applied system innovation

MDPI

Article

Realtime Calibration of an Industrial Robot

Seemal Asif * and Philip Webb

School of Aerospace, Transport and Manufacturing Cranfield University, Cranfield, Bedford MK43 0AL, UK
* Correspondence: s.asif@cranfield.ac.uk

Abstract: In large scale, complex and low volume manufacturing systems, robotics are now considered unavoidable for automating the factory operations. The aerospace industry focuses on a high variety and quality but extremely low volume. The precision it requires for numerous tasks is unique and distinct from any other manufacturing industry. This can comprise accurate position, module assembly, inspection, fastening, etc. The scale of the robot invites different types of errors during operation, which can be either because of the kinematics of the robot or because of the environment (noise, temperature, load, etc.). There are packages available from robot manufacturers for the correction and compensation of errors on the robot to achieve accuracy. There are two associated problems: 1. cost and 2. static nature. They are very costly and they do not provide correction in realtime fashion (dynamic); the robot stops, waits for the correction, and then moves to the next position. The external tool to monitor the accuracy also requires attaching with the robot permanently. These are dedicated resources. These tools for accurate measurement are expensive and attached permanently to a robot, which means wastage of resources. These measuring tools are called metrology devices and attaching these devices and the robot to the network means that other robots/machines can also use these expensive tools for measurement. Our aim was to address two problems in this project: 1. calibration (error correction and compensation of robot) and 2. dynamic and realtime processing. It helped to perform the dynamic error correction and the compensation of an industrial robot. The results showed the error correction was achieved in the region of 0.02 mm.

Keywords: real-time error correction; calibration; industrial robot calibration; kinematic calibration; laser interferometer; PID; MPC

check for updates

Citation: Asif, S.; Webb, P. Realtime Calibration of an Industrial Robot. *Appl. Syst. Innov.* **2022**, *5*, 96. https://doi.org/10.3390/asi5050096

Academic Editor: Subhas Mukhopadhyay

Received: 16 August 2022
Accepted: 29 August 2022
Published: 30 September 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Calibration is a technique that is used to maximise robot positioning accuracy. Owing to the cost and mechanical structure of industrial robots, it is becoming impossible to change the hardware structure of the robot to obtain improved positional accuracy. Precise accuracy was not very necessary in the beginning era of robotics, but because of the increase in robot population and the demands of industry for efficiency and repeatability of the robot, it is currently becoming a sought-after area of research.

There are numerous types of errors that can deviate a robot from its original intended position. Some are geometrical and some are nongeometrical, for example, backlash, wear and tear, etc. ISO 9283 also measures the robot performance by using various parameters that include accuracy, repeatability, pose, variation, drift, path, trajectory, etc. [1]. The project also tries to apply the calibration mechanism by using this standard, which guides use of a high-accuracy, large-volume 3D or 6DOF measurement system to test a robot's performance statically and dynamically.

There are ranges of errors that can affect the positional accuracy of the robot. These can be:

- Dynamic/systematic—via servo system, friction, inertia, etc.;
- Static—effected by the environment, e.g., load, temperature, and noise;
- Kinematics—related to the geometrical error of robot mechanics.

Calibration generally has three levels that depend on the application and its requirements [2]:

Level 1 includes the joint level that needs to determine the relationship between actual joint displacement and its signal generated by its transducer, which is known as mastering. This is similar to image restoration or factory reset on a machine.

Level 2 includes kinematics modelling. To perform this type of calibration, the actual rather than assumed kinematics/geometry of the entire robot must be well known.

Level 3 includes the nonkinematics or nongeometrical error handling, which affects the position of the robot; these can be friction, noise, temperature, etc.

In the early days of robotics, the maximum error of a robot was corrected using the mastering/remastering technique, and it is often recommended by robot manufacturers as well to remaster the robot after performing certain activities, for example, change of motor or gearbox. It is a known fact that 97% of positional accuracy can be achieved by removing the position-zero error of robots, and the solution is remastering [3].

