UKAEA-RACE-PR(23)02

Ipek Caliskanelli, Matthew Goodliffe, Craig Whiffin, Michail Xymitoulias, Edward Whittaker, Swapnil Verma, Robert Skilton

Engineering Interoperable, Plug-and-Play, Distributed, Robotic Control Systems for Futureproof Fusion Powerplants Paper

Engineering Interoperable, Plug-and-Play, Distributed, Robotic Control Systems for Futureproof Fusion Powerplants Paper

Ipek Caliskanelli, Matthew Goodliffe, Craig Whiffin, Michail Xymitoulias, Edward Whittaker, Swapnil Verma, Robert Skilton

This is a preprint of a paper submitted for publication in Robotics

Article

Engineering Interoperable, Plug-and-Play, Distributed, Robotic Control Systems for Futureproof Fusion Power plants

Ipek Caliskanelli ‡ *, Matthew Goodliffe ‡, Craig Whiffin, Michail Xymitoulias, Edward Whittaker, Swapnil Verma and Robert Skilton

[1] UK Atomic Energy Authority, Remote Applications in Challenging Environments (RACE), Culham Science Centre, Abingdon, OX14 3DB; [ipek.caliskanelli, matthew.goodliffe, craig.whiffin, michail.xymitoulias, edward.whittaker, swapnil.verma, robert.skilton]@ukaea.uk

* Correspondence: ipek.caliskanelli@ukaea.uk

‡ These authors contributed equally to this work.

Abstract: Maintenance and inspection systems for future fusion power plants (e.g. STEP and DEMO), are expected to require the integration of hundreds of systems from multiple suppliers, with lifetime expectancies of several decades, where requirements evolve over time, and obsolescence management is required. There are significant challenges associated with integration, deployment, and maintenance of very large-scale robotic systems incorporating devices from multiple suppliers, where each may utilise bespoke, non-standardised control systems and interfaces. Additionally, the unstructured, experimental, or unknown operational conditions frequently result in new or changing system requirements, meaning extension and adaptation is necessary. Whilst existing control frameworks (e.g. ROS, OPC-UA) allow for robust integration of complex robotic systems, they are not compatible with highly efficient maintenance and extension in the face of changing requirements and obsolescence issues over decades-long periods. We present the CorteX software framework as well as results showing its effectiveness in addressing the above issues whilst being demonstrated through hardware representative of real-world fusion applications.

Keywords: remote handling; interoperable; control system;

Citation: Caliskanelli and Goodliffe et al. Title. *Journal Not Specified* **2021**, *1*, 0. https://doi.org/

Received:
Accepted:
Published:

1. Introduction

The Joint European Torus (JET) is the world's largest active magnetic confinement facilities (MCF). The JET [1] project was set up by EURATOM in the late 1970s in order to study the feasibility of controlled nuclear fusion. The experimental device has been operating since 1983 and comprises a toroidal shaped vacuum vessel of 3m major radius in which a plasma is created, heated to temperatures of up to 300 million degrees and controlled. The JET machine is based at the UKAEA Culham Science Centre.

Remote Applications in Challenging Environments (RACE) is a robotics lab within the UKAEA, an executive non-departmental public body, sponsored by the Department for Business, Energy & Industrial Strategy. RACE was established in 2016 to gather experience from 25 years and over 40,000 hours of remote operations and maintenance of JET, and explore how they could be used to help with wider robotics challenges.

Over the many years of JET operations, hardware has become obsolete, maintenance requirements have changed, operations have become more complex, and the original remote handling equipment has struggled to keep up. Maintenance has become more difficult as direct hardware replacements become rarer and system interdependencies and compatibilities restrict upgrades or the alternatives that can be used. In addition to this, systems that operate nuclear facilities are likely to contain large quantities of bespoke hardware. Training the operations workforce on bespoke components takes time and effort, and the associated cost for this task is very high.

RACE was early to identify the need for an interoperable, extensible, futureproof control system in order to meet the current and future requirements of nuclear applications. The control system framework used in this study - CorteX, builds upon the lessons learned, knowledge, and experience gained over multiple decades maintaining a nuclear facility, presenting a solution to potential challenges ITER and DEMO will face. More importantly, the control system framework used in this study is also capable of tackling many of the challenges faced by Sellafield, TEPCO or similar organisations in decommissioning and can be used to overcome existing issues in today's nuclear sector.

CorteX minimises operating personnel training requirements by providing a standardised user interface, agnostic to the robot hardware. Software maintenance efforts are also minimised when changing system functionality or replacing components, due to a modular, reconfigurable, extensible control framework architecture. This provides a high level of support for expanding facilities, and helps utilise the performance of the workforce, providing cross-deployment hardware and software compatibility.

The main goal of this research is to establish the effectiveness of CorteX in controlling long-term robotic systems. The proposed CorteX solution is tested on TARM: a 1980s built serial manipulator that has been used in ex-vessel actives in JET. The accuracy of multi joint positioning using CorteX is measured with the positions of individual joints of TARM that are captured using a Vicon motion capture system over five repetitive iterations. Experimental results illustrate high accuracy in positioning of the TARM using CorteX.

The paper is organised as follows. Section 2 provides brief background information on the available off-the-shelf market products in the field of robotic middlewares and control systems that are considered by the nuclear sector. Section 3 is split into two parts: Section 3.1 covers CorteX - future-proofing, interoperable framework used in this study; Section 3.2 describes the TARM - the 40 year old ex-vessel manipulator, its evolution over time, and the challenges faced when operating old facilities. Section 4 describes the evaluation methods used to measure the accuracy of CorteX's control of the TARM. The paper is closed with the main conclusion of this study in Section 5.

2. Background

Robot Operating System (ROS) [2] is an open-source, multi-lingual platform, that provides a modular, tool-based, re-usable system and it is primarily used within the academic community. Over the years, it has gained popularity and, in some cases, has been accepted for non-critical industrial applications where time-criticality, mission-criticality, safety-criticality, and QoS are not required. Given these requirements are fundamental to most nuclear applications, ROS is not adequate. Open-source platforms, such as ROS, also bring up potential security threats due to the exposed code which may be exploited, creating another concern for nuclear applications.

ROS provides a structured communications middleware layer which is designed around commonly used sensory data (e.g. images, inertial measurements, GPS, odometry). Although the structured messages promote modular, re-usable software, ROS messages do not cope with the continuously evolving nature of software, causing compatibility issues. The highly coupled solutions created in ROS create issues for long-term maintainability and extensibility - crucially important factors for large scale industrial systems. Integration of ROS components is fairly easy for small-scale projects, but isn't a practical solution for large-scale engineering problems, due to the effort required for integration and modification when the system configuration changes (i.e. not easily extensible).

