

Enabling Real-Time Robot-Tracker Communication via MQTT on the Factory Cloud for Metrology Automation

Seemal Asif*

Centre for Robotics and Assembly,
Cranfield University, Cranfield
MK43 0AL, UK
0000-0001-7048-0183

Jon Hall

University of Sheffield, AMRC,
Factory 2050, Sheffield
S9 1ZA, UK
0000-0002-8494-8645

Daniela Sawyer

University of Sheffield, AMRC,
Factory 2050, Sheffield
S9 1ZA, UK
0009-0001-0056-2578

Abstract—In large-scale precision manufacturing, achieving real-time high-precision alignment between industrial robots and metrology systems remains challenging. This paper proposes a cloud-integrated communication architecture that enables real-time, bidirectional data exchange between industrial robots and laser trackers. The system leverages the MQTT publish/subscribe protocol within the AMRC Factory+ cloud environment to orchestrate metrology data flows. Robot controllers publish motion commands while laser trackers provide real-time position feedback, forming a closed-loop calibration process. Measurement data is processed through dedicated adapter layers and calibration modules, then disseminated by an MQTT broker to a central control system which computes alignment corrections and updates robot motion. The architecture employs edge computing and cloud services for modularity, scalability, and low latency, and integrates with existing metrology software (e.g., Spatial Analyzer, PolyWorks). These findings underscore the efficacy of cloud-based robot-tracker coordination for dynamic, high-precision calibration and control. The proposed approach provides a robust foundation for future Industrial IoT (IIoT) deployments in metrology-driven manufacturing systems.

Keywords—MQTT communication, cloud-based metrology, industrial IoT, real-time robot calibration, High Precision Manufacturing

I. INTRODUCTION

Manufacturing robots are increasingly connected via cloud and edge computing architectures to leverage greater computational power and data resources. Cloud robotics is the paradigm of off-loading heavy computation or data storage from robots to cloud servers, enabling robots to benefit from virtually unlimited resources on demand[1][2]. By tapping into cloud services, robots become lighter and smarter – able to access global libraries of data (maps, CAD models, etc.) and to learn from other machines’ data via shared cloud knowledge bases [1]. However, a well-known challenge is meeting real-time control requirements when communication involves remote cloud servers[3]. Network latency and uncertainty can be prohibitive for fast control loops. To address this, modern architectures introduce edge or fog computing layers between robots and the cloud [4]. Edge computing resources (industrial PCs, local servers, or 5G edge nodes) are located closer to the factory floor to perform time-critical processing with low latency. In an IIoT-based robotic cell, this could mean using a local edge controller to coordinate multi-robot motion or safety interlocks in real time, while higher-level optimization or data aggregation occurs in the cloud. Research prototypes have demonstrated that

combining 5G wireless networks with edge/cloud coordination can achieve real-time teleoperation and control.

A digital twin is a live virtual model of the robot and process that runs in the cloud or edge; by synchronizing this model with real sensor data in real time, one can predict and control robot behavior more accurately despite network delays[5]. Xu et al. (2021) proposed a Digital Twin-based Industrial Cloud Robotics framework where each physical robot is paired with a cloud-based “digital robot” that mirrors its state, and robot control services operate on these synchronized twins[5]. This architecture aimed to overcome latency and state inconsistency by ensuring the cloud’s model is always updated with live data, enabling fine-grained real-time control from the cloud side. In summary, cloud and edge computing are being actively integrated into robotic manufacturing systems: the cloud provides massive computing power and global data access, while edge computing and high-speed networks (5G, TSN) ensure that low-latency control and coordination are maintained. Together, they form a tiered real-time control architecture – where critical control loops run at the edge (or on the device), and computationally intensive or big-data-driven tasks run in the cloud – achieving both high performance and intelligence in automation systems[1]

II. FACTORY+

Factory+ is an open and modular framework designed to streamline and standardize data management for manufacturing organizations[6]. It champions the concept of connecting devices and applications to infrastructure instead of each other, offering a consistent and normalised method for data extraction, transport, storage, protection, and consumption. Data is collected from machines through edge devices which are located in the control cabinets of those machines and publish data to a central location via the local area network. This creates an IIoT infrastructure that is designed for dynamic environments, where machine configurations and data requirements may change, and the IIoT infrastructure can be quickly adjusted to accommodate that. Factory+ establishes a core set of interconnected components, forming the foundation for an Industry 4.0 architecture that fits well within the contemporary industrial landscape.

