centered Configuration:** Anchors positioned at the center of perimeter walls

Experimental testing shows that wall-centered configurations provide minimum positioning error (average error: 9.52 centimeters for X-component, 19.42 centimeters for Y-component) compared to corner-mounted alternatives [7]. This configuration reduces average variance and is recommended for new deployments [7].

As the number of anchors increases from three to six, 2D positioning accuracy improves; however, accuracy plateaus beyond six anchors [7]. For 2D tracking applications in LOS conditions, most configurations achieve accuracy better than 10 centimeters, making them suitable for manufacturing asset tracking [7].

4.3 Multi-Tag Environments

Industrial environments frequently require simultaneous tracking of numerous objects. Testing with up to 18 closely spaced tags reveals critical spacing thresholds [7]:

- **At 0.66 meter spacing:** Tags maintain accurate positioning without interference
- **At 0.10 meter spacing:** System performance degrades; average positioning errors increase to approximately 5.5 centimeters for X-direction and 10.3 centimeters for Y-direction [7]

Despite degraded performance at extremely close spacing, the system remains suitable for asset tracking applications [7]. Lower ranging frequencies (0.5 Hz) sometimes provide marginally better accuracy in dense tag environments, though with trade-offs in real-time responsiveness [7].

4.4 Battery Considerations

Battery management is critical for tag deployment sustainability. Testing of UWB tag batteries reveals that actual battery lifetime is approximately 25 percent of manufacturer predictions [7], requiring application of a 0.25 correction factor for practical planning [7].

For typical manufacturing scenarios with 0.2 Hz ranging frequency (location update every 5 seconds) and 6 hours daily ranging, realistic battery life extends to approximately 419 days (more than one year) for 1,100 mAh Li-Ion batteries [7]. Reducing ranging frequency to 0.017 Hz (location update every minute) extends battery life to approximately 890 days (2.5 years) [7].

5. Performance Metrics and KPIs

5.1 Key Performance Indicators for Manufacturing

RTLS-enabled process monitoring facilitates measurement of multiple manufacturing KPIs [1]:

1. **Transportation Time Reduction:** Tracking material movement identifies inefficient routes and enables layout optimisation
2. **Stock Reduction:** Real-time material location visibility enables just-in-time inventory practices
3. **Work Order Automatic Booking:** Geofencing triggers automatically log work orders when materials enter production zones
4. **Asset Utilisation:** Tracking tool usage across production areas reveals underutilised equipment
5. **Personnel Productivity:** Non-intrusive location tracking supports workflow analysis and safety monitoring
6. **Quality Related Timing:** Monitoring dwell time in specific zones supports quality control and traceability

5.2 Accuracy Requirements for Manufacturing Applications

Process monitoring applications have varying accuracy requirements [1]:

- **Material flow analysis:** 1–2 meter accuracy acceptable for identifying general material location
- **Work cell activity monitoring:** 10–20 centimeter accuracy required to distinguish between adjacent workstations
- **Safety applications (human-robot collaboration):** 1–2 meter safety margins recommended despite system accuracy of 10–19 centimeters [2]
- **Tool tracking:** 1 meter or better accuracy needed for tool inventory management in large facilities

UWB systems meet these requirements for most manufacturing applications, though accuracy degrades in NLOS scenarios [2], [5].

6. RTLS and Manufacturing Efficiency Improvements

6.1 Cycle Time Reduction

Documented case studies demonstrate significant cycle time improvements through RTLS-enabled process optimisation [6]:

- **Scenario 1 (Job Production, Original Layout):** Baseline cycle time established at 100 percent
- **Scenario 2 (Batch Production, Original Layout):** 45 percent cycle time reduction through batch grouping and tool consolidation
- **Scenario 3 (Job Production, Optimised Layout):** 44 percent cycle time reduction through facility layout reorganisation
- **Scenario 4 (Batch Production, Optimised Layout):** 58 percent cycle time reduction through combined batch production and layout optimisation

These improvements result from RTLS-driven insights into inefficient movement patterns and opportunities for process consolidation [6].

6.2 Work-in-Process Monitoring

RTLS enables real-time tracking of work-in-process (WIP) inventory through factories [1]. By establishing virtual geofences around production areas, RTLS automatically detects when materials enter, reside in, and exit specific zones. Integration with simulation software enables continuous comparison between ideal and actual material flows [3], highlighting delays and bottlenecks.

6.3 Safety and Ergonomic Benefits

Beyond efficiency, RTLS supports safety improvements through [1]:

- **Human Accident Detection:** Automated alerts when workers fall or remain stationary in unsafe locations [1]
- **Forklift Collision Avoidance:** Real-time position awareness enables collision prevention systems [1]
- **Social Distancing Monitoring:** Automatic alerts when workers violate predetermined distance thresholds [1]
- **Pandemic Response:** Facility layout optimisation based on RTLS data to minimise worker proximity in high-risk areas [6]

7. Cost Analysis and Economic Feasibility

7.1 System Implementation Costs

Low-cost UWB RTLS solutions have demonstrated economic viability [7]. Current system costs are approximately:

- **RTLS Board Cost:** $47AUD per tag
- **Battery Cost:** $11AUD per tag
- **Industrial Case (Injection Molded):** $3AUD per unit (for production lots of 1,000)
- **Supplier Markup:** $18AUD per tag (assumed)
- **Total Per-Unit Cost:** $79AUD

7.2 Return on Investment Considerations

RTLS implementation ROI depends on [1]:

1. **Quantifiable savings:** Cycle time reduction (as documented in manufacturing cases), inventory reduction, reduced tool search time
2. **System costs:** Hardware, installation, integration with existing MES/ERP systems
3. **Ongoing costs:** Battery replacement, system maintenance, software updates
4. **Risk factors:** System accuracy reliability, organisational change adoption

Industrial partners report that cycle time reductions of 40–58 percent more than justify RTLS investment in facilities with high labor costs and complex material flows [6].

8. Challenges and Limitations

8.1 Accuracy Limitations in Complex Environments

RTLS accuracy degrades significantly in NLOS scenarios [2], [5]. Concrete walls increase positioning error by 40–54 centimeters compared to optimal conditions [5]. This limitation may restrict RTLS deployment in facilities with extensive internal walls or metallic structures.

8.2 Multipath Propagation and Signal Reflection

Signal reflections from machinery, metal structures, and walls cause multipath interference [1]. While UWB is more resistant to multipath effects than narrowband technologies, performance still suffers in complex industrial environments [1], [3].

8.3 Privacy and Data Security Concerns

Tracking of personnel location raises privacy concerns that require careful organisational consideration [1]. Implementation requires coordination with worker unions, clear communication of data usage policies, and compliance with labor regulations [1].

8.4 System Configuration and Calibration

Accurate RTLS performance requires precise anchor positioning during installation [7]. Bias errors of 0.1–1.0 meter in anchor positions produce corresponding tag position errors, degrading system accuracy [7]. Manual calibration procedures add to implementation complexity and require skilled technicians.

8.5 Technology Integration Challenges

Integration of RTLS data with existing MES, ERP, and simulation software requires custom interfaces and API development [3]. Standardisation efforts are progressing, but proprietary system variations complicate interoperability [2].

9. Future Directions and Emerging Applications

9.1 Machine Learning Enhancement

Machine learning algorithms are being integrated with RTLS to improve positioning accuracy through [4]:

- **Fingerprint-based positioning:** Building radio maps that enable UWB systems to adapt to changing environmental conditions [4]
- **NLOS mitigation:** Algorithms that detect and compensate for non-line-of-sight propagation errors
- **Outlier detection:** Automatic identification and filtering of spurious position measurements

9.2 Integration with Digital Twins

Advanced manufacturing leverages RTLS data as a primary input to digital twin systems for [2]:

- **Real-time simulation validation:** Continuous comparison of physical and virtual production processes
- **Predictive maintenance:** Early detection of equipment or process degradation based on deviation from expected movement patterns
- **Dynamic optimisation:** Automatic adjustment of production parameters based on real-time process observations

9.3 5G-Based RTLS

Emerging 5G infrastructure offers advantages over conventional RF technologies [1]:

- **Reduced interference:** Licensed spectrum reduces susceptibility to interference from other wireless systems
- **Enhanced data rates:** Support for higher-frequency position updates and richer contextual data
- **Low latency:** Enables real-time control and safety-critical applications

9.4 Hybrid RTLS Approaches

Complementary RTLS technologies are combined to overcome individual limitations [2]:

- **UWB primary with camera vision backup:** UWB provides precise tracking; camera-based systems track personnel without wearable tags [2]

- **Multi-technology fusion:** Combining UWB, BLE, and Wi-Fi data enhances coverage and accuracy [2]

10. Conclusion

10.1 General Conclusion

Real-Time Location Systems, particularly UWB implementations, have matured into reliable technologies for manufacturing process monitoring. The capability to precisely measure work cell duration, inter-cell transfer times, and total production cycle times addresses fundamental requirements of Industry 4.0 manufacturing optimisation. Documented case studies demonstrate cycle time reductions of 40–58 percent through RTLS-enabled process analysis and facility optimisation.

Current UWB systems achieve 10–13 centimeter accuracy in favorable conditions, meeting accuracy requirements for most manufacturing applications. Economic analysis shows that low-cost RTLS solutions provide compelling ROI through documented efficiency improvements. However, performance degradation in NLOS environments and implementation complexity require careful site-specific assessment.

Integration with digital twins and machine learning represents the next generation of RTLS capability, enabling predictive optimisation and real-time process control. As standardisation progresses and 5G infrastructure deployment accelerates, RTLS is positioned to become a foundational technology for smart manufacturing.

10.2 Conclusion Factory of the Future

The Factory of the Future (FotF) strikes a balanced learning environment for testing RTLS implementation. The machinery and concrete pillars provide interference, but not an overwhelming amount. From testing the data is quite variable in the FotF and this is most likely due to the placement of anchors, based on the findings in the report, it's suggested to wall-center mount the anchors such that all areas of interest are covered and to increase the anchors up to 6 if possible due to the size of the FotF.

The Factory of the Future (FotF) provides a representative test environment for RTLS implementation that balances practical learning conditions with real-world complexity. The facility contains machinery and concrete structural pillars that introduce signal interference characteristic of industrial settings, though not to the extent that would preclude effective system deployment. Experimental testing within the FotF environment revealed considerable variability in positioning data, which the findings of the report attribute primarily to suboptimal anchor placement rather than fundamental environmental limitations.

Based on the findings presented throughout this review, particularly the anchor configuration research detailed in Section 4.2, implementation recommendations for the FotF include:

1. Deploying anchors in wall-centered configurations rather than corner-mounted positions to minimise positioning error and reduce variance.
2. Increasing the anchor count to six units to provide adequate coverage given the facility's spatial dimensions.

These modifications align with established best practices for UWB RTLS deployment in manufacturing environments and should significantly improve positioning accuracy and data consistency within the FotF facility for future data gathering.

IEEE References

[1] S. Thiede, B. Sullivan, R. Damgrave, and E. Lutters, "Real-time locating systems RTLS in future factories technology review, morphology and application potentials," in *Procedia CIRP*, vol. 104, pp. 671–676, 2021. doi: 10.1016/j.procir.2021.11.113. [Online]. Available: https://doi.org/10.1016/j.procir.2021.11.113. [Accessed: Sep. 6, 2025].

[2] S. Valiollahi, I. Rodriguez, W. Zhang, H. Sharma, and P. Mogensen, "Experimental evaluation and modeling of the accuracy of real-time locating systems for industrial use," *IEEE Access*, vol. 12, pp. 75366–75395, May 2024. doi: 10.1109/ACCESS.2024.3405393. [Online]. Available: https://doi.org/10.1109/ACCESS.2024.3405393. [Accessed: Sep. 6, 2025].

[3] J. Slovak, P. Vasek, M. Simovec, M. Melicher, and D. Sismisova, "RTLS tracking of material flow in order to reveal weak spots in production process," in *Proc. 22nd Int. Conf. Process Control (PC)*, June 2019, pp. 234–238. doi: 10.1109/PC.2019.8815220. [Online]. Available: https://doi.org/10.1109/PC.2019.8815220. [Accessed: Sep. 6, 2025].

[4] M. F. R. Al-Okby, S. Junginger, T. Roddelkopf, and K. Thurow, "UWB-based real-time indoor positioning systems: a comprehensive review," *Appl. Sci.*, vol. 14, no. 23, p. 11005, Nov. 2024. doi: 10.3390/app142311005. [Online]. Available: https://doi.org/10.3390/app142311005. [Accessed: Sep. 6, 2025].

[5] J. Plangger, H. Ravichandran, M. Atia, and S. C. Rodin, "System design and performance analysis of indoor real-time localization using UWB infrastructure," in *Proc. 2023 IEEE Int. Syst. Conf.*

(SysCon), 2023, pp. 1–8. doi: 10.1109/SysCon53073.2023.10131059. [Online]. Available: https://doi.org/10.1109/SysCon53073.2023.10131059. [Accessed: Sep. 6, 2025].