The second generation of Robot Operating System [3], ROS2, provides deterministic real-time performance in addition to the existing ROS features. Proprietary ROS message formats are converted into Distributed Data Service (DDS) participants and packages; thus providing a high-performing, reliable communication backbone which helps to achieve determinism at the communication layer. In order to facilitate discoverability

ROS2 inherits this functionality from DDS and is therefore heavily coupled and dependant on this service. ROS2 is backwards compatible with ROS via message converters, which inherit limited discoverability, causing almost non-existent interoperability and creating highly coupled solutions, making it very hard to extend any ROS system. Although ROS2 has resolved the reliability, timeliness, determinism and high-fidelity issues ROS previously suffered from, it has not resolved the maintainability and limited re-usability issues for large-scale engineering problems, as there is no change to the strict message structures.

Fieldbus protocols (e.g. EtherCAT, Modbus, PROFIBUS, Control Area Network (CAN) bus, serial communications) are standardised as IEC 61158 for industrial use and are used to interface to various pieces of hardware. The fieldbus network (e.g. TwinCAT/EtherCAT master, Modbus, PROFINET, Control Area Network (CAN) open, OPC-UA) technologies are used when a network of hardware is required, as opposed to a single point-to-point interface.

In order to control the hardware accessible by these fieldbus networks, an interface must be provided between the fieldbus and the control system. TwinCAT is one of the more appealing solutions, as it built upon an EtherCAT master specifically for this purpose. Another commonly used option in industrial applications is OPC Unified Architecture (OPC-UA). A machine-to-machine communication protocol used in industrial automation under IEC 62541 specification, to provide an interface between PLC level hardware and control software, such as ROS/ROS2.

In order to achieve a distributed control system, the information from a local machine has to be distributed over a network. Middleware such as DDS, OPC-UA, MQTT, and ZeroC ICE can be used to communicate information from a local machine to other networked devices. The data-centric Pub/Sub protocol Data Distribution Service (DDS) OpenSplice [4] offers highly dynamic, timely, reliable QoS. Device-centric OPC-UA [5] provides a standardised communication protocol and allows users to organise data and semantics in a structured manner, which makes OPC-UA an interoperable platform for multi-vendor, industrial systems. To ensure interoperability and increase re-usability, standardised but extensible base message types are provided by the OPC-UA Foundation. From this perspective, OPC-UA is the most similar middleware to CorteX, however it does not provide control functionality.

The Message Queuing Telemetry Transport (MQTT) [6] protocol provides a lightweight and low-bandwidth approach which is more suitable for resource-constraint internet-of-things (IoT) applications and machine-to-machine communications; it is orthogonal to OPC-UA, but not interoperable like OPC-UA. ZeroC ICE [7] provides a remote procedure call (RPC) protocol that can use either TCP/IP or UDP as an underlying transport. Similar to DDS, MQTT, and OPC-UA, ZeroC ICE is also a client-server application. Although asynchronous, the event-driven nature of ZeroC ICE makes it unsuitable for real-time applications where QoS and durability are key; the same characteristics help improve scalability. Its neatly packaged combination of a protobuf-like compact IDL, an MQTT-like architecture, broker executables, autodiscovery features, and APIs in various languages make ZeroC ICE a popular middleware choice for non-real-time applications. Createc Robotics has been developing Iris [8], an open platform for deployment, sensing, and control of robotics applications. Iris combines 3D-native visualisation, a growing suite of ready to use robotics applications, and system administration tools for application deployment. As a platform, Iris intends to introduce an open standard designed to enable interoperable robotics and telepresence system modules. However, Iris message types do not implement type introspection effectively, which creates the same limitations as ROS, and ROS2.

Supervisory Control And Data Acquisition (SCADA) [9] networks play a vital role in modern critical infrastructures such as power generation systems, water plants, public transports, gas, and oil industries. In SCADA networks, data acquisition systems, data transmission systems and Human Machine Interface (HMI) software are integrated for providing the centralized monitoring and control system for processing inputs and outputs. SCADA networks are also utilised for collecting field information, transferring it to a central computer facility, and displaying the information for users graphically or textually. As a result, it allows the users to real time monitor or control an entire network from a remote location. Despite the many advantages of SCADA, its monolithic nature creates a single point of failure potentially causing severe security issues.

Nuclear industry is extremely hesitant towards using open-source, low TRL control systems frameworks due to safety and security concerns. Therefore, this paper does not include a review on valueable, blue-sky academic research. Instead, this section reviews commercially available off-the-shelf technologies (COTS) that are widely used in nuclear sector. Academic survey papers such as [10–13] analyse the features of some of these COTS products and demonstrate their performance, comparatively.

CorteX is designed from the ground up to work as a decentralised, distributed, interoperable control system compatible with pub/sub, service-oriented applications. Although DDS is used to distribute information across a CorteX network, the discoverable, self-describing, interoperable, functionality of the CorteX protocol is middleware agnostic, and is therefore not dependant on DDS for this functionality. The homogeneous structure of CorteX's simplex, and the standardised interface allows all components of a CorteX system to be inherently interoperate at a basic level. When combined with the ontological type model, the syntactic and semantic meaning of the simplexes can be standardised to increase interoperability between systems and allow enhances discoverability of morphologies. CorteX offers a significant advantage over solutions that use strict message types in order to standardise communication (such as ROS/ROS2), which results in a highly coupled solution with a limited level of scalability and extension. CorteX is designed to expand both in size and functionality as the long-term requirements of nuclear applications demand.

3. Control of the TARM robot as a case study for the CorteX control system

3.1. CorteX Design

CorteX is a long-term maintainable and extensible robotic framework that provides an interoperable communication standard, control methods applicable to current robotic technologies, and validation routines to test the stability of the developed platform.

To achieve long-term maintainability and extensibility, an ideal system infrastructure should implement two concepts: 1) loose coupling between components; 2) high cohesion of highly granular modular components. In order to achieve the ideal system infrastructure, CorteX uses a building blocks methodology. Required system functionality is provided by bringing together multiple common plug-and-play components called simplexes. Each simplex uses the same structure for internal data representation and has the same external interface. This data representation can be used as part of a communications protocol to allow distributed components of a single control system to exchange data without prior knowledge of each other. This means a CorteX control system can grow to incorporate new hardware and control features, while minimising the impact on the local system and without modifying other distributed components.

CorteX's self-describing distributed data model (see Fig.1 #1) consists of a collection of simplexes. Each simplex contains information, how it is connected to other simplexes in the system, available functionality, and an associate type. These types form a software ontology that contain rules regarding the syntactic information stored in each simplex and morphological rules to create standardised structures. The inherited nature of the ontology provides semantic meaning to the various control systems components. Within the CorteX framework, we use the ontology to build a common structure of domain-specific information, which when distributed can be reused with multiple components to make explicit assumptions about their purpose. In addition, the morphology is used to provide syntactic meaning and a create structures between components using types represented in the ontology. These structures are used to standardise distributed components and allow explicit assumptions to be made regarding the contents of a given system and facilitate interoperability.