However, it is becoming impossible to remaster the robot after performing every single task, and remastering should not be used to achieve positional accuracy without a reason, for example, severe collision, relocation or disassembly of the robot. Remastering deletes all the tool and base calibrations. Because of the change of era and demand for accurate position during runtime, especially in the aerospace sector, it is now becoming necessary to achieve the correct position during the execution of the program on the robot controller. Calibration needs to correct the position of the robot, which can be affected by different kinds of errors, ranging from wear and tear to kinematics.

2. Literature Review

Several researchers have taken different approaches to dynamic error compensation, in which the robot position is continually updated. The dynamic and near-realtime error compensation model was proposed by Nanyang Technological University, Singapore, and their experiment shows that they achieved the accuracy up to 25.5 μm for high-speed CMMs (coordinate measuring machines) [4]. They used a laser interferometer, which was directly attached to the CNC machine controller. The error was compensated for CMMs in this research.

Research by the Physikalisch-Technische Bundesanstalt, Germany, proposed a digital filter at the end of the sensor (for sensor-based error compensation), which was an accelerometer that helps to boost the speed and enable realtime error compensation [5]. The technique was used to filter at the accelerometer end, and to provide the filtered data for further calculation. The resultant algorithm could be useful for realtime calculations, but further filtering was required. The same technique can be used for the calibration calculation to boost the speed and optimise the data.

Research in the Department of Mechanical Engineering and Applied Mechanics, The University of Michigan, was carried out on the realtime error compensation technique for CNC machines [5,6]. The main focus was to test the techniques of multiple regression analysis (MRA) and artificial neural network (ANN) for the error compensation based upon analysing the geometric and thermal errors. In realtime, they were able to achieve (an improved performance of) 7.2 μm thermal distortions from 92.4 μm. For the ANN technique, a large number of data were needed to train the artificial neural networks; only then was it able to work correctly. Both of the techniques proved successful in an environment in which resources were not accessed through the network. The CNC machine has been discussed as it needs error correction as well and research groups have performed work on it to provide realtime error correction for CNC machines [4]. The CNC machine measurements provide the X, Y and Z position, which needs to be corrected along with the orientations. Robot position applies a similar concept in terms of end-effector, which has its X, Y and Z position and the rotations around X, Y and Z. Their axis may differ from application to application, but the general principle is almost the same. The robots and CNCs are controlled by a controller and both require programming. The robots can move

their arm in their working envelope and in different orientations can reach the position; however, CNC machines may have no more than 2 or 3 DoF. CNC machines are mostly used where a heavy machining operation is required and each CNC machine is designed to perform only one specific task, such as drilling, turning, milling, etc., whereas robots can perform multiple tasks with various different end-effector tools. Hence, CNC machines are not as flexible as robots. Both robots and CNCs can be controlled from their controllers by using a computer.

Research has been carried out at the University of Bath in the Department of Mechanical Engineering on the subject of realtime error compensation for a small three-axis machine tool using a laser tracker [7]. A 35 μm accuracy was achieved on the three-axis machine tool from 1.4 mm, which is 40 times reduction in the error. The error compensation system was deployed on the computer, which was also responsible for the collection of data from the laser tracker (FARO ION, Lake Mary, FL, USA) and the computer was directly attached to the three-axis machine tool. No control system or algorithm was specified in the research paper. They used a simple feedback loop to ensure the realtime performance feedback, but the figures for the time in which the improved accuracy were achieved are missing from the described work.

The model-based error compensation for five-axes CNC machines was proposed, which achieve 2 ms sampling time for position control [8]. The focus was on geometric errors using the ideal kinematic model. The algorithm used the Jacobian matrix [2,9] to form the linear relationship. The experiment was executed under ideal conditions.