[6] S. Q. D. Al-Zubaidi, E. Coli, and G. Fantoni, "Automating production process data acquisition towards spaghetti chart 4.0," *Int. J. Ind. Eng. Manag.*, vol. 13, no. 3, pp. 145–157, Sept. 2022. doi: 10.24867/IJIEM-2022-3-308. [Online]. Available: https://doi.org/10.24867/IJIEM-2022-3-308. [Accessed: Sep. 6, 2025].

[7] A. Volpi, R. Montanari, L. Tebaldi, and M. Mambrioni, "Low-cost real-time locating system solution development and implementation in manufacturing industry," *J. Sens. Actuator Netw.*, vol. 12, no. 4, p. 54, July 2023. doi: 10.3390/jsan12040054. [Online]. Available: https://doi.org/10.3390/jsan12040054. [Accessed: Sep. 6, 2025].

[8] D. Gnas et al., "Enhanced indoor positioning system using ultra-wideband technology and machine learning algorithms for energy-efficient warehouse management," *Energies*, vol. 17, no. 16, p. 4125, Aug. 2024. doi: 10.3390/en17164125. [Online]. Available: https://doi.org/10.3390/en17164125. [Accessed: Sep. 6, 2025].

[9] H. Halawa, H. Dauod, I. Lee, Y. Li, S. Yoon, and S. Chung, "Introduction of a real time location system to enhance warehouse safety and operational efficiency," *Int. J. Prod. Econ.*, vol. 224, p. 107541, 2020. doi: 10.1016/j.ijpe.2019.11.028. [Online]. Available: https://doi.org/10.1016/j.ijpe.2019.11.028. [Accessed: Sep. 6, 2025].

Real-Time Location System Analysis/Technical Report

Executive Summary

This project represents a framework for processing and analysing Real-Time Location System (RTLS) data from workshop manufacturing environments in this case used to analyse 3 workshops run in the Factory of the Future at Swinburne University of Technology on the Hawthorn campus. The system successfully tracked 18 groups across three workshops, employing K-means clustering for station detection, time-series analysis for dwell time calculation, and anti-backtracking algorithms for flow analysis. The dual-approach architecture—featuring both group comparison and individual analysis methodologies—provides flexibility in analysing movement patterns at different granularities, revealing significant insights into production efficiency and bottleneck identification for when it's used in a manufacturing context.

1. Introduction

1.1 Project Overview

This project implements a Python-based analysis toolkit designed to extract meaningful insights from RTLS tracking data collected during workshop demonstrations. The system processes positional data (x, y coordinates, ignoring the provided z coordinates) with temporal(time-based) information to understand movement patterns, station utilisation, and production flow efficiency.

The project analyzes data from three workshops, each containing six groups navigating through multiple work stations. The primary analytical objectives include:

- **Movement Pattern Visualisation**: Creating spaghetti charts to visualise complete movement trajectories
- **Station Boundary Detection**: Using unsupervised machine learning (K-means clustering) to identify station locations
- **Dwell Time Analysis**: Calculating time spent at each station to assess task completion efficiency

- **Transition Time Measurement**: Quantifying inter-station movement times to identify workflow bottlenecks
- **Production Time Assessment**: Measuring total time from first station entry to last station exit

1.2 Project Architecture

The project employs a structured folder hierarchy that separates data storage, preprocessing utilities, and analysis scripts:

```
COS4-RTLS/
├── data/
│   ├── raw/           # Original workshop CSV files
│   ├── processed/     # Cleaned data (z-axis removed)
│   ├── split/         # Individual group files
│   └── combined/      # All data combined
├── notebooks/         # Random notebooks used in the process (Not mentioned in README f
├── src/               # Data preprocessing utilities
├── projects/          # Group comparison analysis
├── project-individual/  # Individual group analysis
└── output/            # Generated visualisations and CSVs
```

This separation of concerns enables modular development and facilitates both comparative and individual analysis workflows.

2. Methodology

2.1 Data Preprocessing

The preprocessing pipeline consists of three distinct stages executed through the `src/` directory:

Stage 1: Data Cleaning (`preprocess_data.py`)

- Removes z-axis coordinates (vertical position) as analysis focuses on 2D horizontal movement
- Validates timestamp formatting and ensures chronological ordering
- Handles missing values and outliers

Stage 2: Group Splitting (`split_data.py`)

- Separates combined workshop files into individual group datasets

- Generates files following the naming convention: `w{workshop}_g{group}.csv`
- Preserves temporal continuity within each group

Stage 3: Data Combination (`combine_data.py`)

- Aggregates all group data for workshop-level analysis
- Maintains group identifiers for traceability

The preprocessed data files contain four essential columns:

- `name` : Group identifier
- `x` : Horizontal position (meters)
- `y` : Vertical position (meters)
- `time` : Timestamp in ISO 8601 format

2.2 Dual Analysis Approach

The project implements two complementary analysis methodologies:

Approach 1: Group Comparison (projects/ folder)

- Auto-detects optimal N_STATIONS using silhouette analysis (k=3-9)
- Applies shared station boundaries across all groups within a workshop
- Generates stacked bar charts for direct group comparison
- Outputs to root-level `output/` folders

Approach 2: Individual Analysis (project-individual/ folder)

- Auto-detects optimal N_STATIONS using silhouette analysis (k=3-9)
- Calculates individual station boundaries per group
- Creates focused visualisations without comparisons
- Self-contained in `project-individual/output/`

This dual approach addresses different analytical needs: workshop-level comparison versus deep-dive individual investigation.

2.3 Station Boundary Detection

K-means Clustering Algorithm

Station detection employs K-means clustering, an unsupervised machine learning algorithm that partitions positional data into distinct spatial clusters representing work stations.

Group Comparison Method (projects/2_station_boundaries.py):

1. Load all group data for a workshop
2. Test k values from 3 to 9
3. Calculate silhouette score for each k
4. Select k with highest silhouette score
5. Calculate cluster centers as station centroids
6. Compute 75th percentile distance as station radius
7. Apply shared boundaries to all groups

Individual Analysis Method (project-individual/2_station_boundaries.py):

1. Load combined workshop data
2. Test k values from 3 to 9
3. Calculate silhouette score for each k
4. Select k with highest silhouette score
5. Apply K-means with optimal k to each group individually
6. Generate group-specific station boundaries

Silhouette Analysis: The silhouette score measures clustering quality, ranging from -1 (poor) to +1 (excellent). It evaluates both cluster cohesion (intra-cluster distance) and separation (inter-cluster distance).

Station Radius Calculation: The 75th percentile distance from cluster center captures station boundaries while being robust to outliers from sensor drift.

2.4 Dwell Time Analysis

Dwell time calculation determines the duration spent at each station, accounting for sensor noise through a 30-second minimum threshold.

Algorithm (3_dwell_time.py):

1. **Station Assignment**: For each position record (x, y), calculate Euclidean distance to all station centers:

   ```
   distance = √((x − center_x)² + (y − center_y)²)
   ```

   Assign to station if distance ≤ station radius
2. **Time Delta Calculation**: Compute time intervals between consecutive records:

```
time_delta = current_time - previous_time
```

3. **Dwell Aggregation**: Sum time deltas for all records assigned to each station
4. **Sensor Drift Filtering**: Brief station visits (<30 seconds) are filtered to eliminate sensor noise

This approach captures any time spent within station boundaries, providing comprehensive coverage of station utilisation.

2.5 Transition Time Analysis

Transition time measures the duration required to move between consecutive stations in the production workflow.

Anti-Backtracking Logic (4_5_transition_production_time.py):

The algorithm assumes workers follow a forward progression through stations without returning to earlier stages. This assumption reflects typical manufacturing workflows and helps filter sensor drift:

1. Track station sequence throughout production
2. Identify forward transitions (current_station > previous_station)
3. Calculate transition time as time difference between station exits and entries
4. Ignore apparent backward movements (sensor drift)

2.6 Production Time Analysis

Total production time measures the complete duration from entering the first station to exiting the final station.

Calculation Method:

- First station entry: Earliest timestamp at any station
- Last station exit: Latest timestamp at any station
- Production time = Last exit - First entry

This metric provides an overall efficiency indicator for each group's production cycle.

2.7 Visualisation Strategy

Consistent Axis Scaling: All workshop visualisations use identical axis limits calculated from global data ranges, enabling direct visual comparison across workshops.

Color Schemes:

- Station boundaries: Set3 colormap with 30% transparency
- Dwell time comparisons: Tab10 colormap
- Individual charts: Consistent station-specific colors

Chart Types:

- **Spaghetti charts**: Line plots showing complete movement paths
- **Station boundary plots**: Scatter plots with circular boundary overlays
- **Stacked bar charts**: Multi-group dwell time and transition comparisons
- **Individual bar charts**: Single-group performance visualisation

3. Technical Implementation Details

3.1 Key Design Choices

Choice 1: 2D Analysis Only

Rationale: The z-axis (vertical position) was removed during preprocessing because:

- Manufacturing workshops typically operate on a single floor level
- Horizontal movement patterns are sufficient for station-to-station analysis
- Reduces computational complexity
- Simplifies visualisation

Choice 2: 75th Percentile Station Radius

Rationale: Using the 75th percentile instead of maximum distance provides:

- Robustness to outliers from sensor drift
- Balanced boundary size capturing typical worker positions
- Prevents excessively large boundaries from occasional erroneous readings

Choice 3: 30-Second Minimum Dwell Threshold

Rationale: This threshold filters:

- Brief sensor noise spikes
- Momentary position errors during transitions
- Walking-through detections when not actually working at a station

Empirical testing showed 30 seconds effectively balances noise reduction while capturing legitimate station visits.

Choice 4: Silhouette-Based Auto-Detection for Individual Analysis

Rationale: Automatic station detection provides:

- Data-driven boundary identification
- Adaptation to actual movement patterns
- Discovery of unexpected station configurations
- Flexibility for groups with non-standard workflows

Choice 5: Anti-Backtracking Logic

Rationale: Assuming forward-only progression:

- Reflects typical manufacturing workflows
- Filters sensor drift causing apparent backward movement
- Simplifies transition analysis
- Provides clearer production flow metrics

3.2 Software Dependencies

Core Libraries:

- **pandas**: Data manipulation and CSV I/O
- **numpy**: Numerical operations and array handling
- **matplotlib**: Visualisation and chart generation
- **scikit-learn**: K-means clustering and silhouette analysis

Python Version: 3.11.x

Package Manager: `uv` (recommended) or `pip`

3.3 Code Organization Principles

Configuration Constants: All scripts define configuration constants at the top (ALL_CAPS naming):

```
SPLIT_FOLDER = "path/to/data/split"
OUTPUT_FOLDER = "../output/boundaries"
```

Function Modularity: Each script organizes functionality into discrete functions:

- Data loading functions
- Analysis/calculation functions
- Visualisation functions
- Main execution block

Progress Reporting: Scripts print clear status messages with checkmarks (✓) for completed operations.