The connections between simplexes are described using 'relationships'. A simplex's type can define not only the relationships it must have (to be considered of that particular type), but also how many (minimum, maximum, or absolute) simplexes must be related to it and their associated types. These relationship rules produce a system with a particular morphology, which is consistent between all systems using the same types. These morphologies tend to fall into one of two distinct groups: structural (e.g. a robot arm composes of a serial manipulator and a gripper) and behavioural (e.g. an inverse kinematics solver requires an input of a Cartesian position and outputs a number of joint angles).

CorteX is provided as a suite of libraries that can be easy integrated into C++ applications. CorteX Core provides the interoperable data and types models. CorteX CS extends the Core library to provide a high-performance control system environment. CorteX Toolkits contain simplexes capable of domain specific functionality and hardware interfaces, which are assembled together for each specific application. Finally, CorteX Explorer provides a graphical user interface to allow operators to view and command the CorteX control system, (see Fig.1 #5). This user interface may also be extended using CorteX Toolkits to provide more intuitive interfaces for various control system components and hardware, (see Fig.1 #6).

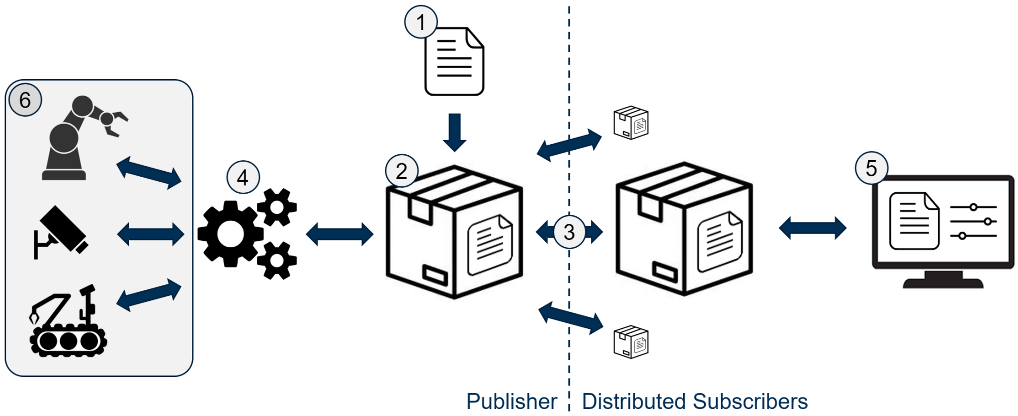

Figure 1. Suite of CorteX libraries - providing data and type models, communication, control, graphical user interface, and Toolkits for extension.

1. A standard way of describing systems,
2. A software implementation of this standard,
3. A scalable communication interface,
4. An extension to facilitate the control of these systems,
5. A graphical user interface for operating these systems,
6. A framework to facilitate the extension and expansion of these systems (e.g. hardware interfaces).

CorteX attempts to solve the main problems associated with interoperability and extensibility using a self-describing data representation. Standardised but extensible data interfaces are developed to provide interoperability, whilst semantic meaning is self-described by the components through types associated with a software ontology for robotic and control system components. To aid with structural interpretation in data exchange between these interfaces, software morphologies are implemented and used to provide syntactic meaning. The robotic and control system knowledge structure is distributed across the CorteX agents at run-time. We have a book chapter on CorteX, further details on CorteX design can be found in *CorteX: A software framework for interoperable, plug-and-play, distributed, robotic systems-of-systems* [14].

3.1.1. CorteX Quality Assurance

Encapsulation combined with loose coupling between components and high cohesion of fine-grained modular components, along with the use of standardised interfaces help in achieving modularity and testability. Quality and maintainability requirements are achieved by modern life-cycle management processes and effective component-based development techniques. CorteX is extensively unit tested providing a high level of code coverage as part of the software quality control.

Extensive analysis of CorteX memory profiling has been performed in order to ensure that CorteX is a lightweight framework. In addition to this, CorteX runs with a mostly static memory footprint to ensure minimum runtime allocation and memory leakage. Scalability, which is crucial to achieving extensibility, is evaluated and confirmed using memory tests ranging between 1 - 1000 simplexes and show acceptable linear growth. Timeliness and fidelity are important features of nuclear applications. Although CorteX is not a deterministic system, the deviation in latency, jitter, and loop cycle duration is less than 40 microseconds with a loop cycle of 1kHz. Based on real-time characterisation and applied software quality management, we believe CorteX delivers the performance and functionality required by long-term control system solutions for nuclear facilities. We consider CorteX to be TRL 6, as it has been demonstrated in a relevant environment.

3.2. TARM

Telescopic Articulated Remote Manipulator (TARM) is a 1980s built, 22 degree-of-freedom serial manipulator with payload of 600kg that was custom-designed to carry out ex-vessel maintenance activities at JET. It features a large vertical telescopic mast with a vertical movement range of up to 11m, and a horizontal boom with 8 DoF [15]. Highly dexterous, delicate in-vessel remote operations at JET are carried out by MASCOT. MASCOT is 1960s built, high-fidelity haptic master-slave manipulator, that allows the master operator to feel every action of the slave, from carrying a new component to tightening a bolt [16,17]. As the TARM was originally intended to carry out ex-vessel operations on the JET reactor, TARM has the capability of mounting the MASCOT manipulator as end-effector, either on the horizontal boom (where its labelled as the end-effector connection in Fig.2) or on the vertical mast (where its labelled as B3 on Fig.2). A 6 tonne capable Crane J on a rotational ring is positioned at the top of the TARM to perform heavy lifting. For example, if a heavy component would have been installed in the JET assembly hall, Crane J would have been used to lift the component in and out of position, whereas MASCOT would have been used for bolting or similar delicate and dexterous lightweight operations including connecting surfaces. Similarly, 250kg capable Crane K was designed to be used in the vertical mast only configuration, when MASCOT was attached to B3 in order to lift heavy items.

The J1 joint is used to rotate Crane J around the vertical mast. The B1 joint was used to rotate the entire TARM around a vertical axis for positioning around the torus. The full structure was designed to be mounted to an overhead crane in the JET building to allow 2D positioning within in the building.