Recent research has emerged as part of the MANU 602 project in the domain of realtime robot error correction [10]. The group was able to achieve positional accuracy of 0.05 mm by doing realtime error correction. The system is not a generic solution for kinematic calibration for general industrial robots. It is heavily based on FANUC's dynamic path modification (DPM) software package. C-Track was used for the sensor measurements, which is an optical CMM. The system is also dedicatedly performs the error correction locally as there is no evidence for it to work on a busy and open network. The system relied on the DPM, which makes its use specific to Fanuc.

Within a theoretical environment, realtime error compensation in 3DoF has been conducted by Harbin Institute of Technology, China [11]. They analysed the errors using Jacobian and applied BP neural networks to compensate for the errors in 1000 iterations. The experiment was performed in a simulation environment by using the MATLAB Neural Network Toolbox. They were able to achieve an accuracy of 0.08 mm in the simulation environment.

Research was carried out at the Dalian University of Technology, China, in 2015 for correcting the positional errors in a 6DOF industrial robot [12]. The work was executed in a simulation environment and they used the SIASUN SR165C (SIASUN Robot & Automation CO., Ltd, Shenyang, China) robot for kinematics study. They were able to reduce the total position error from 0.541 to 0.07 mm, which is an 87% reduction in the error.

For academic dynamic error correction, a study was carried out by a Canadian university, École de technologie supérieure, Montreal, for which a Mecademic's Meca500 robot with a 0.5 kg payload was used [13]. The system requires permanent attachment of the external measurement device and was able to achieve circular radial accuracy of about 0.100 mm. The effect of the system for linear paths with different payload robots is unknown.

Proportional–integral–derivative (PID) is the basic controller to be used in a control system that can handle the present and past, and can predict future errors based on the information received [14]. The MPC (model predictive control) is well known and used in a variety of industrial applications. For example, it performs very well where speed, time and optimization are the issues. It has been proven in a case study for analysing and maintaining water level, in which MPC is superior to PID when speed and performance are concerned [15]. In the study, it showed PID loss with 5.5% more packet delays as compared to MPC, and the difference increased over time. The packet delays/communication issues

affect robot performance by introducing delay in the overall process. If the robot is waiting for the error to come back and is static during that period, the static period can cost the factory line in both performance and expense. Hence it is important to compensate for the delays.

Most of the literature presented above dealt with industrial robots and related areas. The error correction has been widely researched, but mostly carried out with an expensive metrology device dedicated to operational use. The gap here is to combine the realtime error correction with the network so that the resources can be shared amongst other operations, thus significantly reducing cost and increasing flexibility.

3. Error Correction Framework

The system proposed in this thesis was developed and realised using an experimental approach. The framework is developed to handle the problem of realtime (dynamic) error correction and resource sharing. The system is developed in two stages:

1. Static error correction;
2. Realtime (dynamic) error correction.

3.1. Stage 1: Static Error Correction

The first step is to realise a system that starts with a static error correction, which is simpler to handle when compared to a realtime error correction.

The system contains multiple modules/subsystems along with databases, a laser tracker, and a robot controller. The system is aimed to work with any kind of metrology device and robot controller as long as it can obtain measurement data from a tracker and position feedback from the robot controller. A normal ethernet network is used to link the system together. The system is designed to be distributable, which can work in isolation (LAN) as well as through the network (campus network).

Figure 1 shows the architecture of the system. The network-based control system includes the PID. It receives the robot position from the robot controller and measured robot position from the Leica Tracker through the network. It gives the error feedback to the error correction and compensation module to perform the calibration procedure. It also gives the error feedback to the database for logging and analysis purposes. The error correction and compensation module then gives the compensated positional output to the robot controller so it can fix the robot position. It also gives the position output to the network-based controller to close the loop. The updated position is also saved in the databases.

The laser interferometer is the best choice for the project due to the flexibility required. The manipulator's arm needs to be measured during its movement. The interferometer device is required according to ISO 9283, where it requires the use of a high-accuracy, large-volume 3D or 6DOF measurement system [1]. The best available interferometer for the project is Leica Tracker AT901.