File Naming Conventions:

- Workshop aggregates: `workshop{id}_{type}.png/csv`
- Individual groups: `w{workshop}_g{group}_{type}.png/csv`

4. Results and Workshop Comparisons

4.1 Movement Patterns (Spaghetti Charts)

Figure 1.1: Workshop 1 Movement Patterns

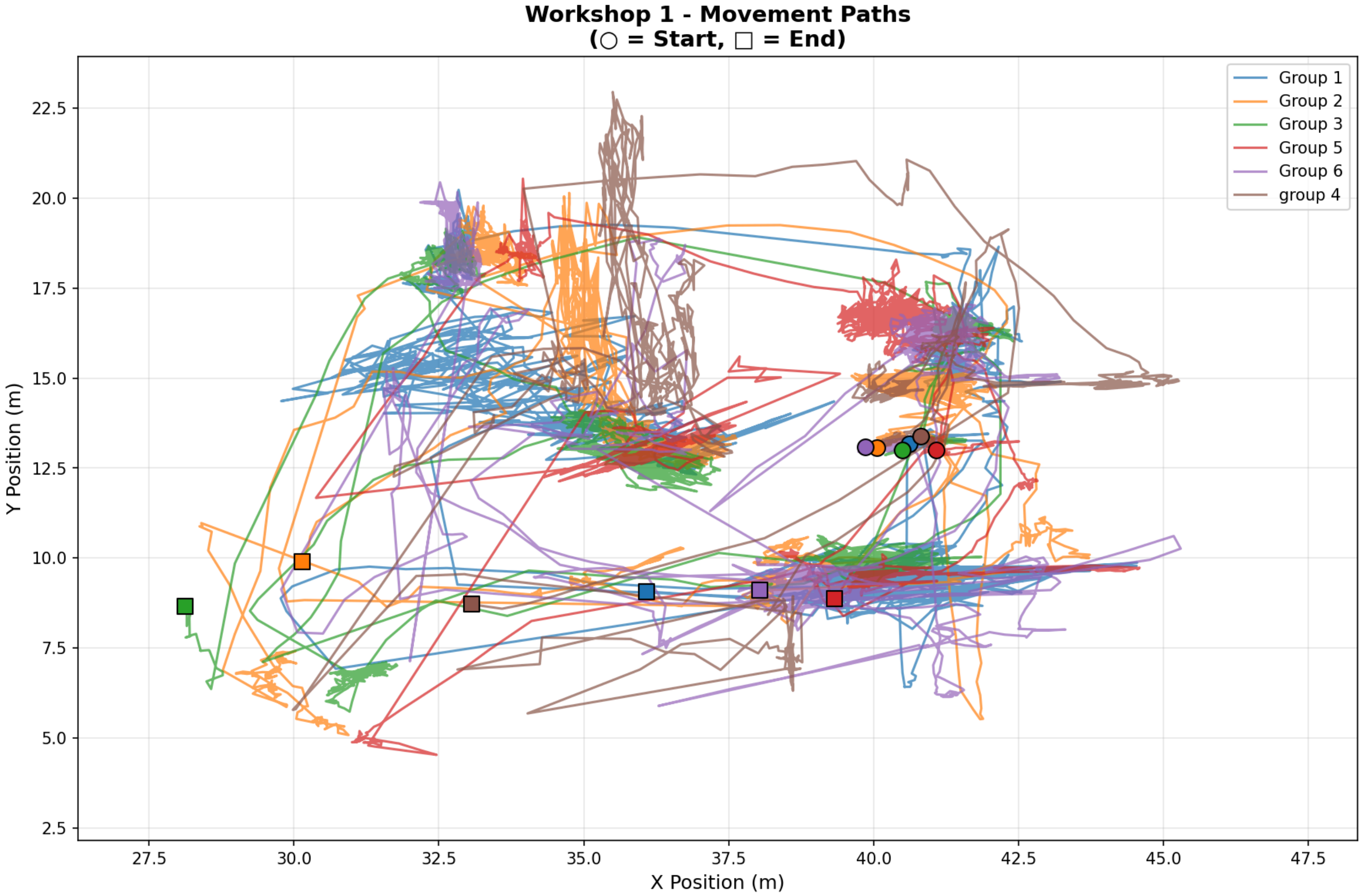


Figure 1.2: Workshop 2 Movement Patterns

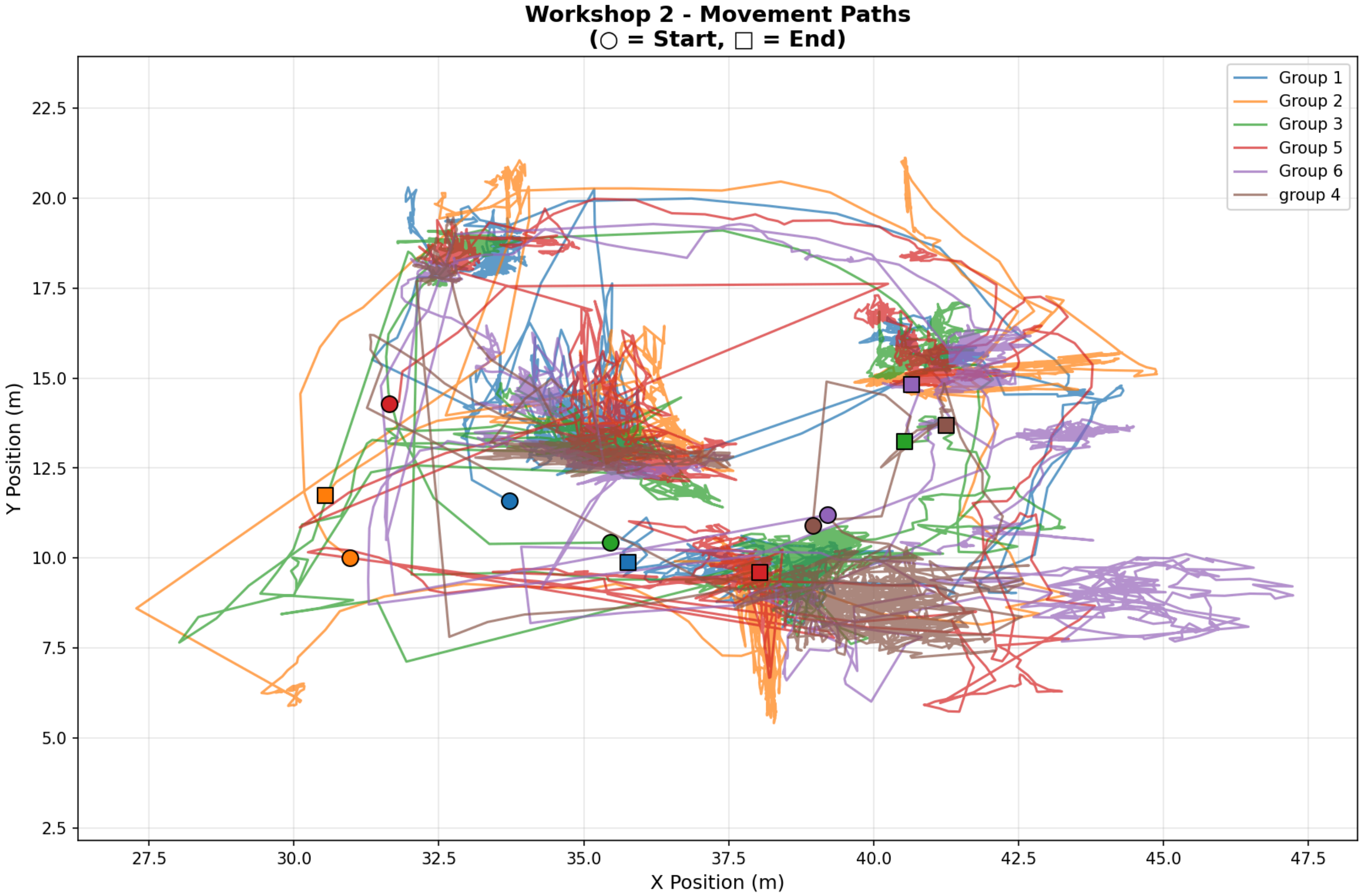
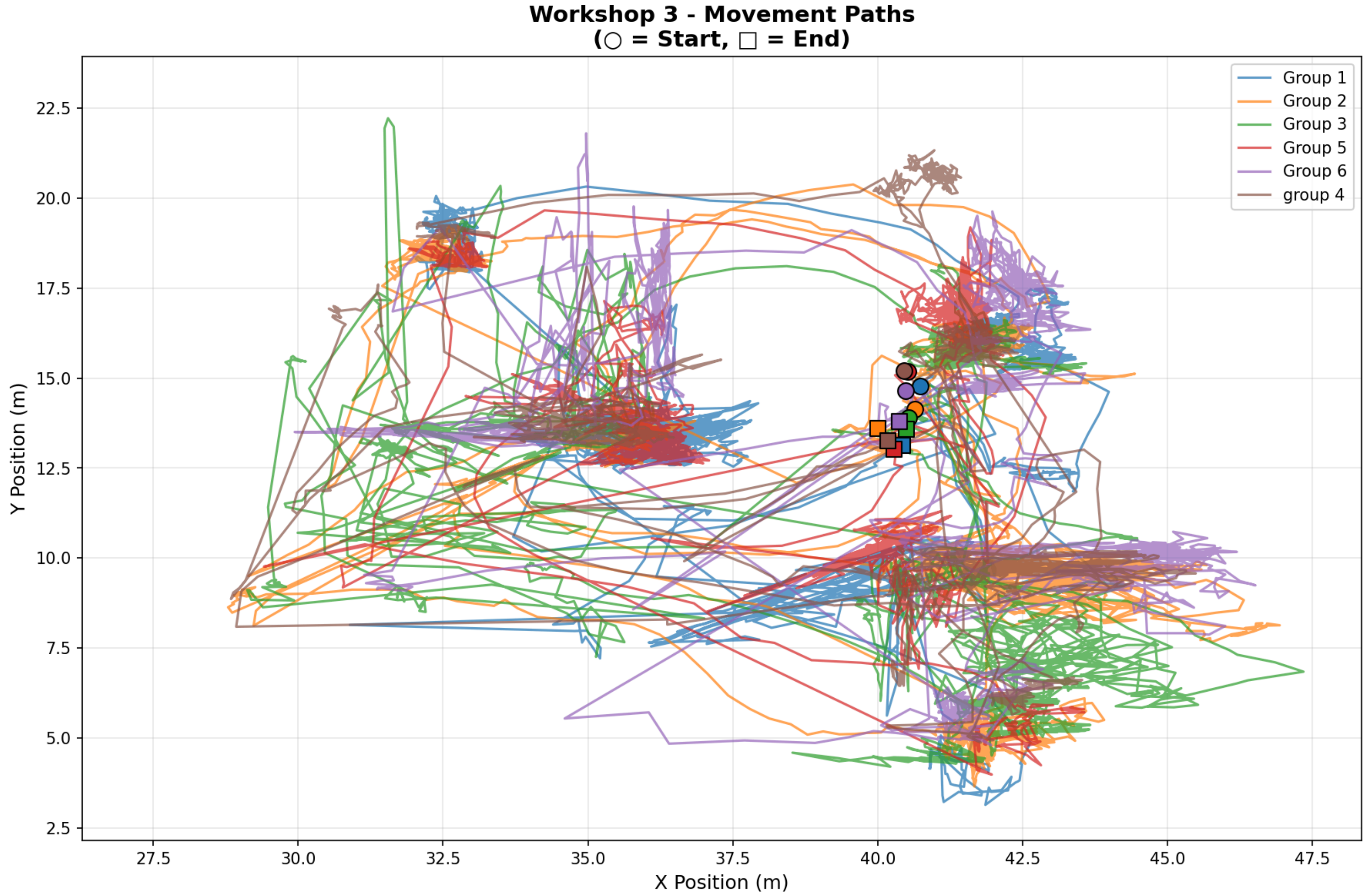


Figure 1.3: Workshop 3 Movement Patterns



Analysis: The spaghetti charts reveal distinct movement characteristics across workshops:

- **Workshop 1**: Shows relatively compact movement patterns with clear clustering around five distinct locations
- **Workshop 2**: Demonstrates tighter spatial distribution with less dispersed movement
- **Workshop 3**: Exhibits more dispersed movement patterns with greater spatial coverage

These patterns suggest variability in sensor data reliability.

4.2 Station Boundary Detection

Figure 2.1: Workshop 1 Station Boundaries (Combined View)

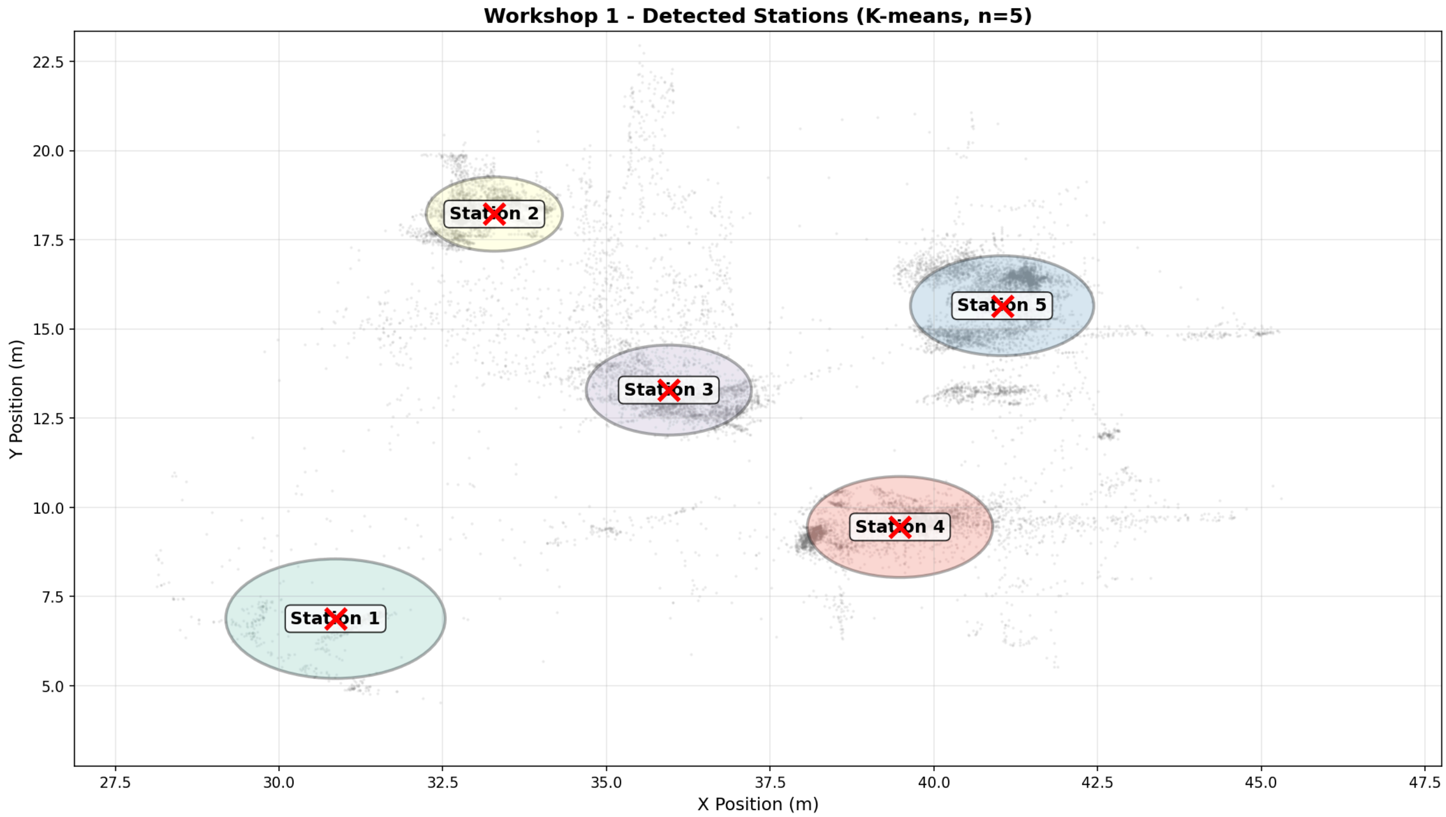


Figure 2.2: Workshop 2 Station Boundaries (Combined View)