Reduction in high radiation activities in the JET programme lead to lower radiation levels in the torus hall and allowed for manual interventions, causing the cancellation of ex-vessel remote operations and making the TARM redundant. Until its move to RACE in 2016, TARM was predominantly used for training the JET operators and supplying spare components for the JET machine. After RACE was formed in 2016 from the original JET remote handling unit, TARM was considered for repurposing. The JET machine and the TARM are unique in terms of robotics and control systems; they illustrate the effects of time on the requirements, hardware and technology, and project the importance of future-proofing in long-lived nuclear facilities. The JET machine is still in use and plasma experiments are still taking place, therefore, it is yet impossible to apply blue-sky remote manipulation research that can potentially harm the machine. However, the TARM:

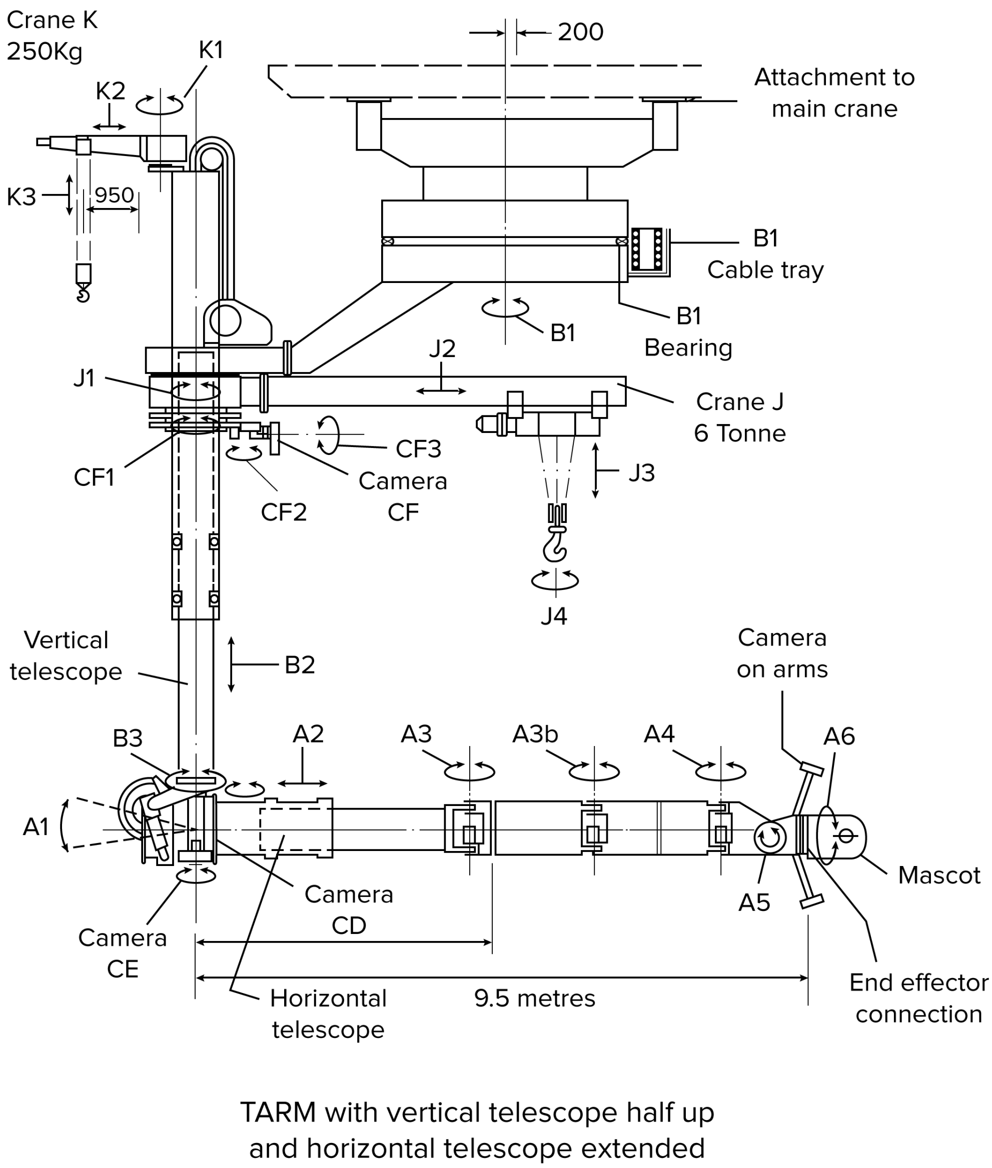

Figure 2. Joints labelled TARM back in original place at the JET Assembly Hall.

a 40+ year old, custom-built machine that suffered from changing requirements and hardware obsolescence as much as JET suffers, provides an experimental testbed for the early development of control and monitoring systems for long-lived nuclear facilities. Once TARM had been relocated to the RACE workhall (see Moving the TARM), the electrical and control components were either removed or upgraded. It is currently used as a test platform for a number of R&D projects, including APCS, RAIN and CorteX, and will be used to support developments for the JET 2024 campaign and IRTF programme.

TARM has a slightly different configuration and reduced capability at RACE as shown in Fig.3. The B1 joint and its crane attachments have not moved to RACE. Crane K (joints K1-3) has also been excluded in this new configuration. Crane J (joints J1-4) has remained as can be seen in Fig.3, however the joints have not been commissioned and are currently not in use. The A1, A2, A3, A3b, A4, A5 and A6 joints are commissioned and currently operational. Fig.4 illustrates an technician carrying out electrical checks on the end-effector connectors on the A6 joint before deploying MASCOT on the TARM.

All the A joints are operated with their original motors and gear boxes, however the electrical drive systems have been replaced with modern counterparts. Originally,