Although Factory+ is an abstract framework, the Advanced Manufacturing Research Centre (AMRC) has enhanced its practical application through the development of the AMRC Connectivity Stack (ACS). ACS comprises open-

source components curated by the AMRC to support a comprehensive, end-to-end implementation of the Factory+ framework; now adopted as an operational standard across the AMRC to collect and consume research data. Factory+ is an edge-driven architecture, enabled by a fleet of edge agents that each communicates with one or more devices (e.g. a PLC) to collect data. Each Edge Agent then organises that data into a pre-defined Schema which assigns a label, data type, and unique identifier to each variable in the data payload. The payload is then published to a central MQTT broker that connects to all of the edge agents and Factory+ components. The structured data is stored within the data warehouse that contains all of the historical data from across the Factory+ infrastructure.

Communication through the MQTT broker is standardised according to the Sparkplug specification, which defines a structure of Nodes and Devices within the network and how data packets and device status messages are reported by each client[7]. Each Sparkplug Node can publish and subscribe to messages of 5 types: BIRTH, DEATH, DATA, CMD, and STATUS. CMD messages are used to control Nodes from other components, which can be used to communicate data securely from one node to another. Following the Sparkplug specification, only the “primary application” can send command messages. In Factory+ this is handled by the Commands component, which receives requests from other components via its REST API or MQTT interface, authorises the permissions of the requesting component via the Authorisation component, and sends command messages to the relevant Sparkplug Nodes.

The core premise of Factory+ is the Unified Namespace (UNS), which is a data model that uses centralised data collection, hierarchical structure, and high-context data structures to make data from any source readily available within an IIoT infrastructure. One of the key features of UNS is the “hub and spoke” layout which means that nodes within the IIoT network do not connect directly with each other, but to a central data broker. This means that modifying components of the network or adding new nodes becomes simplified, and the process for connecting a new device is the same as any existing device. This also decouples the data producers from the data consumers, since they are no longer communicating through point-to-point connections. By using a publish/subscribe data distribution model the central broker can determine which data is sent to a particular consumer based on the subscriptions of that consumer. Data producers simply publish all their data to the central broker with no required knowledge of where it is being consumed.

The use of a central data broker and the publishing of contextualised data within the UNS means that some edge processing is required. In an industrial environment

controllers, PLCs and data acquisition systems will communicate using a range of protocols, so the data packets will need to be translated before they can be published to the UNS. Some data sources will also not include the contextual meta-data in their data packets, so this will need to be added at the edge processing stage. This means that all the data published to the UNS will have the same data structure and use the same communication protocol, so data consumers can interpret data from any device connected to the UNS using the same connection to the central broker and interpret that data in the same way. The edge devices mark the outer border of the UNS, and outside of that the UNS places no requirements on the devices producing data other than making that data available in some way. This structure means that data produced by multiple sources can be consumed by any other node on the network with minimal prior knowledge of the data producer. In this project, this means that positional data from two different robots can be collected and analysed using a single connection into the ACS infrastructure. The results of the data analysis can then be published back into the infrastructure and sent to the robot as command messages.

III. AIMS METROLOGY FACTORY+ BRIDGE

A. Setup

For this project, two serial arm robots have been connected to ACS and constantly publish positional data to the central broker. These robots are referred to as the MABI cell Figure 1 and the VIEWS cell Figure 1. The VIEWS cell is configured for drilling the leading edge of aeroplane wings. The robot positions the drilling head above the wing skin and the drill moves on a linear actuator to generate the hole through the skin and into the rib beneath. The positioning and depth of these holes is crucial for the structure and aerodynamics of the wing surface. A serial robot is generally not accurate enough to achieve this, so external metrology systems are required.

The MABI cell is configured for robotic machining. Performing manufacturing operations usually reserved for CNC machines. For a serial robot to achieve a similar level of accuracy as a cnc requires a significant increase in the control of the robot position and movement path. For this reason, the MABI robot uses a combination of external encoders and external metrology systems so improve the path accuracy of the robot.

Both robot cells use Siemens controllers equipped with OPC UA servers. An edge device has been deployed in each cell to collect the positional data from the OPC UA server, format the data into the Factory+ data structure, and publish it to the central broker. Using the OPC UA protocol also allows data values to be written to the robot controller through the same communication channels.