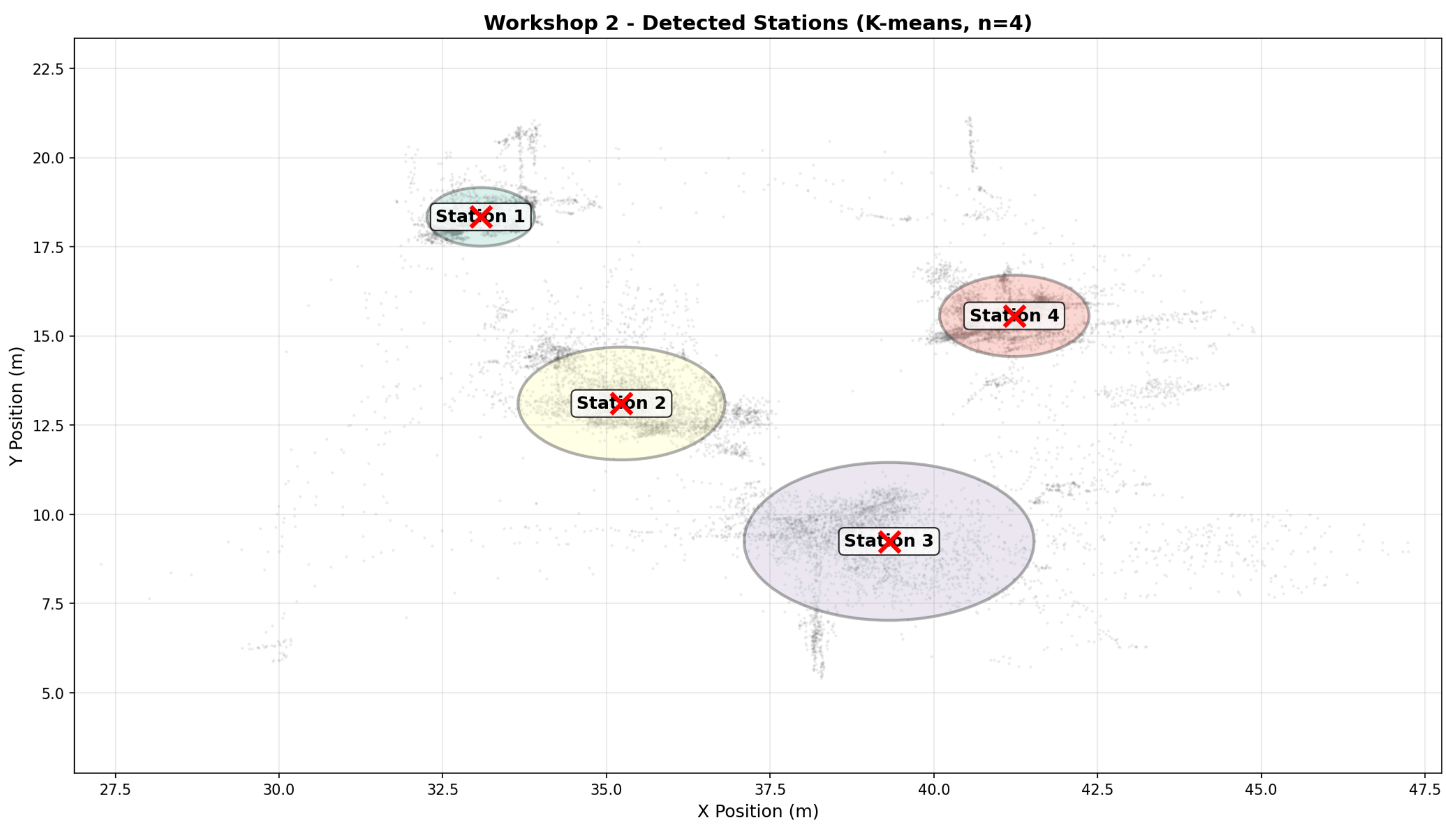
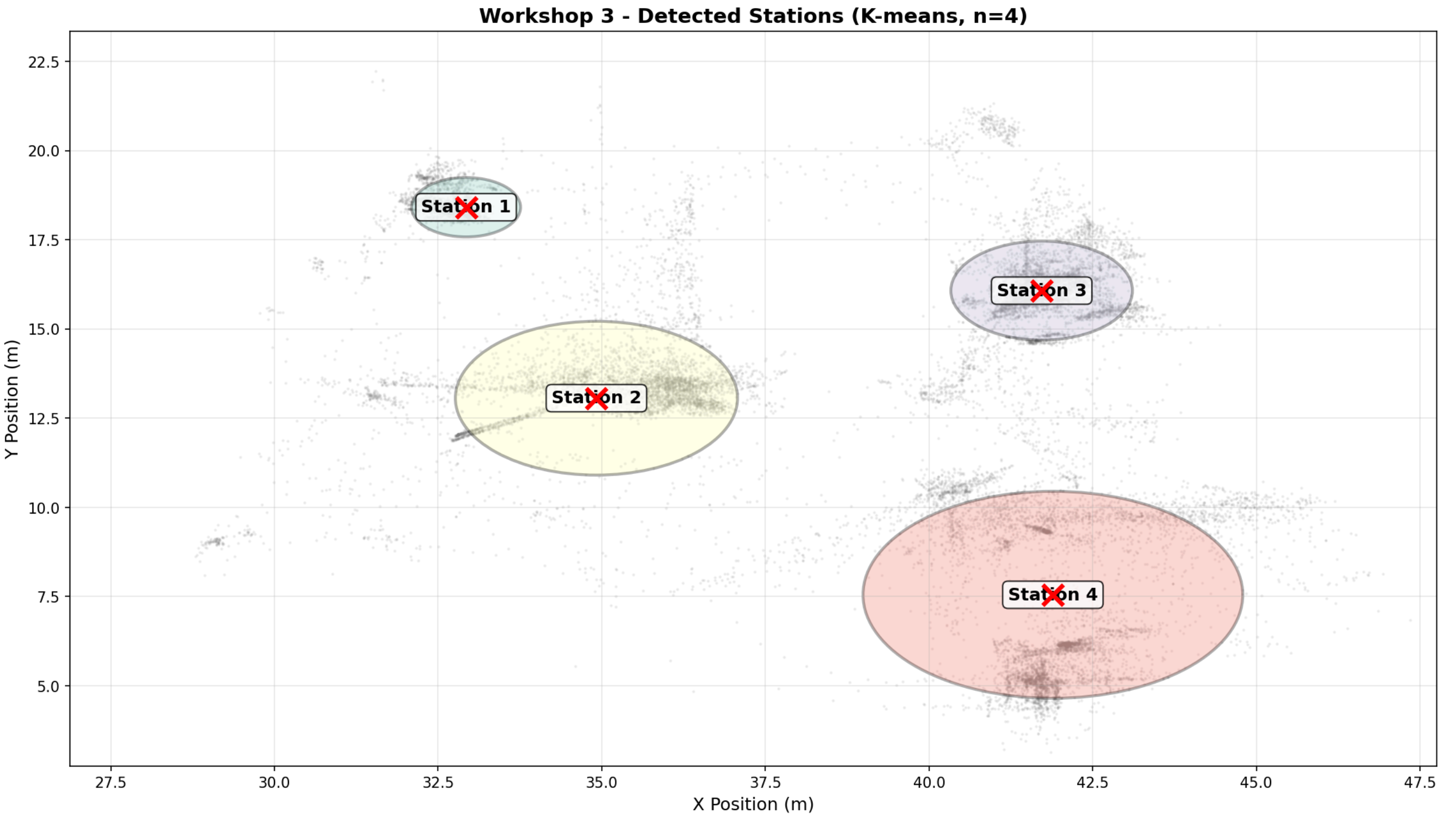


Figure 2.3: Workshop 3 Station Boundaries (Combined View)



Analysis: The K-means clustering successfully identified 5, 4, and 4 stations for workshops 1, 2, and 3 respectively. Station boundaries are visualised as circles with 75th percentile radii, overlaid on actual position data (gray points). Red 'X' markers indicate cluster centers.

Station Spatial Distribution:
This data is at odds of what happened on the days but as shown in the next section this improves for individual group analysis, but these visualisations are helpful to get an idea of how scattered the data is and to provide an explanation for the odd dwell-time and transition time information shown later.