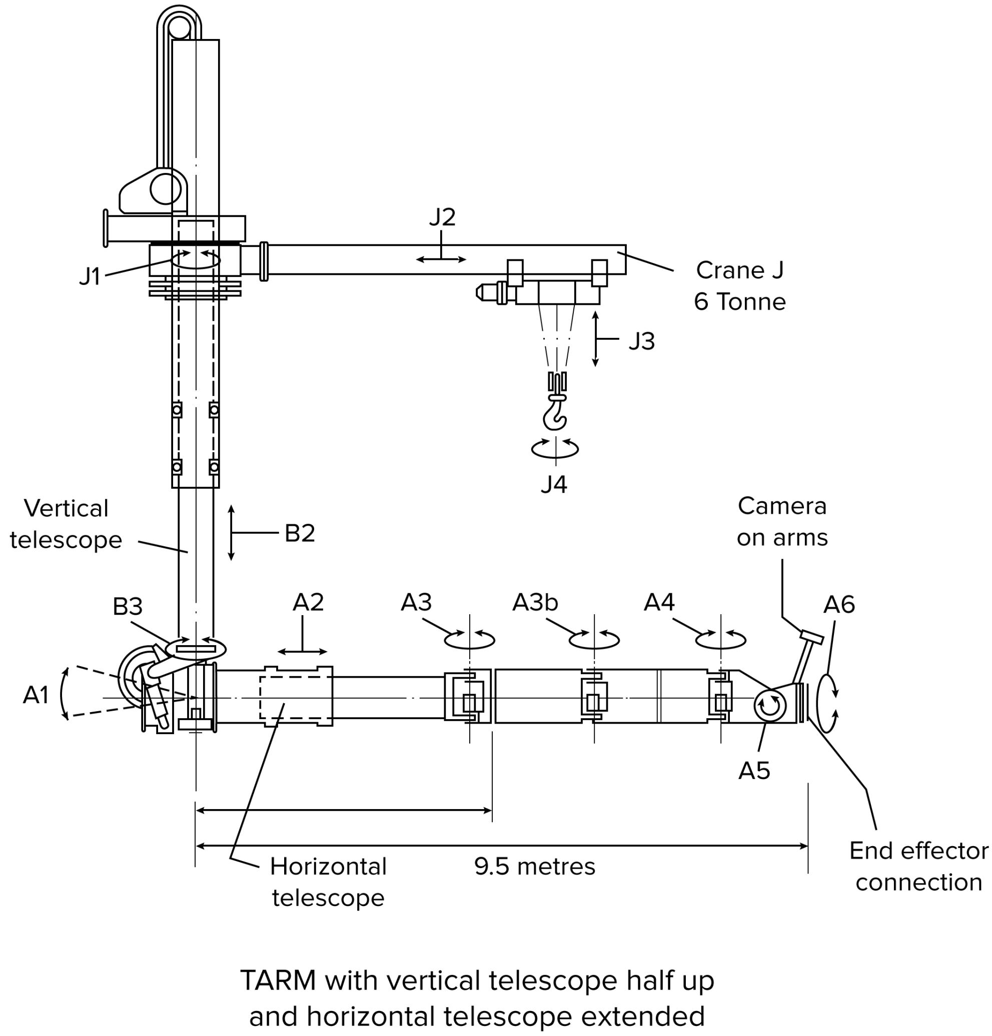

Figure 3. Joints labelled TARM the RACE building.

joint position feedback was provided by resolvers, which are now supported by modern encoders on the motors.

4. Experimental Setup and Performance Evaluation

4.1. Experimental Setup

A Viewing System comprising of multiple PTZ cameras, a control room with several monitors, and a video multiplexer was used for this case study. The TARM is located within the RACE workhall surrounded by PTZ cameras. In order to represent a real operations routine, the operations are carried out from a control room where the operators observe the TARM through the PTZ cameras, from the monitors in the control room. A video multiplexer is used to direct the video outputs onto the monitors in the control room. In this case study, CorteX is used to control the PTZ cameras and control camera to monitor assignment via an HMI. Fig.5 shows the live camera stream in the RACE workhall and Fig.6 illustrates the CorteX HMI which allows pan, tilt, zoom and focus functions for the cameras.

Figure 4. TARM at RACE, 2020.

Figure 5. RACE control room during the experiments: operators use the viewing system to make sure the moves complete safely.

Operations Management Systems is a RACE developed software tool that is designed and customised for remote operations. OMS is used to manage assets, people, tools, and tasks, and is fundamentally used to create and follow strict procedures (e.g. JET remote handling operations are carried out with an older version of OMS). It ensures required assets (people and tools) are available to perform specific tasks. Fig.8 and Fig.7 illustrates safety and operational procedures created for this case study using OMS. Fig.7 represents the full execution sequence applied by the operators. Sequential flow of the procedure starts with the initialisation of operations and involves safety checks, powering the PTZ cameras in the workhall, and clearing the TARM area. The operational area checks are followed by control room checks to ensure the system is safe and functional at which point the TARM operation can begin.

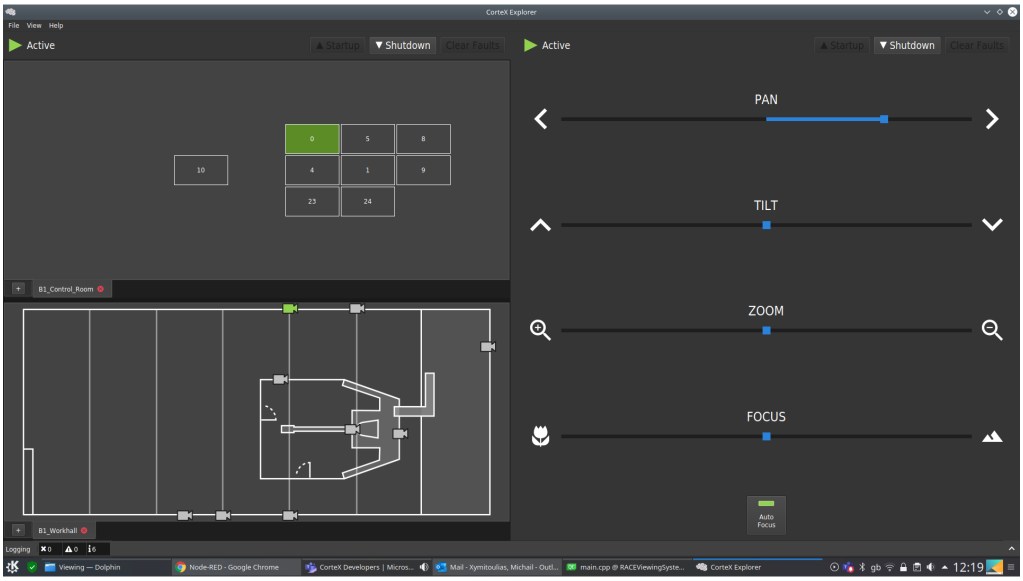

Figure 6. CorteX HMIs: RACE Workhall PTZ camera view illustrating camera selection on top left, selected camera placement in the RACE workhall in the bottom left and camera control functions such as pan, tilt, zoom and focus on the right hand side of the image.

Each rectangle in OMS can contain sub-procedures. For example, running safety checks involves a number of sequential processes such as powering the TARM safety cubicals, checking they function correctly and checking the emergency buttons are functional. In OMS, if a procedure has sub-procedural steps this is represented by a pink rectangle rather than the standard orange. The colour coded rectangular blocks help operators complete all the required steps and follow the procedure more efficiently.

Fig.8 illustrates the content of the 'Run Demonstration' sequence - one of the steps presented in Fig.7. Within this case study, the angular position demands of the TARM joints are set in OMS by the operator. Once the angle is set by the operator, CorteX facilitates the initiation of the move to the requested position. Once the move is completed and confirmed by CorteX, the OMS operator performs a visual check records completion of the procedure presented in Fig.8.