As the first stage of the project, the static error correction was developed. This system was developed using a UR5 robot to test the static error correction through the network. The system used the Leica Tracker with a reflector on in 3D. The system was capable of performing static error correction.

Figure 2 illustrates the architecture of the system that was developed. The system could perform the static error correction and achieved the first level of performance in the project experiment.

The CEC manager is the system developed to communicate between the Leica Tracker and UR5 Controller. It can also initialize the tracker and performs the bird bath procedure, which moves the tracker laser and head to look at the bird bath position on the tracker. It also sends the measurement points that the central system sends to the calibration module and the database. The UR5 controller also has the modules to interface with the CEC manager to send the robot position. It also has the module which updates the UR5 robot position based on the corrected position communicated by the CEC manager.

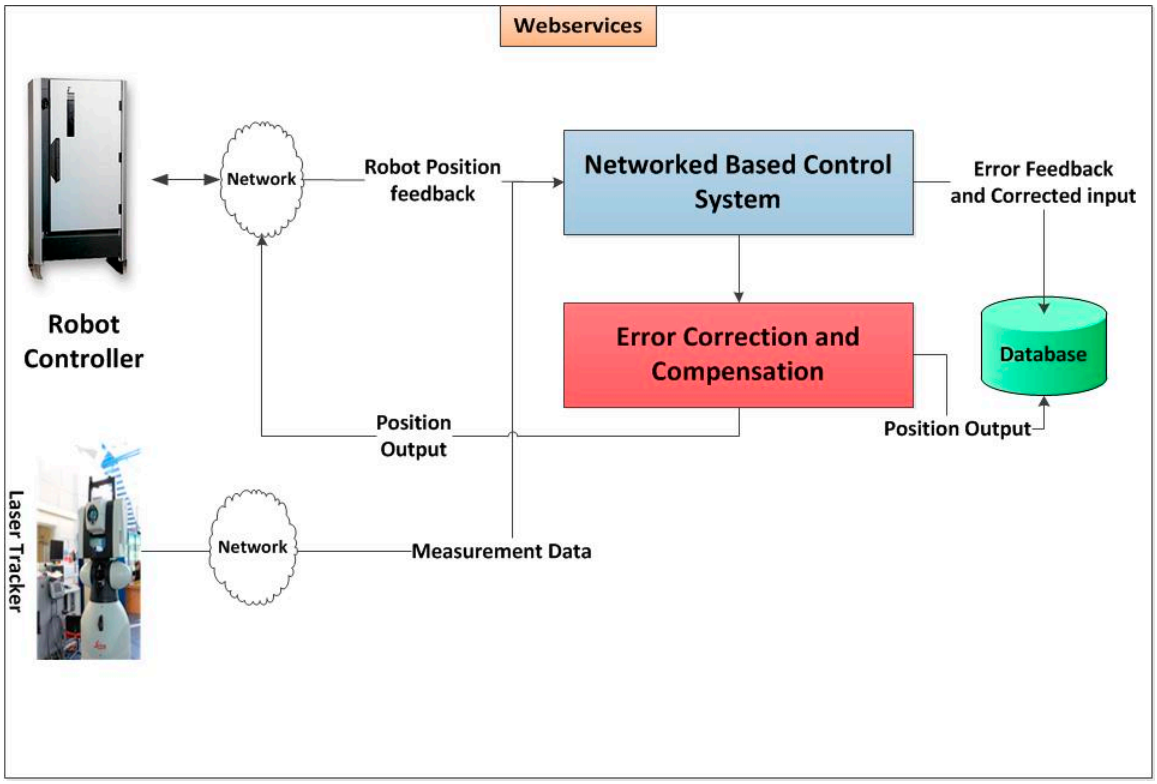


Figure 1. Basic system architecture.