4.3 Individual Group Station Detection Examples

Figure 3.1: Workshop 2, Group 1 Station Boundaries

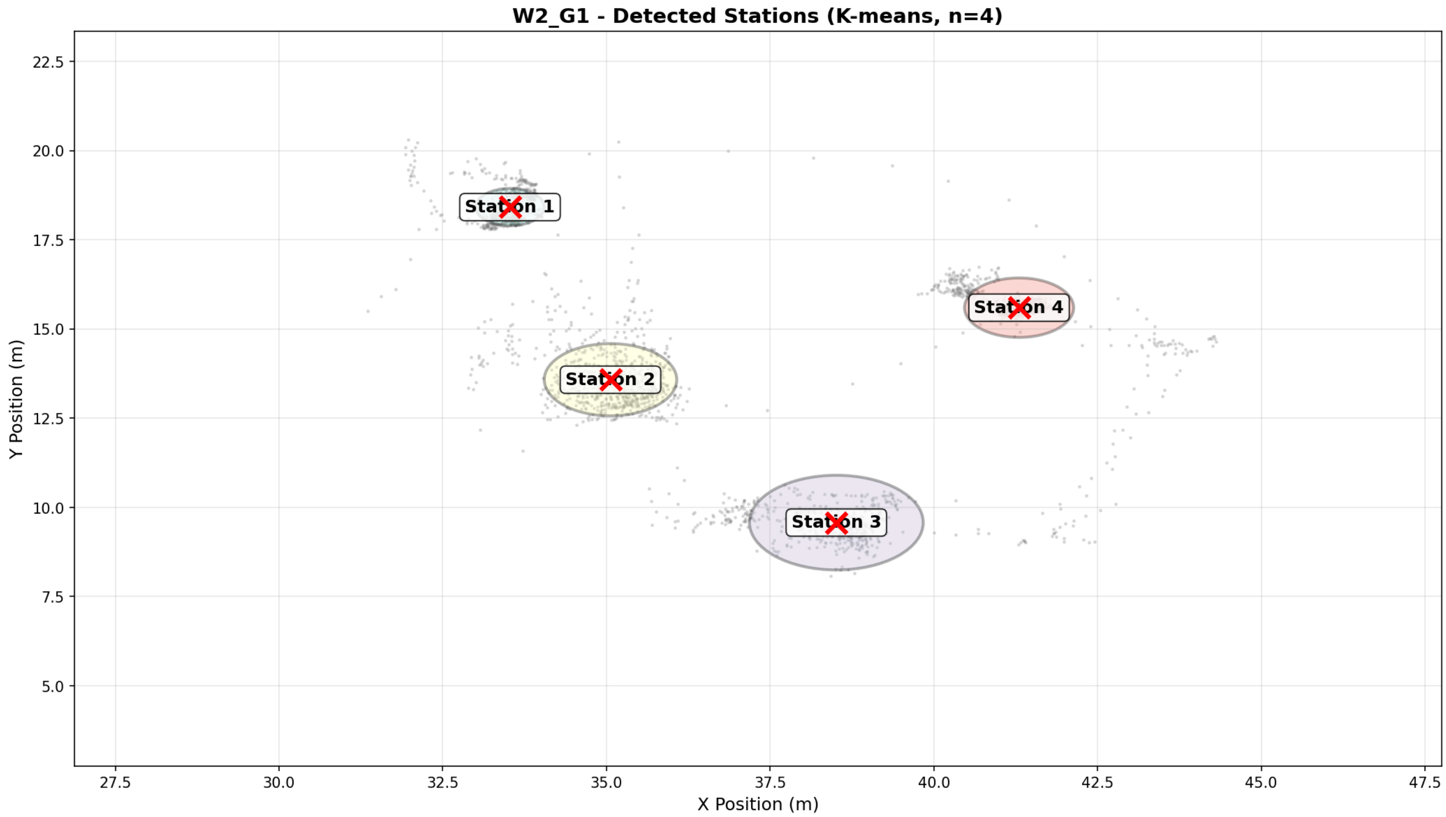


Figure 3.2: Workshop 2, Group 3 Station Boundaries

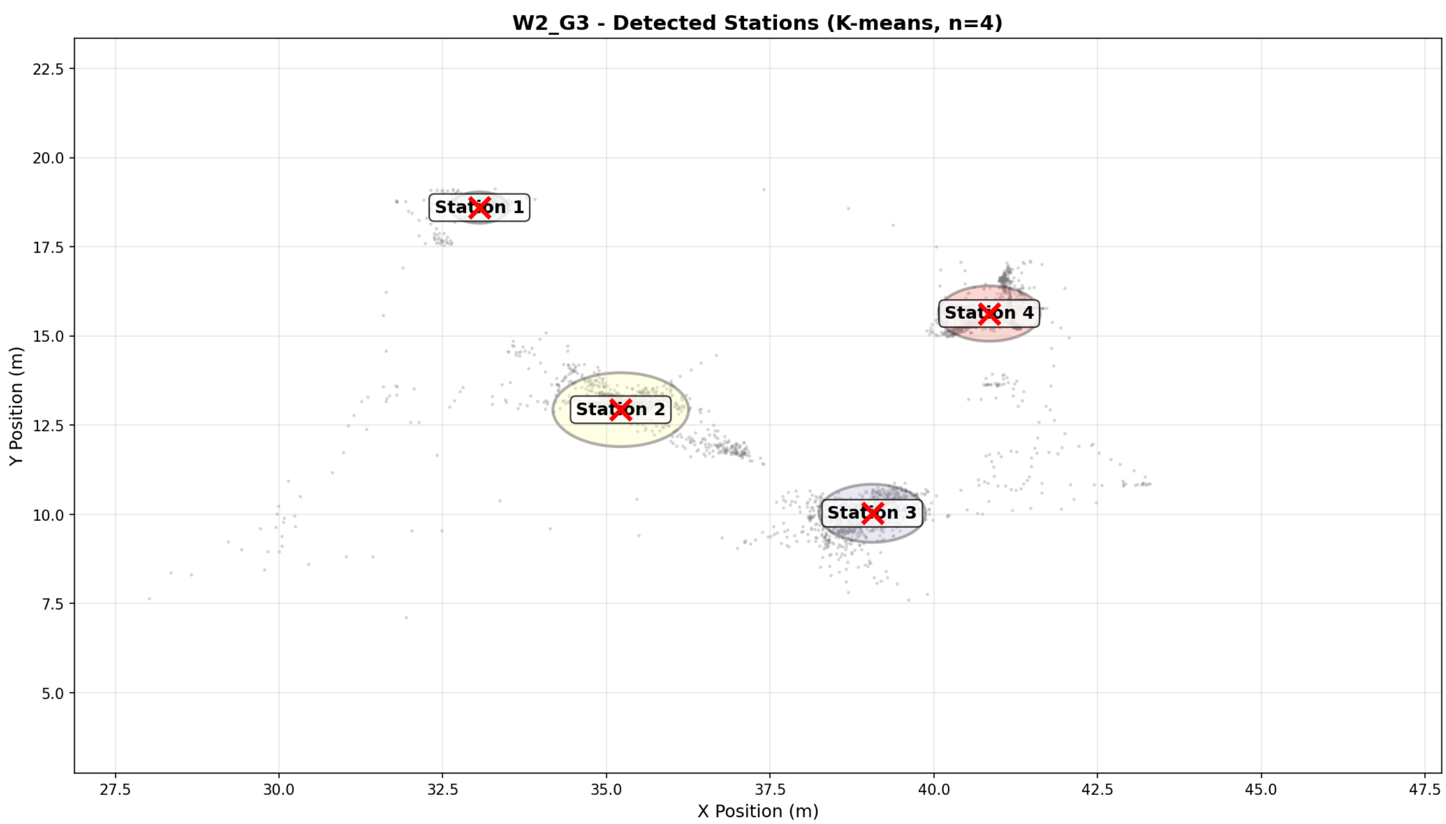
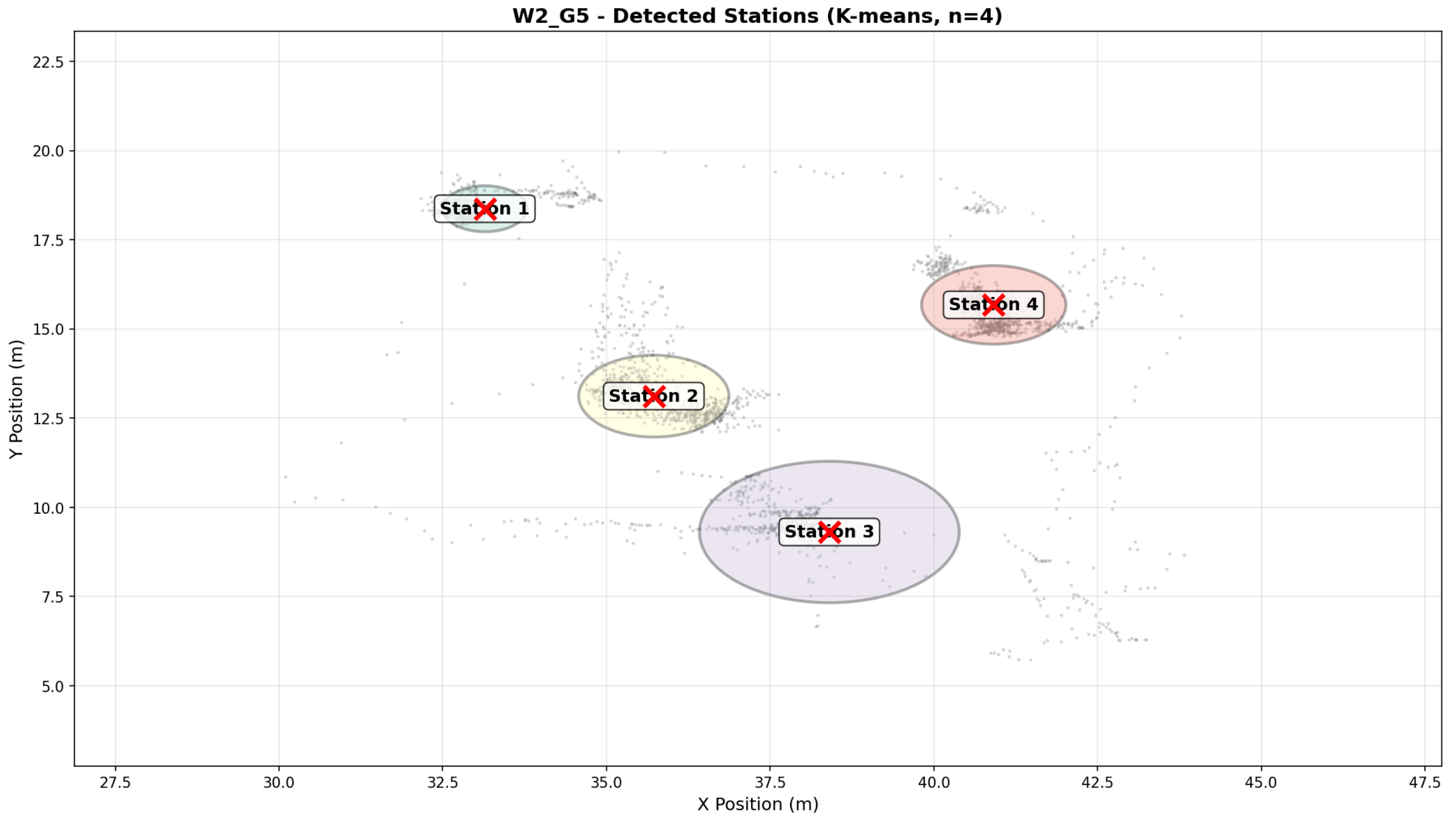


Figure 3.3: Workshop 2, Group 5 Station Boundaries



Analysis: Individual group visualisations demonstrate how some of the sensors had similar data displaying closer to ideal circumstances. Figures 3.1, 3.2, 3.3 all show station's that are pretty close to each other and therefore are some of the best to use for comparison to each other, but even so many data points can be seen outside a station which visually seem to be related to sensor drift or participants wandering as it's not a factory setting but rather a demonstration.

4.4 Dwell Time Comparisons

Figure 4.1: Workshop 1 Dwell Time Comparison (All Groups)

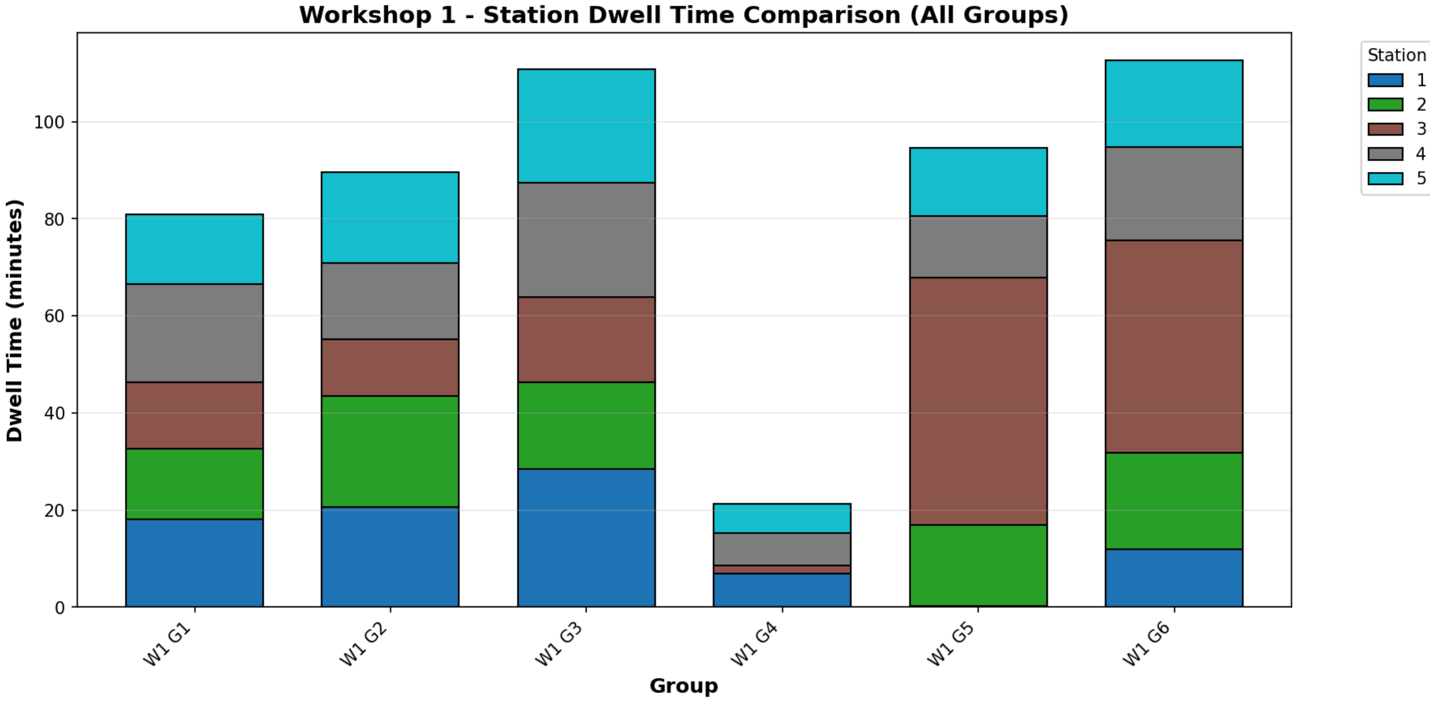


Figure 4.2: Workshop 2 Dwell Time Comparison (All Groups)