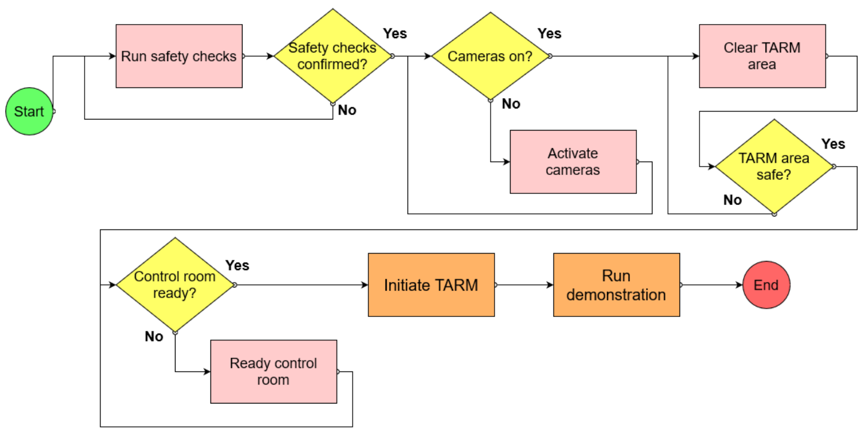

Figure 7. OMS TARM research sequence creator.

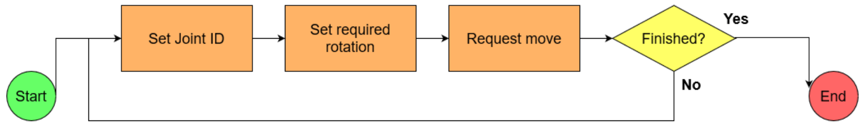

Figure 8. OMS sending position demands to TARM joint controllers.

CorteX Operator HMIs allow operators to observe the state and values within the CorteX control system. A number of custom HMIs have been specifically designed for the TARM operators for this research. At the beginning the Viewing System section, we mentioned the PTZ cameras of the viewing system are controlled by the CorteX control system. In order to present optimal views to the operators, the cameras must be frequently re-positioned to acquire maximum coverage of the current area of interest at any given time. For this purpose, the GUI shown in Fig.6 is designed to facilitate camera position and optical adjustment. The view is split into three sections: the monitor layout from the control room is show in the top left; a plan view of the RACE workhall including the TARM area, showing the physical camera locations is placed at bottom left; and PTZ and optical controls for the selected camera are shown on the right. When a camera is selected from the map (bottom right), any monitors currently displaying the feed from the selected camera are highlighted in the monitor display (top left). The pan, tilt, zoom, focus, and auto-focus controls on the right are also enabled if the selected camera is capable of these functions (i.e. PTZ rather than static). Camera to monitor assignment as achieved by dragging a camera from the map to the desired monitor and dropping.

Fig.9 shows a view used to observe and control a dual-axis EtherCAT DS402 (motor drive) device. The view is split into two halves (top and bottom) to display both axes: Axis A and Axis B, respectively. The left side of each axis view shows the current state of the axis from the *DS402 Observation*, the centre drive control area shows state values and control buttons for the *DS402 Processor*, and the right side shows the demand state of the drive from the *DS402 Modification*.

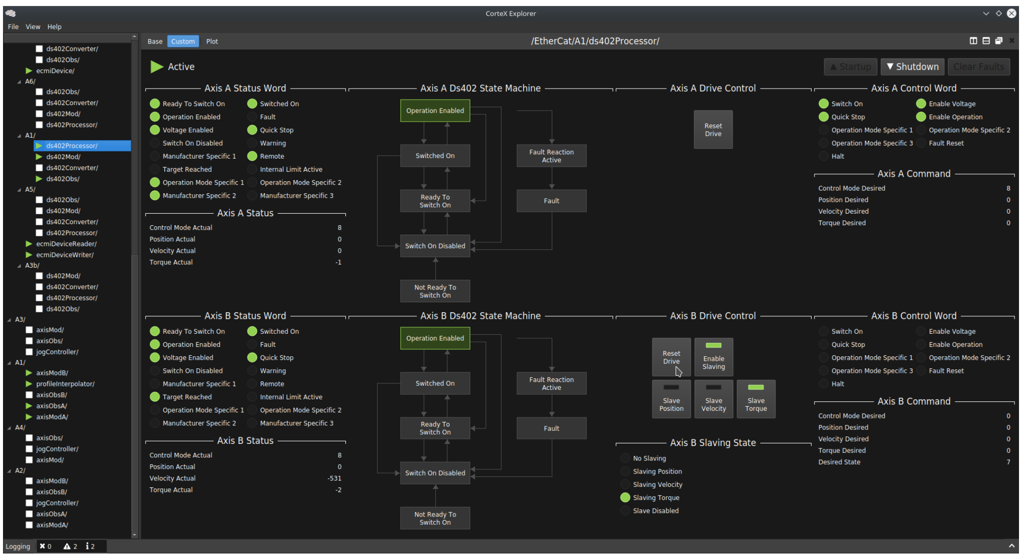

Figure 9. CorteX HMIs: Dual DS402 Processor View illustrating the simplex tree on the left, and dual axis status, state machine, control and demand state in a split view.

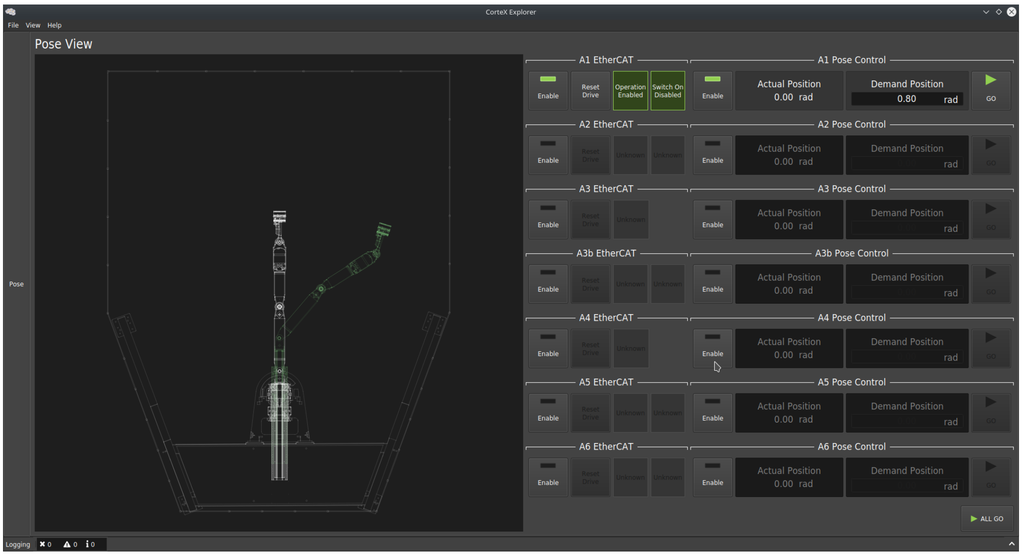

Figure 10. CorteX HMIs: TARM Pose View.