CEC is the network-based control system, which includes the PID. It receives the robot position from the robot controller and measured robot position from the tracker through the network. It gives the error feedback to the error correction and compensation module to perform the calibration procedure. It also gives the error feedback to the database for logging and analysis purposes. The error correction and compensation module then gives the compensated positional output to the UR5 robot controller so it moves the robot arm to the corrected position. It also gives the position output to the CEC manager to close the loop. The updated position is also saved in the databases.

The physical cell is shown in Figure 3 (right side) and includes the Leica Tracker and the UR5 arm. The tracker tracks the UR5 robot arm's position by using a 1.5″ SMR (spherically mounted retroreflector) ball probe or the reflector bolt. The SMR is fixed to the robot end-effector (pneumatic gripper) by using glue. The SMR can only provide three-dimensional accuracy, which suffices for the requirements of stage 1 of the project. Only the positional accuracy can be dealt with at this stage. The flow chart of the static error correction for the system is shown on the right side of Figure 3. It shows that the robot has to stop for each waypoint to perform the error correction cycle.

The system is capable of connecting to the tracker through the network and can obtain static and continuous measurements. Continuous measurement means that the Leica Tracker is able to provide the measurements in dynamic fashion or continuous mode. This is called tracking mode, in which the tracker tracks the device (by continuously locking into the SMR, TMAC or TSCAN) and enables the tracker to provide the measurement in fractions of milliseconds. At this stage, the Leica Tracker reflector is used to obtain 3D measurements and so the correction is using a 3D plane. Figure 4 shows the interface of the integrated system for the user. The process is described as follows:

a. The user connects with the Leica Tracker using IP and port. The IP address and port number are required because the tracker is accessible through the network. "Tracker Connection" information turns green.
b. User initializes the tracker (if not pre-initialized after start).

c. The user gives bird bath using the tracker reflector.
d. The user presses the button "Ready to accept Robot Connection".
e. The user starts the robot program.
f. The "Robot Connection" information turns green.
g. The system then begins its process for correcting the errors.