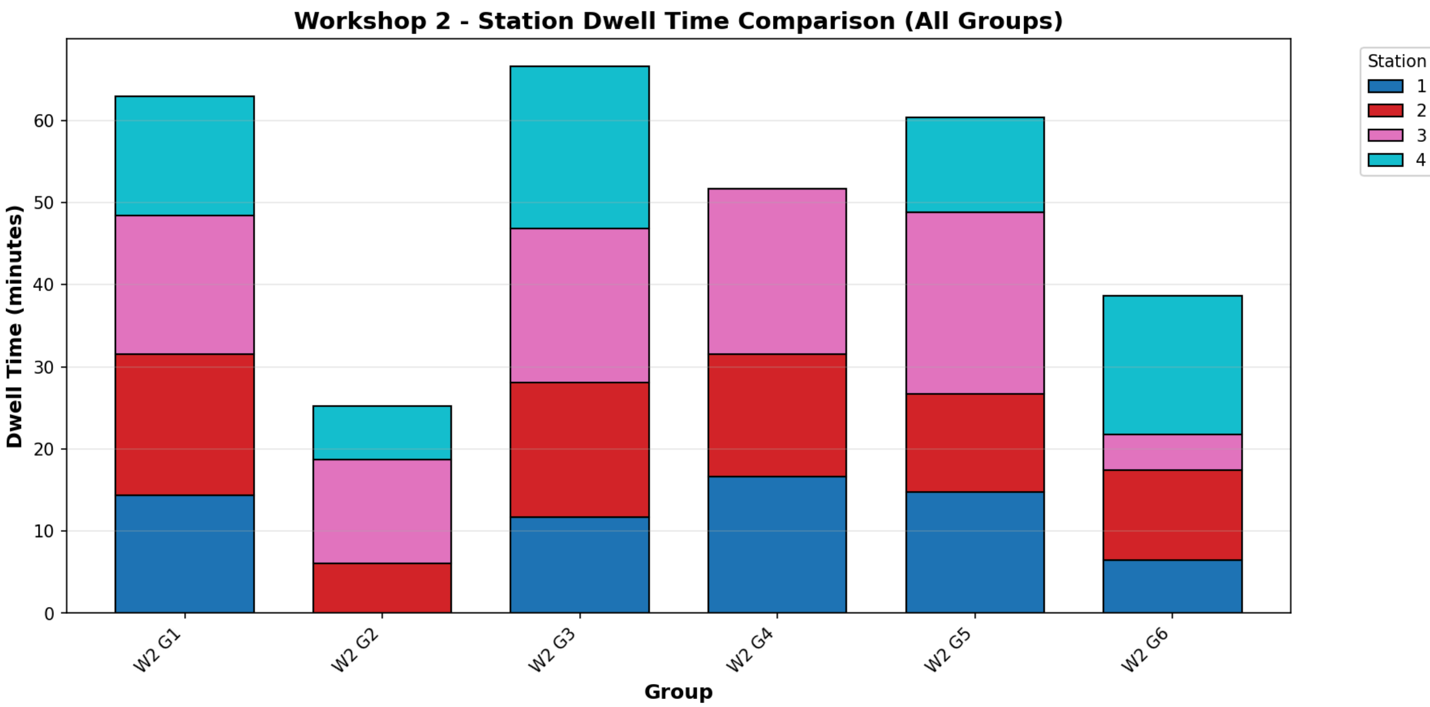
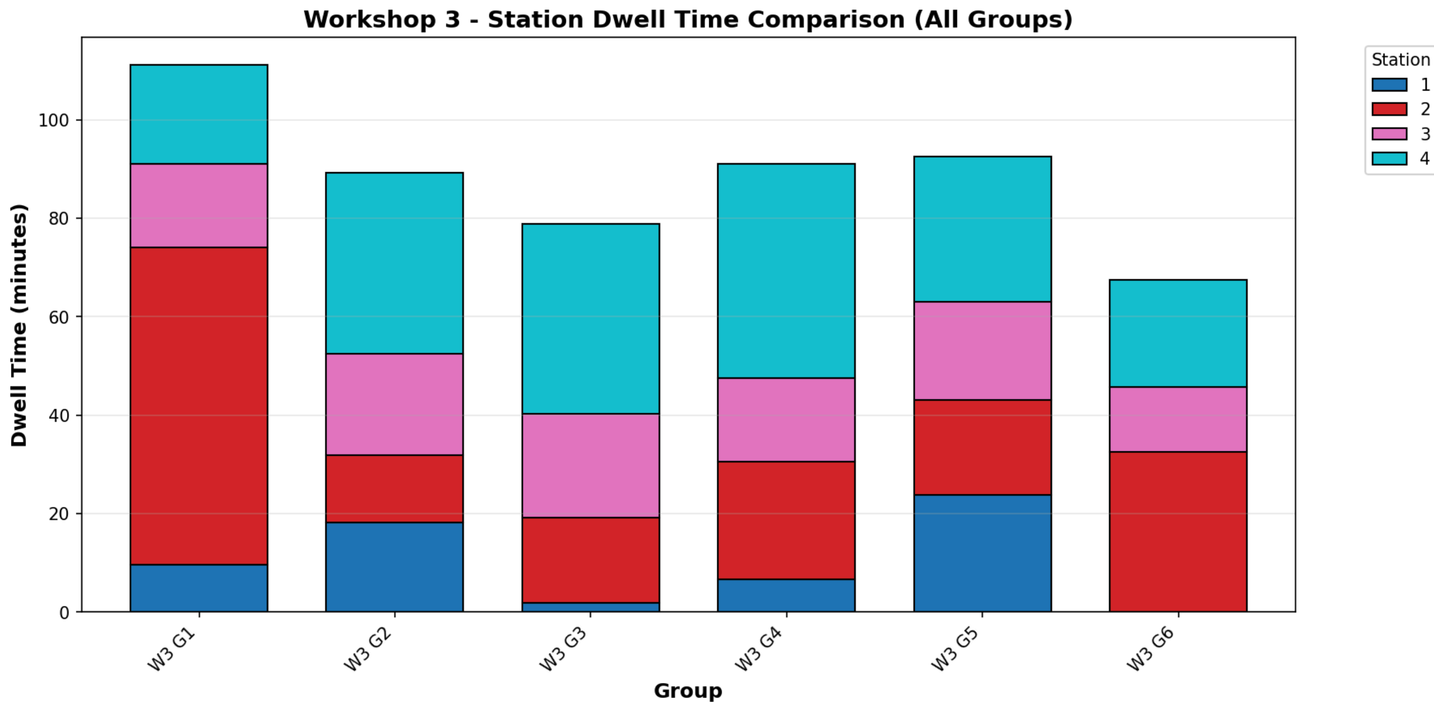


Figure 4.3: Workshop 3 Dwell Time Comparison (All Groups)



Analysis: Stacked bar charts reveal significant variability in dwell times across groups:

Workshop 1 Observations:

- Groups show proportionaly similar times at stations but total times vary quite a bit between workshops.
- Group 4 stands out as particularly unreliable and viewing the boundary map located at `output/boundaries/w1_g4_stations.png` we can see a lot of the sensor data is not captured by the stations.
- Station 1 is missing from Group 5 the reason for this can be seen at `output/boundaries/w1_g5_stations.png` as data wasn't recorded in that corner for this group, most likely due to the group being out of anchor range.

Workshop 2 Observations:

- More uniform dwell time distribution on Groups 1, 3, and 5.
- Wildly inconsistent for Groups 2, 4, and 6. Station 1 is missing from Group 2, and Station 4 is missing from Group 4. This is reflected in their relative boundary images.
- Shorter overall dwell times compared to Workshop 1.

Workshop 3 Observations:

- Quite a bit of variability of dwell-time across the groups.
- No dwell-time data for Group 6 at Station 1.
- Dwell-time at station 2 is very high for Group 1.

Conclusion:

Workshop 2 Groups 1, 3, and 5 have the most reliable data for comparison of different groups at the same session. The rest of workshop 2 is quite inconsistent.

This alludes to inconsistencies in testing.