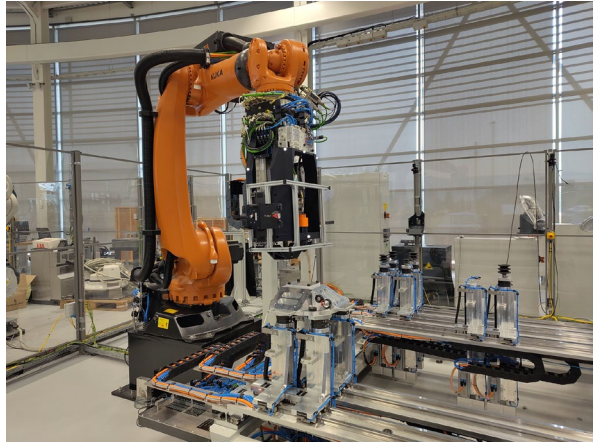


Figure 1: (Left) MABI Cell, (Right) VIEWS Cell

B. System Architecture

AIMS is built on the principles of Service-Oriented Architecture (SOA), wherein each component operates as an independent, interoperable service, it is illustrated in Figure 2. The system comprises several core modules:

- Laser Tracker, responsible for real-time measurement acquisition
- Realtime Adapter, which ensures seamless data transmission
- Central Control System, the processing and decision-making hub

- Robot Controller, which executes motion instructions (the MABI and VIEWS both robot controllers have been developed and utilise)

- Vision System, providing enhanced spatial understanding

These components interact over a network using a combination of MQTT and OPC UA protocols to ensure reliable and secure data flow. The system also integrates an Error Correction and Compensation module that uses historical and real-time data to refine robot movements. A central database stores measurements and corrections, enabling adaptive learning and long-term accuracy.