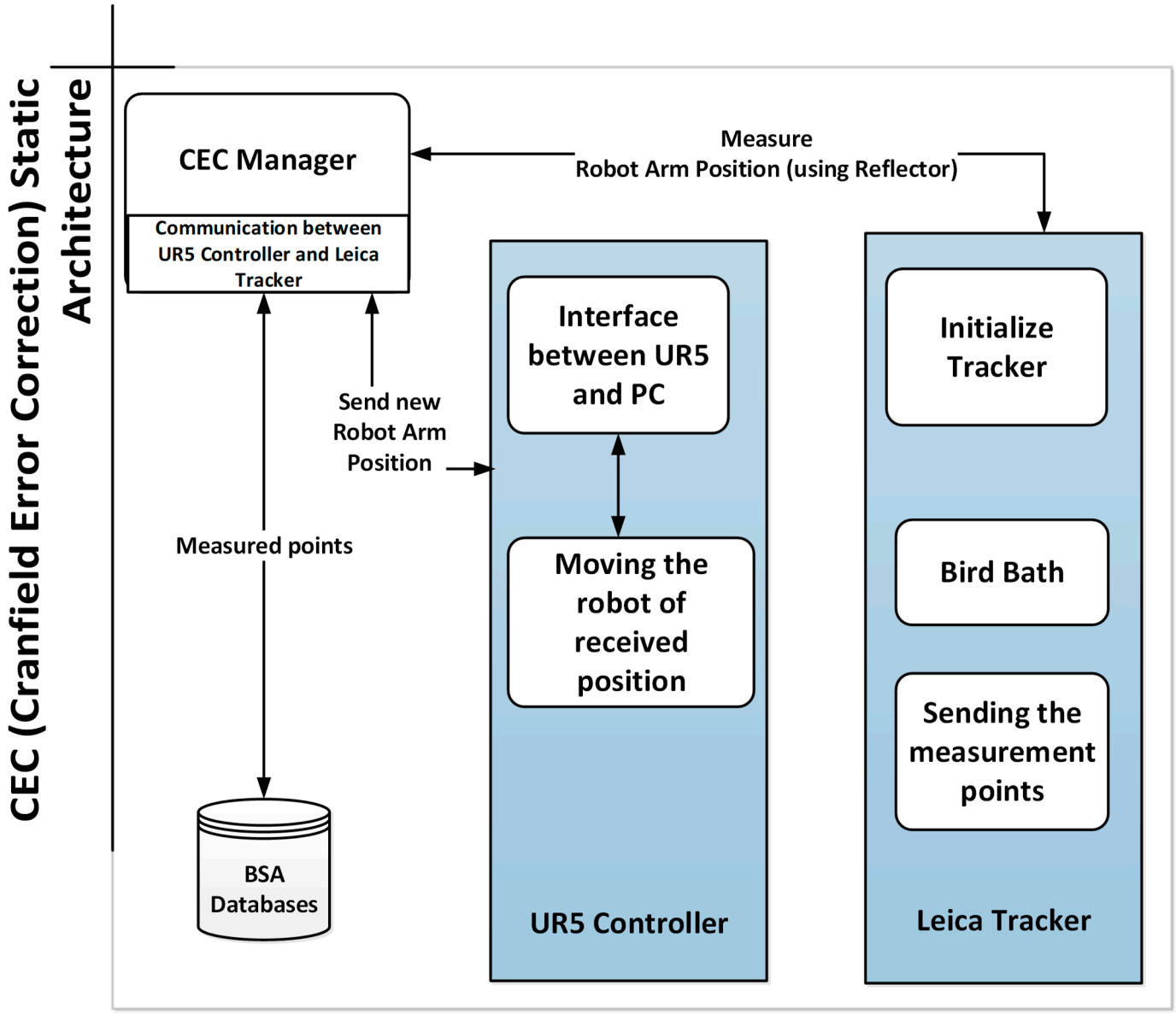


Figure 2. CEC system architecture.

3.2. Stage 2: Realtime (Dynamic) Error Correction

The UR5 controller CB2 has a limit of data exchange at 10 Hz for normal communication and 125 Hz for realtime communication [16]. The realtime 125 Hz frequency is not guaranteed by the Universal Robot [16]. The packets can be lost or missed using the realtime port, so the RTDE (realtime data exchange) software package is recommended for using critical realtime communication [17]. RTDE is only available for controller CB3 or above [17].

The best available robot to carry out the project at that time was a Comau NM45 with a C4G controller. The realtime position update can be enabled using the sensor tracking option for the C4G controller [18]. The ABB provides a similar tracking for dynamic error

correction through the add-on package called DPM (dynamic pose modification), but that also requires explicit control and a sensor mechanism for applying the error correction [10].