4.5 Transition Time Analysis

Figure 5.1: Workshop 1 Transition Time Comparison

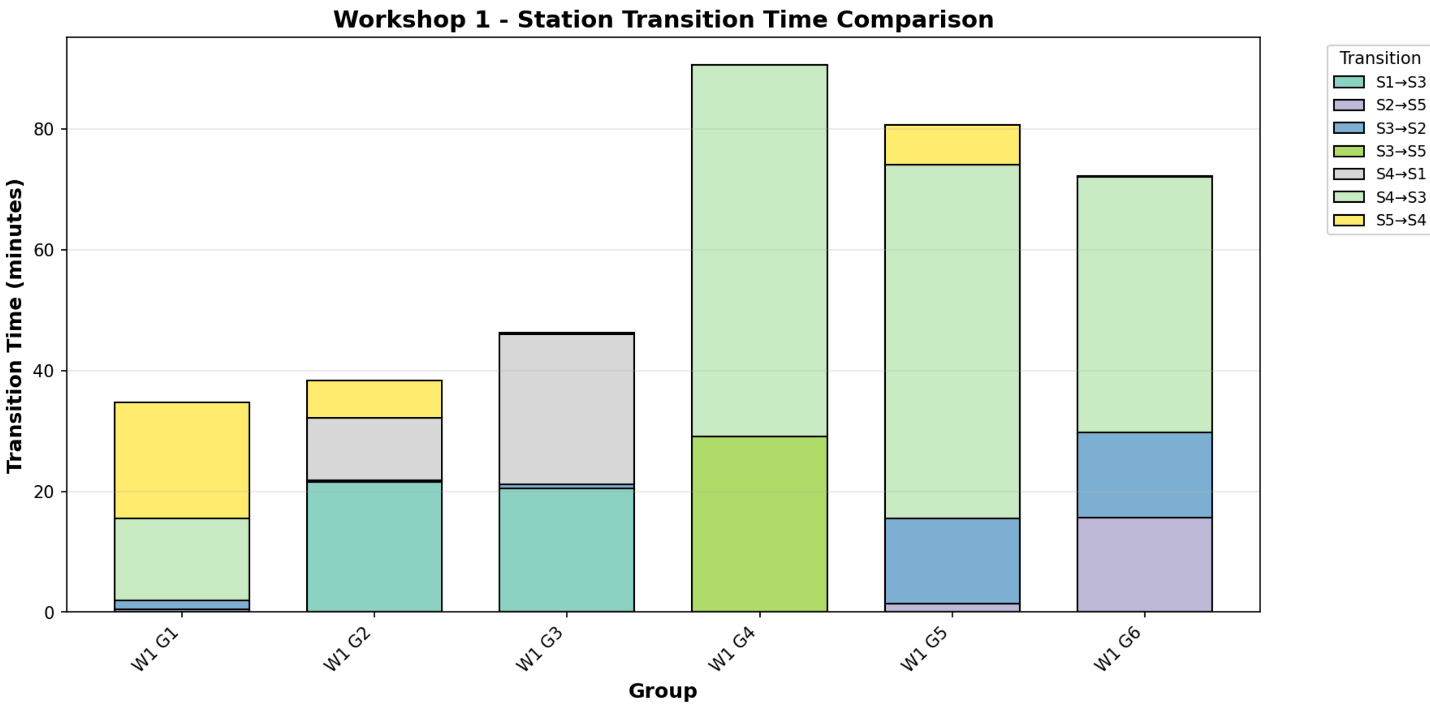


Figure 5.2: Workshop 2 Transition Time Comparison

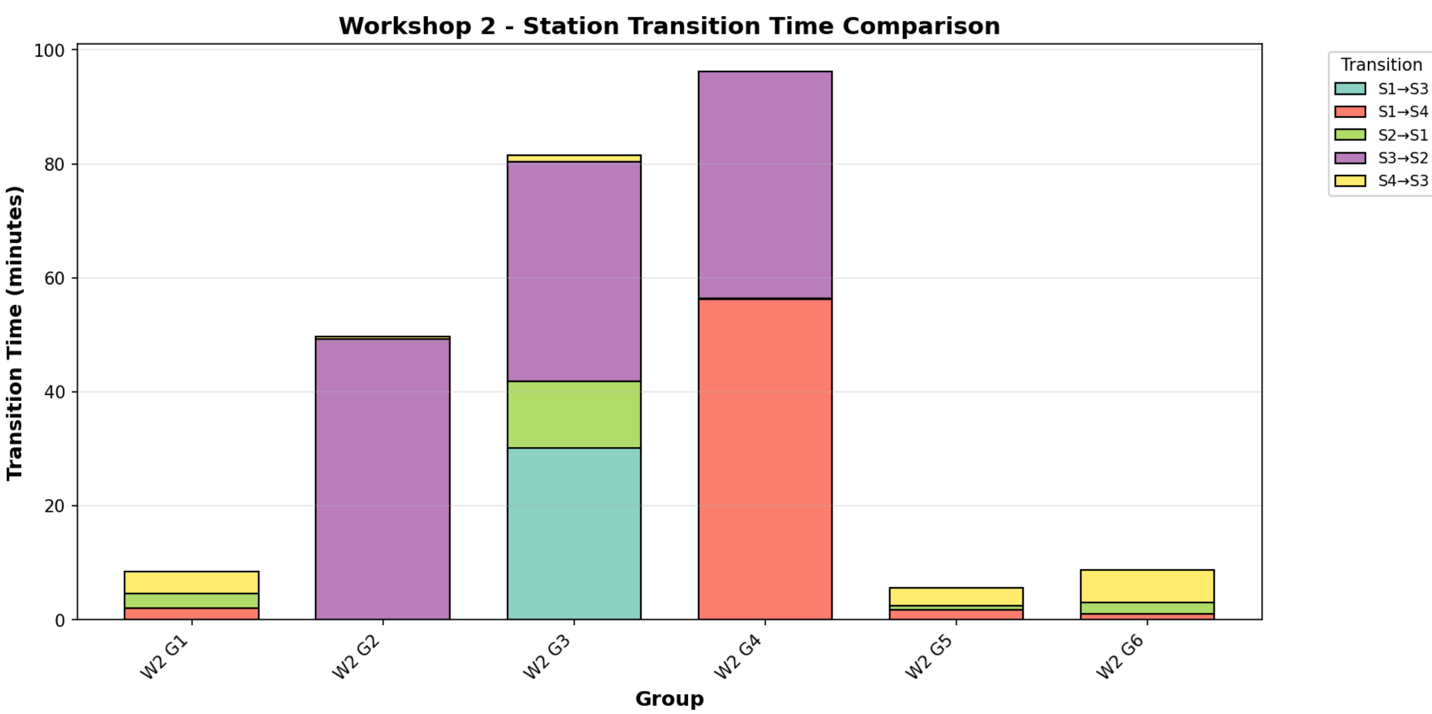
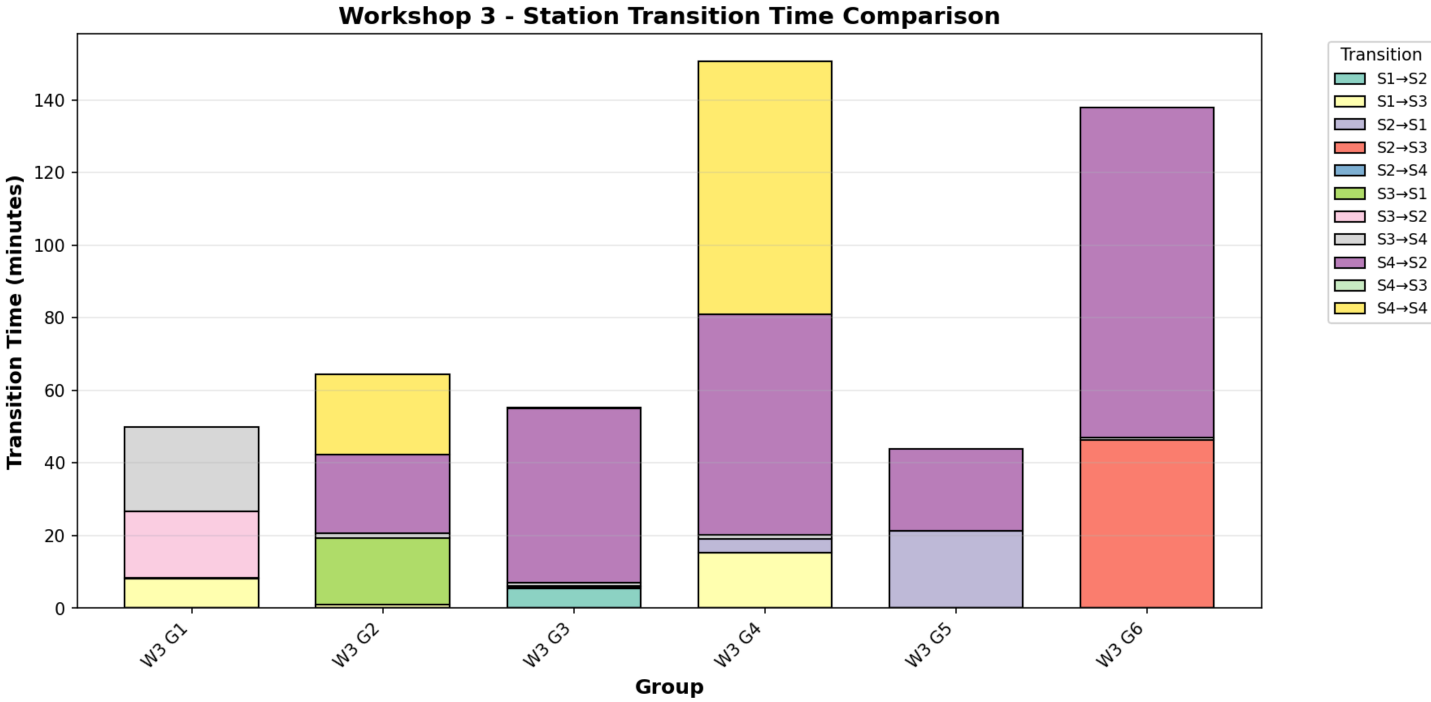


Figure 5.3: Workshop 3 Transition Time Comparison



Analysis:

Transition times are wildly inaccurate, or at-least don't make sense on surface value. More study with cleaner data would need to be done to understand if my transition timing code is working correctly

4.6 Production Time Analysis

Figure 6.1: Workshop 1 Total Production Time

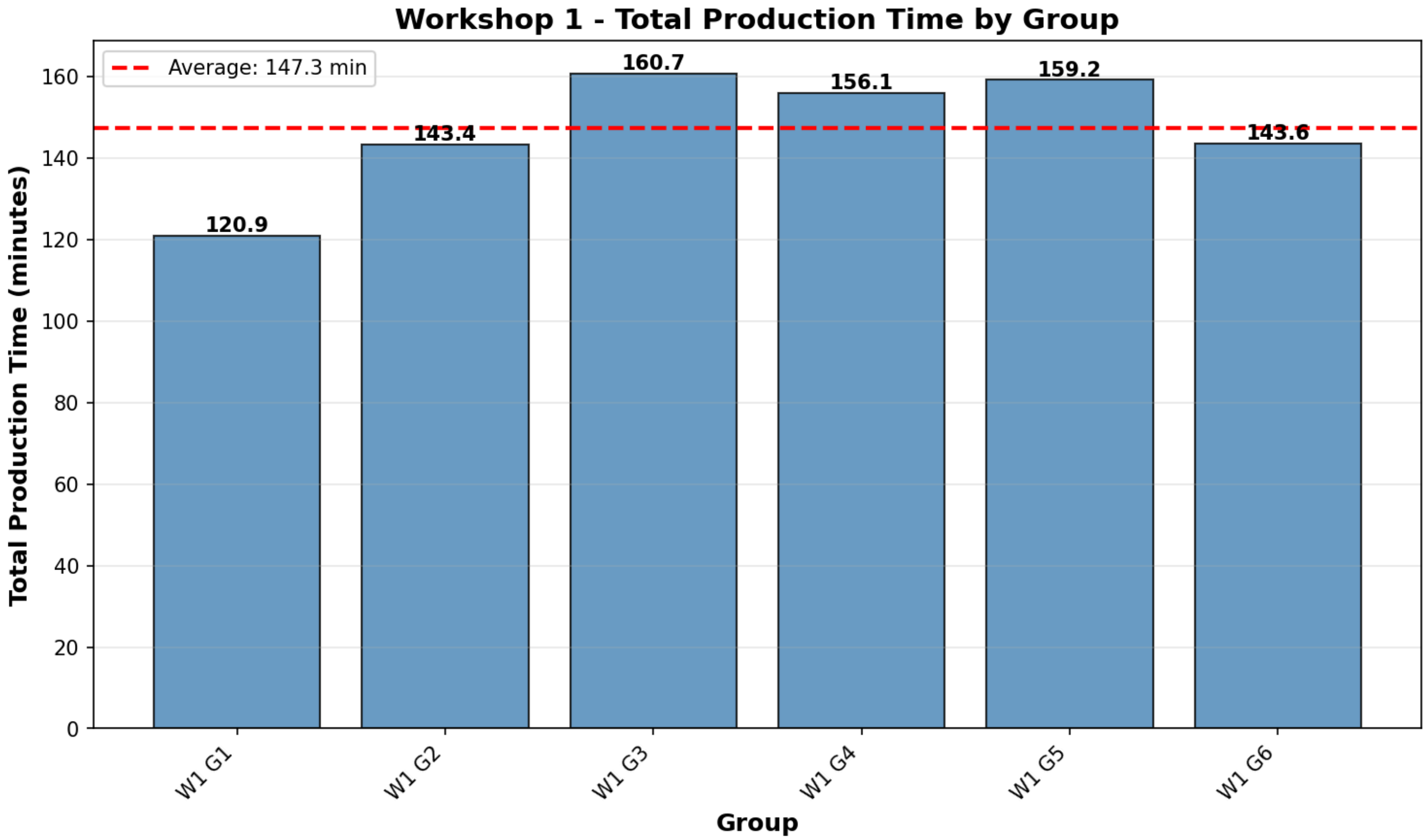


Figure 6.2: Workshop 2 Total Production Time

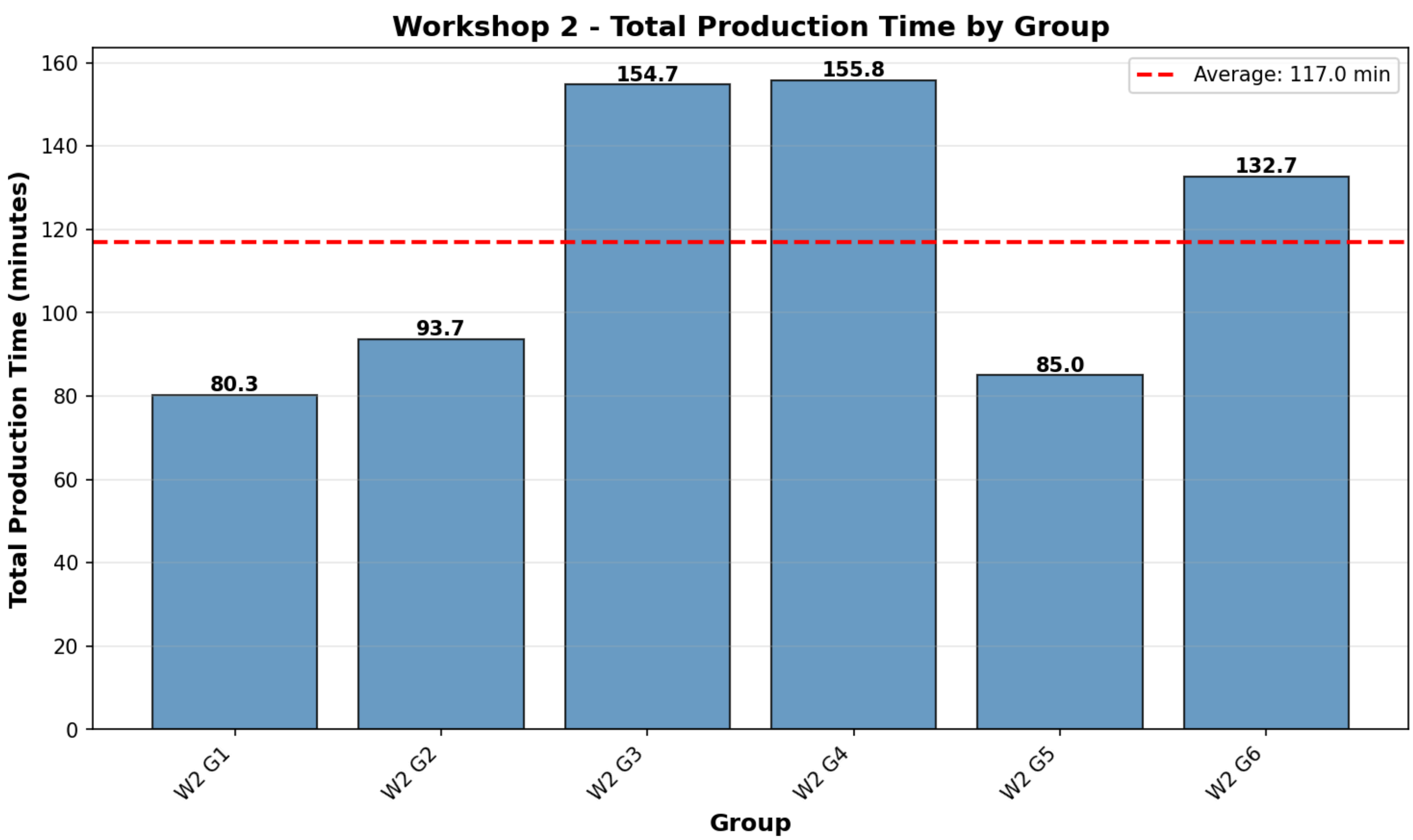
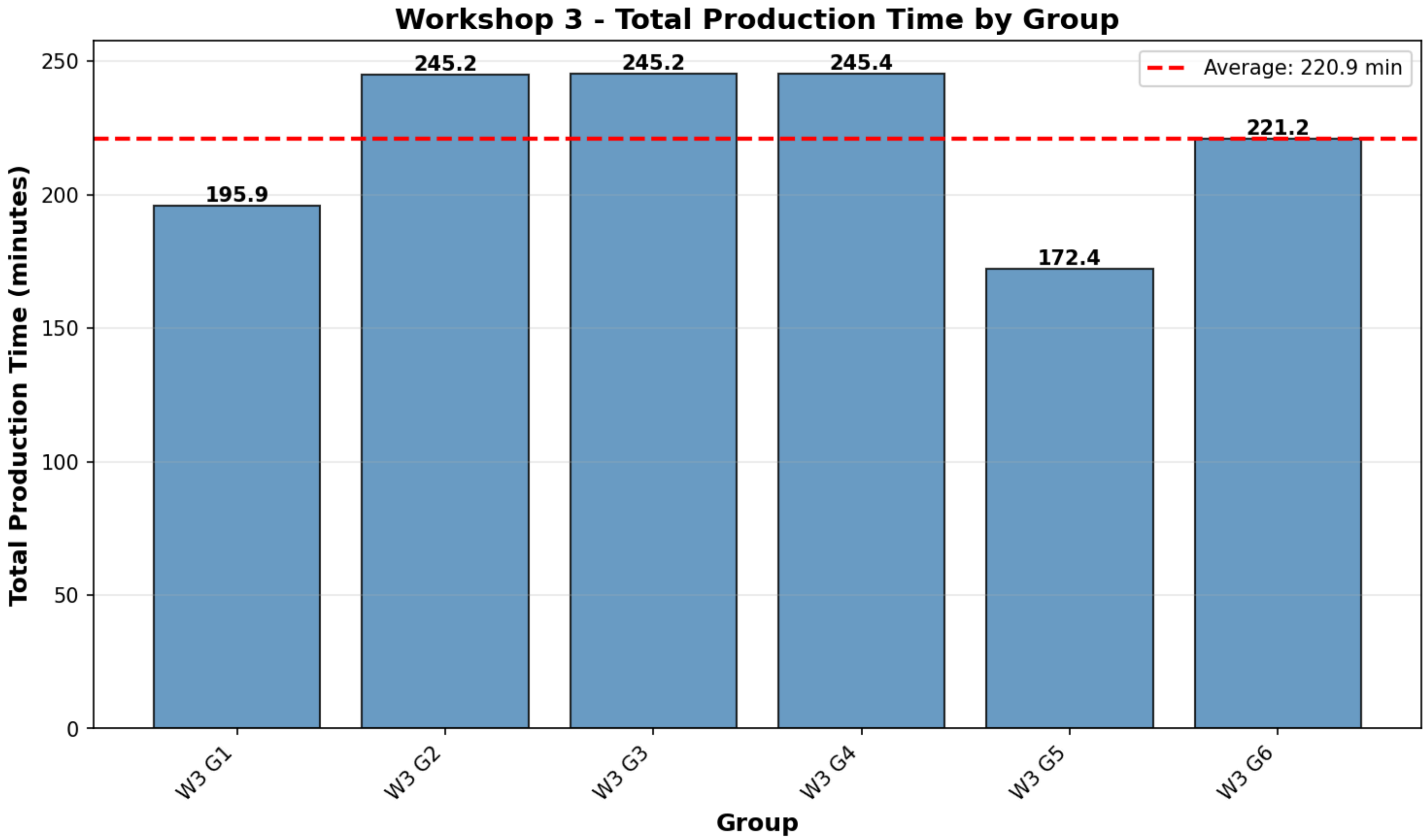