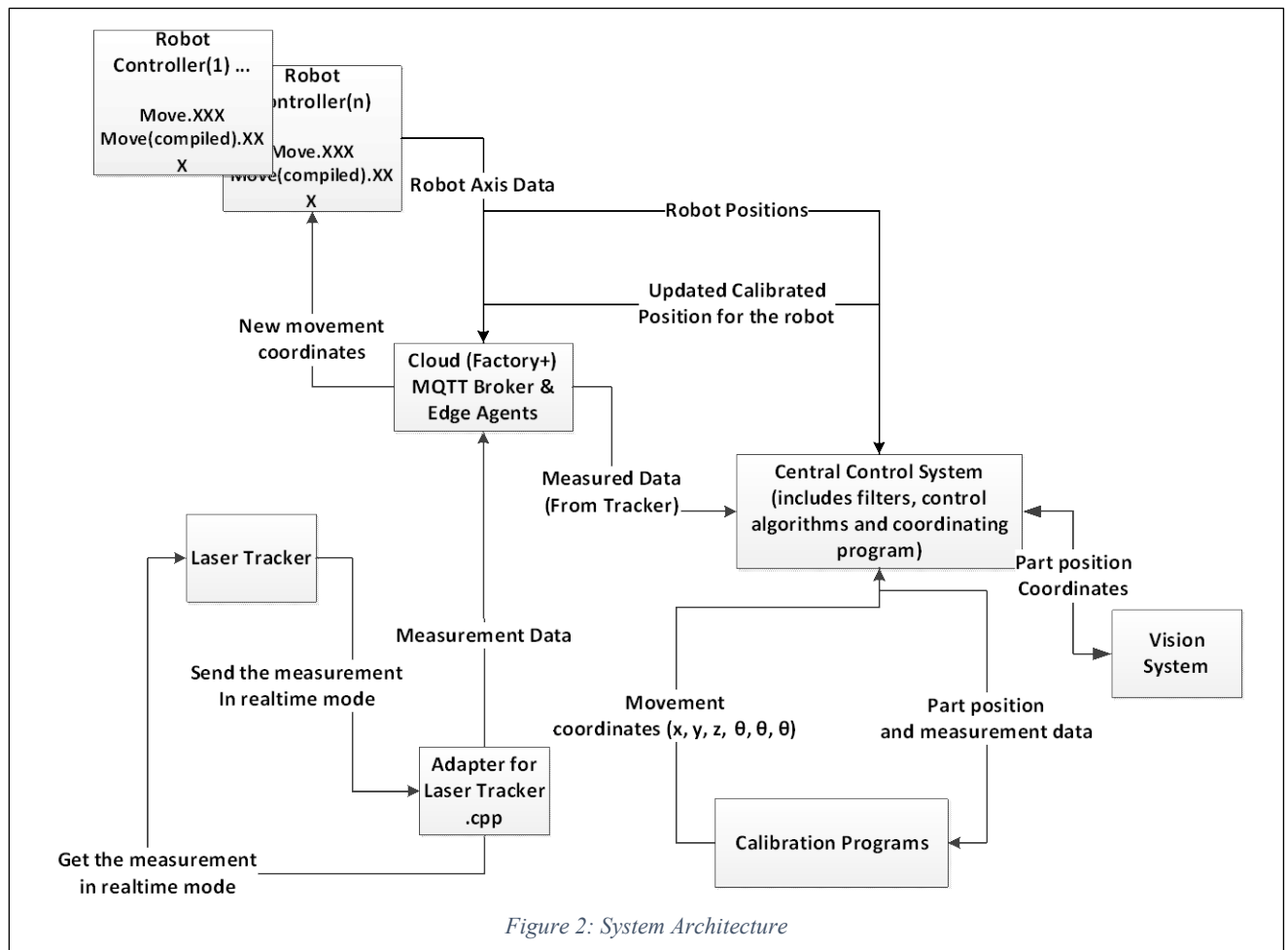


Figure 2: System Architecture

The AIMS MQFactory+ Human-Machine Interface (HMI), as depicted in Figure 3, serves as the central visual and

operational bridge between the metrology systems, the Factory+ infrastructure, and the robotic workcells. This

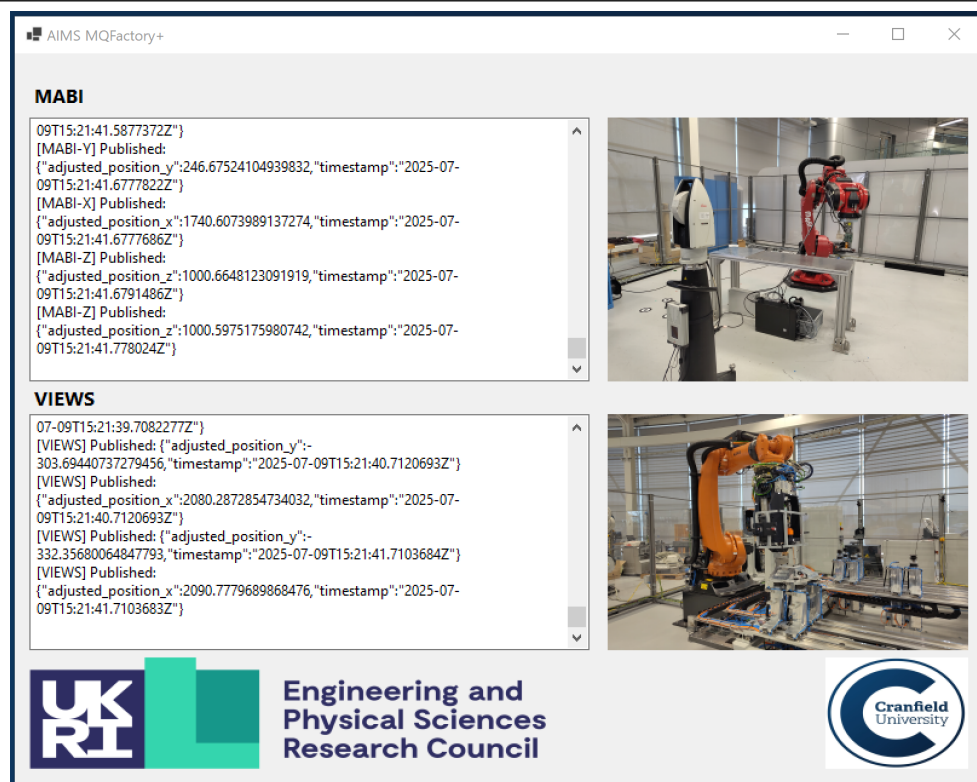


Figure 3: AIMS MQFactory+ HMI

interface is not merely a visualization layer but a functional component within the real-time control loop. It enables operators and system engineers to monitor, log, and verify the bi-directional flow of data between robotic controllers and laser trackers through the MQTT protocol, adhering to the Sparkplug B specification.