Fig.10 demonstrates several capabilities of the CorteX GUI framework. First, the view is split into two halves. The left side shows a representation of the physical TARM manipulator, posed to show not only the current position (in white) but also the target position (in green). This pulls data from a number of *Axis Concepts*, both observations and modifications, to pose the TARM image. This side of the view is specific to the TARM manipulator as the robot visualisation is currently only capable of rendering the TARM. However, the right side of the view is generic and can be used for any multi-axis manipulator. This selection of controls is generated dynamically by searching through the Simplex model for any *Axis Controllers*, and creating a set of controls for each type of controller - in this case, they are all *Axis Pose Controllers*. You will also notice to the left of each Pose Control area is an EtherCAT control area. The controls in this area were also auto-generated, using the morphology to discover the device *Processor* related to each *Axis Controller*, and then generate a view for the specific device processor type - in this case, an *EtherCAT DS402 Processor*. Notice how the EtherCAT control area discovers the number of axes each processor is controlling, producing two status displays for axes A1, A2, A3B, A5 and A6, but only one for A3 and A4.

4.2. TARM Performance Evaluation

It is challenging to control legacy equipment in an accurate and highly repeatable manner. For this case study, we used JET remote handling operations repeatability standard which is up to 10mm. JET in-vessel equipment have 10mm sub-accuracy, and the robotic moves can be controlled within 10mm reputability.

Repeatability analysis for this case study is carried out using a Vicon tracking system to precisely measure the level of accuracy and repeatability achieved after multi-joint positional moves. The Vicon tracking system at the RACE workhall consist of 12 Vicon cameras.

Tracking markers are placed on both ends of each link of the TARM, forming cuboid shaped objects. Tracking system measures the centre of the mass of the identified objects which indicates the centre of each link of the TARM. We demand fixed joint positions in CorteX during this study. Rotary joints A3 is set to -0.1 radius, A4 to 1.48 radius, and A3b to -1.36 radius. Linear joint A2 is set to 2.90 meters.

For simplicity, the decimal points of the link positions acquired by the Vicon system are not taken into consideration in this research. Moving the A2, A3, A3b, and A4 joints of the TARM, a sequence of defined moves was tested repeatedly both forwards and backwards, five times based on the set joint demands defined above. Results shown in

Table1 illustrate the deviation between repeated poses to be between 1 and 3mm. Please see this YouTube video of this study.

5. Conclusion

This paper presents the applicability of the CorteX control system for remote handling robotics using the case study of the TARM. CorteX is an interoperable, plug-and-play, distributed robotic system of systems. We developed CorteX to tackle the implementation of control systems for robotic devices in complex, long-lived nuclear fusion facilities. In Section 3.1 we explained the details of the CorteX design. A brief summary on the applied CorteX quality assurance is provided in Section 3.1.1.

TARM presents a unique application in terms of remote handling devices. There is no other 40 year old, ex-vessel remote handling equipment capable of 600kg loads in the world that is used as a testbed for R&D. The capability and uniqueness of the TARM is explained in Section 3.2. The past and current configuration of the TARM is also described in the same section.

In Section 4.1, the experimental setup used in this case study, including the viewing system, operations management system and CorteX HMIs is explained in detail. Custom-made, operator facing procedures are generated in OMS, and HMIs are developed for CorteX for this case study. A Vicon motion capture system is used to estimate the accuracy of the CorteX control systems framework on the TARM. The experimental results shown in Section 4.2 illustrate the deviation between repeated poses to be between 1 and 3mm. Based on the accuracy of the results in this case study, the adaptive and interoperable nature of CorteX, and the applied software quality management, we believe CorteX promises to deliver the needed control system solutions for long-lived nuclear facilities.

6. Author Contributions

Ipek Caliskanelli: Conceptualization of this study, Research Methodology, Software architecture, Software Evaluation.
Matthew Goodliffe: Software Lead, Software Quality and Assurance, Equipment Safety.
Craig Whiffin: Software Development, Operations.
Michail Xymitoulias: Software Development, Operations.
Edward Whittaker: Software Development.
Swapnil Verma: Software Development.
Robert Skilton: Conceptualization, sponsor.

7. Funding

CorteX is intellectual property of the UKAEA. This work is partly supported by the UK Engineering & Physical Sciences Research Council (EPSRC) Grant No. EP/R026084/1.

8. Institutional Review Board Statement

Not applicable

9. Acknowledgement

Special thanks to Matt Lobb for his assistance operating the TARM, Matthew Turner for his help in designing the OMS procedures for this case study, and Stephen Wells for his help in CorteX project and product management.