Figure 6.3: Workshop 3 Total Production Time



Analysis:
Total run-times are somewhat consistent for the same workshop which is expected, but the amount of variance is somewhat unexpected because it's simply taking the first sensor reading and last sensor reading (within a station) for total time calculation. This means the sensors maybe weren't all turned on at the same time or some were turned off earlier than others.

5. Comparative Analysis: Dual Approach Evaluation

5.1 Group Comparison vs. Individual Analysis

The project's dual-approach architecture serves distinct analytical purposes:

Table 1: Approach Comparison Matrix

Aspect	Group Comparison (projects/)	Individual Analysis (project-individual/)
Boundaries	Shared per workshop	Individual per group
Visualisations	Stacked comparisons	Individual charts

Aspect	Group Comparison (projects/)	Individual Analysis (project-individual/)
Use Case	Workshop-level comparison	Deep group investigation

5.2 When to Use Each Approach

Use Group Comparison When:

- Comparing performance across groups
- Workshop layout is known and fixed
- Identifying relative performance differences

Use Individual Analysis When:

- Investigating specific group behavior
- Station configuration is uncertain
- Discovering optimal station numbers
- Groups may have followed different workflows

5.3 Trade-offs and Considerations

Group Comparison Advantages:

- Consistent metrics enable direct comparison
- Simpler interpretation for stakeholders
- Faster computation (single clustering per workshop)
- Standardized visualisations

Group Comparison Limitations:

- May miss group-specific patterns
- Assumes all groups used the same stations
- Less accurate for groups with atypical movements

Individual Analysis Advantages:

- Discovers actual group-specific patterns
- More accurate station boundaries per group
- Reveals optimal station configurations

Individual Analysis Limitations:

- Harder to compare across groups directly
- More computational overhead (clustering per group)
- May detect spurious clusters from noise

6. Limitations and Future Work

6.1 Current Limitations

Data Quality Constraints:

- Sensor drift often causes position errors
- Timestamp resolution may miss rapid movements (Only important for rapidly moving assets)
- No validation against ground truth station locations (Could be majorly improved if station locations and boundaries are known)

Algorithmic Limitations:

- K-means assumes spherical clusters (circular station boundaries)
- Anti-backtracking logic may miss legitimate backward movements
- 30-second threshold may filter brief but legitimate station visits

Analytical Constraints:

- No statistical significance testing between workshops
- Limited sample size (6 groups per workshop)
- No temporal pattern analysis (time-of-day effects)

6.2 Potential Improvements

Data Collection Enhancements:

1. Higher-frequency sampling to capture finer movement details
2. Better anchor placement for improved accuracy
3. Integration with task completion timestamps

Algorithm Enhancements:

1. **DBSCAN Clustering**: Handle non-spherical station shapes

Analysis Enhancements:

1. **Network Analysis**: Model workflow as a directed graph

Visualisation Improvements:

1. Interactive dashboards using Plotly or Dash
2. Animation of movement patterns over time
3. Heatmaps of station utilisation density
4. 3D visualisations including time dimension

6.3 Scalability Considerations

Current System Capacity:

- Handles 18 groups (3 workshops × 6 groups) efficiently
- Processing time: ~2-5 minutes for complete analysis
- Memory footprint: Minimal (<500MB for all data)

Scaling to Larger Datasets:

- **100+ groups**: Would require parallel processing
- **Real-time analysis**: Needs incremental clustering algorithms
- **Long-term tracking**: Requires database backend (PostgreSQL, InfluxDB)

6.4 Integration Opportunities

Manufacturing Execution Systems (MES):

- Real-time bottleneck alerts
- Predictive production time estimates
- Resource allocation optimization

Quality Management Systems (QMS):

- Correlate movement patterns with defect rates
- Identify process deviations
- Track operator training effectiveness

Business Intelligence (BI) Platforms:

- Embed visualisations in executive dashboards

- Generate automated reports

7. Conclusions

7.1 Key Achievements

This project successfully demonstrates a comprehensive framework for analysing RTLS data in manufacturing workshop environments. Key achievements include:

1. **Robust Station Detection**: K-means clustering with silhouette analysis effectively identified station boundaries across diverse workshop configurations
2. **Dual-Approach Flexibility**: The parallel implementation of group comparison and individual analysis methods provides analytical versatility for different use cases
3. **Comprehensive Metrics**: The system calculates movement patterns, dwell times, transition times, and production times, providing a holistic view of workshop performance
4. **Scalable Architecture**: Modular design and clear separation of concerns enable future enhancements and scaling

7.2 Technical Contributions

Software Engineering Best Practices:

- Modular code organization with clear function separation
- Comprehensive documentation and inline comments
- Consistent file naming conventions
- Progress reporting for user transparency

7.3 Future Research Directions

Enhancements:

1. Develop predictive models for production time
2. Incorporate machine learning for process optimization
3. Get cleaner data to have clearer results

7.4 Final Remarks

This project demonstrates the value of data-driven analysis in modern manufacturing environments. By transforming raw positional tracking data into analytical insights, the system will with some adaptation enable evidence-based decision-making for workshop optimization, training improvement, and capacity planning. The dual-approach methodology provides analytical flexibility.

The modular architecture and clear documentation facilitate future enhancements and integration with broader manufacturing systems. As RTLS technology becomes increasingly prevalent in industrial settings, frameworks like this project can play a crucial role in extracting maximum value from this rich data source.

Appendices

Appendix A: File Structure Reference

Output File Organization:

```
output/
├── spaghetti/
│   ├── workshop1_spaghetti.png
│   ├── workshop2_spaghetti.png
│   └── workshop3_spaghetti.png
│
├── boundaries/
│   ├── workshop{1-3}_stations.png (3 files)
│   ├── w{1-3}_g{1-6}_stations.png (18 files)
│   └── station_boundaries.json
│
├── dwell_time/
│   ├── workshop{1-3}_dwell_comparison.png (3 files)
│   ├── workshop{1-3}_dwell_times.csv (3 files)
│   ├── w{1-3}_g{1-6}_dwell_times.csv (18 files)
│   └── w{1-3}_g{1-6}_dwell_chart.png (18 files)
│
└── transition_production_time/
    ├── workshop{1-3}_transition_comparison.png (3 files)
    ├── workshop{1-3}_transitions.csv (3 files)
    ├── workshop{1-3}_production_time.png (3 files)
    └── workshop{1-3}_production.csv (3 files)
```

Appendix B: Running the Analysis

Prerequisites:

```
# Install Python 3.11.x
# Install dependencies
pip install pandas numpy matplotlib scikit-learn
```

Data Preparation:

```
cd src
python split_data.py
```

Running Group Comparison Analysis:

```
cd projects
uv run run_all.py
# Or run individual scripts:
# uv run 1_spaghetti_chart.py
# uv run 2_station_boundaries.py
# uv run 3_dwell_time.py
# uv run 4_5_transition_production_time.py
```

Running Individual Analysis:

```
cd project-individual
uv run run_all.py
```

Appendix C: Configuration Parameters

Key Constants in Scripts:

```
# Station Detection
K_RANGE = range(3, 10)  # Individual analysis (auto-detect)
SILHOUETTE_THRESHOLD = 0.5  # Minimum acceptable clustering quality

# Dwell Time Analysis
MIN_DWELL_SECONDS = 30  # Sensor drift filter
RADIUS_PERCENTILE = 75  # Station boundary calculation

# Visualisation
DPI = 150  # Output image resolution
FIGURE_SIZE = (14, 8)  # Default figure dimensions (inches)
ALPHA = 0.3  # Station boundary transparency
```

Appendix D: Data Format Specification

Input CSV Format (data/split/w{X}_g{Y}.csv):

```
name,x,y,time
Group 1,10.5,20.3,2024-01-01 10:00:00
Group 1,10.6,20.4,2024-01-01 10:00:01
...
```

Station Boundaries JSON Format (output/boundaries/station_boundaries.json):

```json
{
  "workshop1": [
    {
      "station_id": 1,
      "center_x": 12.5,
      "center_y": 8.3,
      "radius": 2.4,
      "num_points": 1250
    },
    ...
  ],
  ...
}
```

Dwell Time CSV Format (output/dwell_time/workshop1_dwell_times.csv):

```
group,station,dwell_time_seconds,dwell_time_minutes
W1 G1,1,450.5,7.51
W1 G1,2,380.2,6.34
...
```

Appendix E: Troubleshooting Guide

Common Issues and Solutions:

Issue: "File not found" errors

Solution: Ensure you're running scripts from correct directory (projects/ or project-individual/)

Issue: No output generated

Solution: Check that data files exist in data/split/; run src/split_data.py if needed

Issue: Different number of stations detected

Solution: This is expected in individual analysis; silhouette analysis finds optimal k per workshop

Issue: Memory errors with large datasets

Solution: Process workshops individually; increase system memory; use data subsampling

Issue: Slow processing

Solution: Reduce k_range in silhouette analysis; use fewer iterations in K-means (n_init=10)

References

Technical Documentation

1. README.md - Main project documentation
2. projects/README.md - Group comparison approach documentation
3. project-individual/README.md - Individual analysis approach documentation

Software Libraries

1. pandas: Data manipulation and analysis library
2. numpy: Fundamental package for scientific computing
3. matplotlib: Comprehensive visualisation library
4. scikit-learn: Machine learning library (K-means, silhouette score)

Project Repository: https://github.com/NonExstnt/COS4-RTLS