C. Results

The effectiveness of this process is illustrated in Figure 4, which shows the real-time VIEWS position data, comparing the raw and corrected positional values across the X, Y, and Z axes for both robots. The close alignment between the raw and corrected values indicates high accuracy in the correction algorithm and minimal latency in the feedback loop. Similarly, the positional data for the MABI robot was collected and processed to produce corrected position outputs. Figure 5 presents the real-time MABI position data, illustrating the raw and corrected trajectories along all three spatial dimensions. The stair-step pattern observed in each axis reflects the discrete movements of the robot, and the near-perfect alignment of the corrected and uncorrected values again confirms the accuracy of the system's real-time feedback and correction mechanism.

In our relatively short-range experimental setup, the tracker's noise floor is on the order of a few tens of microns, meaning the robot's final positioning error ($\sim 20 \mu\text{m}$ RMS) is nearly at the measurement system's noise threshold. This indicates excellent data alignment between the robot and tracker coordinate frames and confirms that the system's correction quality is limited primarily by sensor precision rather than algorithmic error. Another key performance metric is the feedback loop latency. Using timestamp logs on the MQTT messages, we measured the end-to-end round-trip delay from the robot's position update to the tracker measurement and back to the robot with a correction. The mean loop latency was found to be approximately 80–85 ms, with a standard deviation of only a few milliseconds. In all experiments the latency consistently stayed below 100 ms, as designed. This corresponds to an update rate of about 10–12 Hz for corrective commands, which is sufficient for the quasi-static and moderately dynamic motions tested. The low latency and jitter observed highlight the efficiency of the Factory+ MQTT broker architecture in routing data with minimal overhead. The system's real-time feedback loop did not exhibit any oscillation or instability; each correction was applied smoothly, bringing the robot closer to the desired position without overshoot. We attribute this stability to the modest magnitude of each correction (generally sub-millimeter) and the robot controller's ability to blend the

incoming adjustments via the standard OPC UA interface. Overall, the results demonstrate a fast and stable feedback loop: the robot and tracker remain tightly synchronized, and errors are caught and corrected within a fraction of a second, leading to a high degree of confidence in the robot's placement for precision manufacturing tasks. Furthermore, the system's feedback corrections were equally effective on both robot platforms, underscoring the generality and modularity of the approach. The Factory+ infrastructure allowed us to interface with two different robot controllers and the common metrology system through a unified data pipeline, without any special-case tuning for one robot or the other. This demonstrates that our MQTT-based communication and correction loop is robust across different robot types and kinematic configurations. It also suggests that the approach can be extended to additional machines with minimal reconfiguration, provided they are integrated into the same Factory+ messaging network. Overall, the real-time robot-tracker communication loop performed reliably in both experimental cells, achieving low latency (sub-100 ms) in a consistent manner. These results validate the efficacy of using an IIoT-centric architecture for metrology-driven robot calibration, and they highlight the potential of cloud-assisted feedback control in demanding manufacturing scenarios.