Joint	Pose	Link	Axis	Cartesian Position - World Frame (mm)										Max Diff.
				Iteration 1		Iteration 2		Iteration 3		Iteration 4		Iteration 5		
				Forward	Reverse	Forward	Reverse	Forward	Reverse	Forward	Reverse	Forward	Reverse	
-	Home	A6	X	-275	-277		-277		-277		-277		-277	2
			Y	5515	5515		5515		5515		5515		5515	0
			Z	1242	1242		1242		1242		1242		1242	0
		A3b Link	X	-175	-176		-176		-176		-176		-177	2
			Y	2455	2455		2455		2455		2455		2455	0
			Z	1692	1692		1692		1692		1692		1692	0
		A3 Link	X	-104	-104		-104		-104		-104		-104	0
			Y	397	397		397		397		397		397	0
			Z	1685	1685		1685		1685		1685		1685	0
		A2	X	-102	-102		-102		-102		-102		-102	0
			Y	398	398		398		398		398		398	0
			Z	1685	1685		1685		1685		1685		1685	0
A3	-0.1 rad	A6	X	-780	-781	-781	-781	-781	-781	-781	-781	-781	-781	1
			Y	5471	5471	5471	5471	5472	5472	5471	5471	5471	5472	1
			Z	1239	1239	1240	1240	1240	1240	1240	1240	1240	1240	1
		A3b Link	X	-376	-377	-377	-377	-377	-377	-377	-377	-377	-377	1
			Y	2437	2437	2437	2437	2437	2437	2437	2437	2437	2437	0
			Z	1691	1691	1691	1691	1691	1691	1691	1691	1691	1691	0
		A3 Link	X	-100	-101	-101	-100	-101	-101	-101	-100	-101	-101	1
			Y	396	396	396	396	396	396	396	396	396	396	0
			Z	1685	1685	1685	1685	1685	1685	1685	1685	1685	1685	0
		A2	X	-102	-102	-102	-102	-102	-102	-102	-101	-101	-101	1
			Y	398	398	398	398	398	398	398	398	398	398	0
			Z	1685	1685	1685	1685	1685	1685	1685	1685	1685	1685	0
A4	1.48 rad	A6	X	302	301	301	301	301	301	301	301	301	301	1
			Y	4732	4732	4732	4733	4733	4733	4733	4732	4732	4733	1
			Z	1247	1248	1248	1248	1248	1248	1248	1248	1248	1248	1
		A3b Link	X	-376	-377	-376	-377	-377	-377	-377	-377	-377	-377	1
			Y	2437	2437	2437	2437	2437	2437	2437	2437	2437	2437	0
			Z	1691	1692	1692	1691	1691	1691	1692	1692	1692	1692	1
		A3 Link	X	-100	-101	-101	-101	-101	-101	-101	-101	-101	-101	1
			Y	396	396	396	396	396	396	396	396	396	396	0
			Z	1685	1685	1685	1685	1685	1685	1685	1685	1685	1685	0
		A2	X	-102	-102	-102	-102	-102	-102	-102	-102	-102	-101	1
			Y	398	398	398	398	398	398	398	398	398	398	0
			Z	1685	1685	1685	1685	1685	1685	1685	1685	1685	1685	0
A3b	-1.36 rad	A6	X	-2447	-2447	-2447	-2447	-2447	-2447	-2447	-2447	-2447	-2447	0
			Y	3588	3588	3588	3588	3588	3588	3588	3588	3588	3588	0
			Z	1239	1239	1239	1239	1239	1239	1239	1239	1239	1239	1
		A3b Link	X	-347	-347	-347	-347	-347	-347	-347	-347	-347	-347	0
			Y	2451	2451	2450	2450	2450	2450	2450	2450	2450	2450	1
			Z	1694	1694	1694	1694	1694	1694	1694	1694	1694	1694	0
		A3 Link	X	-101	-101	-101	-101	-101	-101	-101	-101	-101	-101	0
			Y	397	397	397	397	397	397	397	397	397	397	0
			Z	1686	1686	1686	1686	1686	1686	1686	1686	1686	1686	0
		A2	X	-101	-101	-101	-101	-101	-101	-101	-101	-101	-101	0
			Y	399	399	398	399	399	399	399	398	398	398	1
			Z	1686	1686	1686	1686	1686	1686	1686	1686	1686	1686	0
A2	2.90 m	A6	X	-2573		-2573		-2573		-2573		-2573		0
			Y	6313		6316		6316		6316		6316		3
			Z	1189		1189		1189		1189		1189		0
		A3b Link	X	-472		-472		-473		-472		-473		1
			Y	5180		5180		5180		5180		5180		0
			Z	1654		1654		1654		1654		1654		0
		A3 Link	X	-226		-226		-226		-226		-226		0
			Y	3125		3125		3125		3125		3125		0
			Z	1658		1658		1658		1658		1658		0
		A2	X	-225		-225		-224		-224		-224		1
			Y	3127		3127		3127		3127		3127		0
			Z	1658		1658		1658		1658		1658		0

Table 1: TARM Joint positions gathered by the Vicon motion capture system.

References

1. Team, J. Fusion energy production from a deuterium-tritium plasma in the JET tokamak. *Nuclear Fusion* **1992**, *32*, 187.
2. Quigley, M.; Conley, K.; Gerkey, B.; Faust, J.; Foote, T.; Leibs, J.; Wheeler, R.; Ng, A.Y. ROS: an open-source Robot Operating System. ICRA workshop on open source software. Kobe, Japan, 2009, Vol. 3, p. 5.
3. Maruyama, Y.; Kato, S.; Azumi, T. Exploring the performance of ROS2. Proceedings of the 13th International Conference on Embedded Software, 2016, pp. 1–10.
4. Schlesselman, J.M.; Pardo-Castellote, G.; Farabaugh, B. OMG data-distribution service (DDS): architectural update. IEEE MILCOM 2004. Military Communications Conference, 2004. IEEE, Vol. 2, pp. 961–967.
5. Henssen, R.; Schleipen, M. Interoperability between OPC UA and AutomationML. *Procedia Cirp* **2014**, *25*, 297–304.
6. Banks, A.; Gupta, R. MQTT Version 3.1. 1. *OASIS standard* **2014**, *29*, 89.
7. Henning, M.; Spruiell, M. Distributed programming with ice. *ZeroC Inc. Revision* **2003**, *3*, 97.
8. Robotics, C. IRIS. https://www.createcrobotics.com/. Accessed:28-05-2021.
9. Barr, D. Supervisory control and data acquisition (SCADA). *Systems National Communications System, Communication Technologies* **2004**.
10. Profanter, S.; Tekat, A.; Dorofeev, K.; Rickert, M.; Knoll, A. OPC UA versus ROS, DDS, and MQTT: performance evaluation of industry 4.0 protocols. 2019 IEEE International Conference on Industrial Technology (ICIT). IEEE, 2019, pp. 955–962.
11. Cabrera, E.J.S.; Palaguachi, S.; León-Paredes, G.A.; Gallegos-Segovia, P.L.; Bravo-Quezada, O.G. Industrial Communication Based on MQTT and Modbus Communication Applied in a Meteorological Network. The International Conference on Advances in Emerging Trends and Technologies. Springer, 2020, pp. 29–41.
12. Mühlbauer, N.; Kirdan, E.; Pahl, M.O.; Waedt, K. Feature-based Comparison of Open Source OPC-UA Implementations. *INFORMATIK 2020* **2021**.
13. Silva, D.; Carvalho, L.I.; Soares, J.; Sofia, R.C. A Performance Analysis of Internet of Things Networking Protocols: Evaluating MQTT, CoAP, OPC UA. *Applied Sciences* **2021**, *11*, 4879.
14. Caliskanelli, I.; Goodliffe, M.; Whiffin, C.; Xymitoulias, M.; Whittaker, E.; Verma, S.; Hickman, C.; Minghao, C.; Skilton, R. CorteX: A Software Framework for Interoperable, Plug-and-Play, Distributed, Robotic Systems of Systems. In *Software Engineering for Robotics*; Springer, 2021; pp. 295–344.
15. Burroughes, G.; others. Precision Control of a Slender High Payload 22 DoF Nuclear Robot System: TARM Re-ascending.
16. Hamilton, D.; Preece, G. *Development of the MASCOT telemanipulator control system*; European Fusion Development Agreement. Project, 2001.
17. Snoj, L.; Lengar, I.; Cufar, A.; Syme, B.; Popovichev, S.; Conroy, S.; Meredith, L.; Contributors, J.E. Calculations to support JET neutron yield calibration: Modelling of the JET remote handling system. *Nuclear Engineering and Design* **2013**, *261*, 244–250.