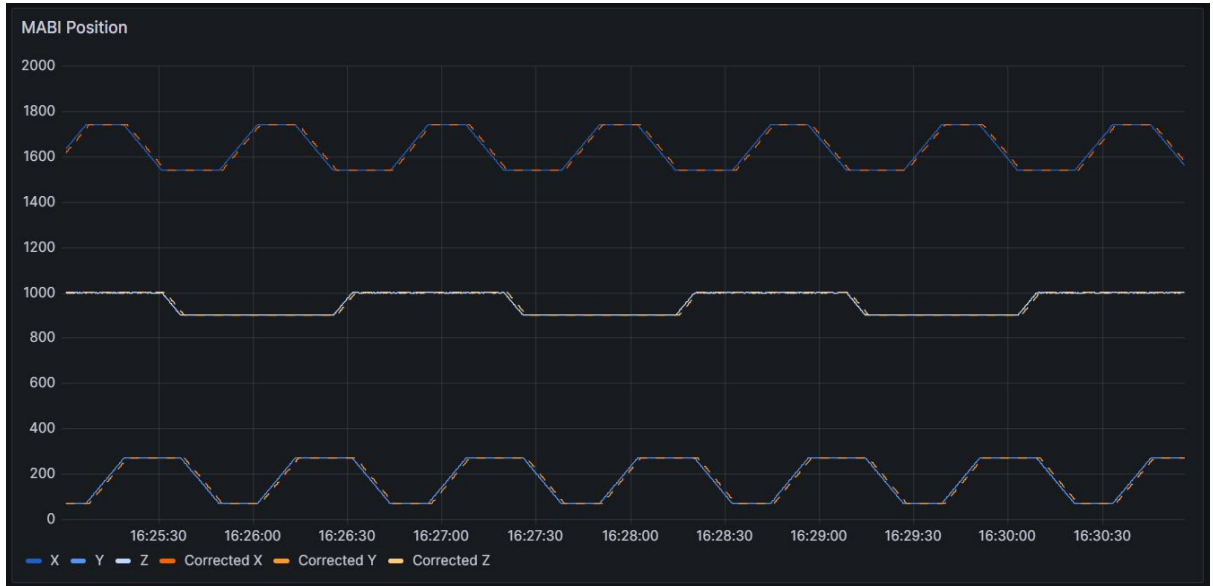


Figure 4: Real-time MABI Position data showing raw and corrected X, Y, and Z axis movements.

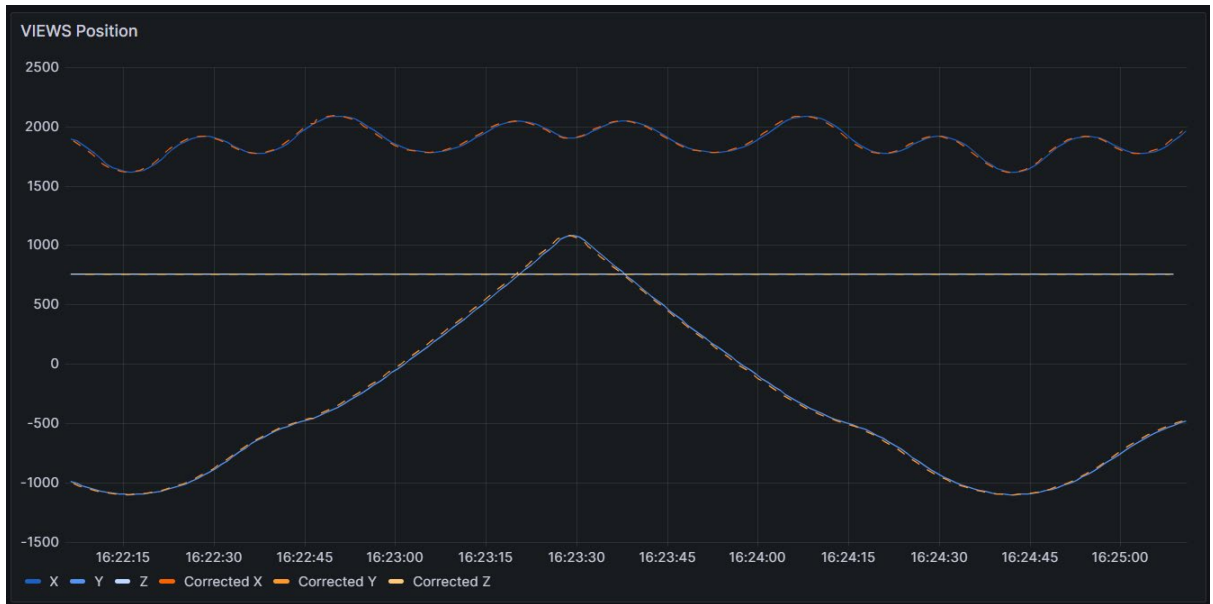


Figure 5: Real-time VIEWS Position data showing raw and corrected X, Y, and Z axis movements.

IV. CONCLUSION

The work has presented a cloud-integrated communication architecture for real-time metrology-based robot correction, leveraging the MQTT protocol within the AMRC Factory+ framework. The system enables industrial robot controllers and laser trackers to exchange data bidirectionally with minimal delay, orchestrated through a lightweight publish/subscribe messaging network. Experimental validation of the prototype system demonstrated that the approach meets the stringent requirements of precision manufacturing. In particular, the closed-loop robot-tracker coordination achieved an average latency well below 100 ms per feedback cycle and delivered, even during dynamic movements. These key findings confirm that real-time coordination via MQTT is

not only feasible but highly effective for high-precision applications. The use of the Factory+ cloud infrastructure proved invaluable in this context: it provided a modular, scalable integration of devices (edge agents, controllers, metrology systems) and a unified namespace for data, which simplified deployment across multiple cells. The successful implementation on two distinct robotic platforms (VIEWS and MABI) further highlights the flexibility and generality of the solution. In summary, the proposed system combines the benefits of industrial IoT connectivity with advanced metrology to dynamically calibrate and control robots in real time, effectively bridging the gap between cloud-based intelligence and on-floor precision control.

Looking forward, several future work directions can improve and extend the capabilities of the system. One important goal is to enhance the time determinism of the

communication and control loop. This could involve integrating real-time networking techniques or Quality of Service (QoS) settings for MQTT to ensure consistently low latencies even under heavier network traffic. Technologies such as Time-Sensitive Networking (TSN) or real-time middleware could be evaluated to bound the worst-case delay and jitter, thereby enabling the system to handle faster robot motions or tighter synchronization requirements. Another promising avenue is the incorporation of AI-driven adaptive calibration. Building on the initial success of the current feedback algorithm, we intend to explore machine learning approaches – for example, using an artificial neural network to predict and compensate for systematic robot errors in parallel with the tracker feedback.

In summary, this research lays a robust foundation for cloud-connected, metrology-guided robotic control in manufacturing. With further enhancements in network determinism, intelligent calibration, and multi-device orchestration, the approach can be scaled up to future Industry 4.0 factories where fleets of robots and smart instruments work in concert with unprecedented precision and coordination.

ACKNOWLEDGMENT

This work part of a collaboration with AMRC Sheffield, funded by the Engineering and Physical Sciences Research Council's Innovation Launchpad Network+ Researcher in Residence scheme [grant numbers EP/W037009/1, EP/X528493/1].

REFERENCES

- [1] O. Saha and P. Dasgupta, 'A Comprehensive Survey of Recent Trends in Cloud Robotics Architectures and Applications', *Robotics*, vol. 7, no. 3, 2018, doi: 10.3390/robotics7030047.
- [2] M. Aissam, M. Benbrahim, and M. N. Kabbaj, 'Cloud Robotic: Opening a New Road to the Industry 4.0 BT - New Developments and Advances in Robot Control', N. Derbel, J. Ghommam, and Q. Zhu, Eds., Singapore: Springer Singapore, 2019, pp. 1–20. doi: 10.1007/978-981-13-2212-9_1.
- [3] S. Asif and P. Webb, 'Managing Delays for Realtime Error Correction and Compensation of an Industrial Robot in an Open Network', *Machines*, vol. 11, no. 9, 2023, doi: 10.3390/machines11090863.
- [4] K. Govindaraj and A. Artemenko, 'Container Live Migration for Latency Critical Industrial Applications on Edge Computing', in *2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2018, pp. 83–90. doi: 10.1109/ETFA.2018.8502659.
- [5] W. Xu, J. Cui, L. Li, B. Yao, S. Tian, and Z. Zhou, 'Digital twin-based industrial cloud robotics: Framework, control approach and implementation', *J Manuf Syst*, vol. 58, pp. 196–209, 2021, doi: <https://doi.org/10.1016/j.jmsy.2020.07.013>.
- [6] ARMC, 'AMRC. Factory+ Framework. Factory Plus portal'. Accessed: Jul. 25, 2025. [Online]. Available: <https://factoryplus.app.amrc.co.uk/>
- [7] Sparkplug, 'The Sparkplug Specification'. Accessed: Jul. 25, 2025. [Online]. Available: <https://sparkplug.eclipse.org/specification/>



DSPACE

<https://dspace.org/>

Enabling real-time robot-tracker communication via MQTT on the factory cloud for metrology automation

Asif, Seemal; Hall, Jon; Sawyer, Daniela

2025-10-29

Institute of Electrical and Electronics Engineers (IEEE)

<https://doi.org/10.1109/ciot67574.2025.11410153>

Asif S, Hall J, Sawyer D. (2025) Enabling real-time robot-tracker communication via MQTT on the factory cloud for metrology automation. In: Proceedings of the 2025 8th Conference on Cloud and Internet of Things (CIoT), 29-31 Oct 2025, London, UK