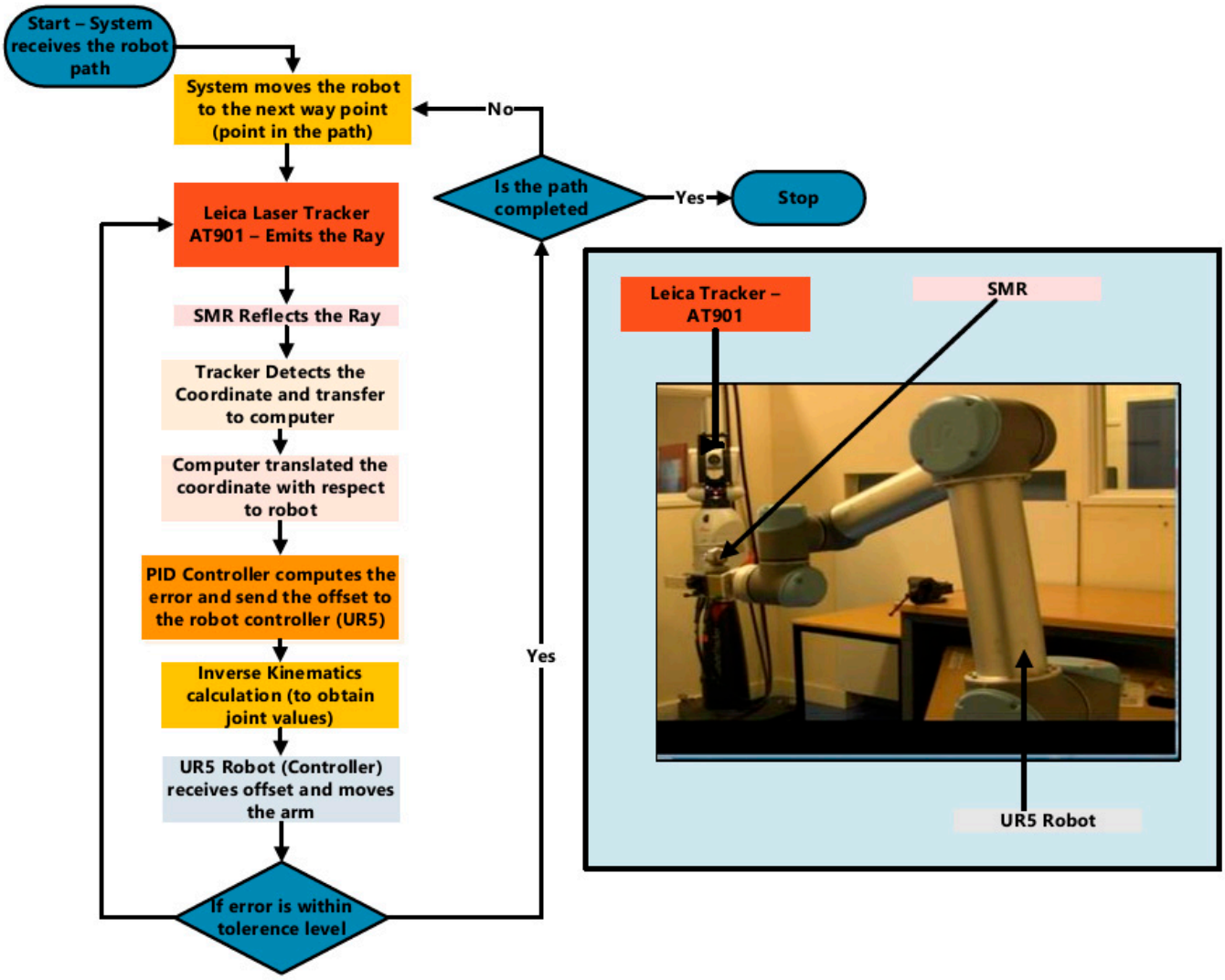


Figure 3. Small-scale system flow chart (**Left**) and physical cell (**Right**).

The NRTEC (network control system for realtime error correction and compensation) used the SoA (service-oriented architecture) design pattern. The system was developed using a modular approach through web services for the communication between those modules. Each module is independent of each other, yet communicate and contribute to the system to function. Figure 5 gives the basic architecture of the system. It illustrates the system flow and the components involved in the system.

The purpose of the NRTEC system manager is to receive the robot position from the robot controller and measured robot position from the Leica Tracker through the network. It also provides the error feedback to the error correction and compensation module to perform the calibration procedure. It also sends the error feedback to the database for logging and analysis purposes. The error correction and compensation module then gives the compensated positional output to the robot controller so it can fix the robot position. It also gives the position output to the network-based controller to close the loop. The database stores the information about the resources (e.g., Leica Tracker, robot), activities performed during the process, the output of the activities and the output of the whole process. It also stores the measured position of the robot, error and the position reported by

the robot controller. The compensated position as the result of the calibration process is also saved in the databases. Following are the major parts of the system:

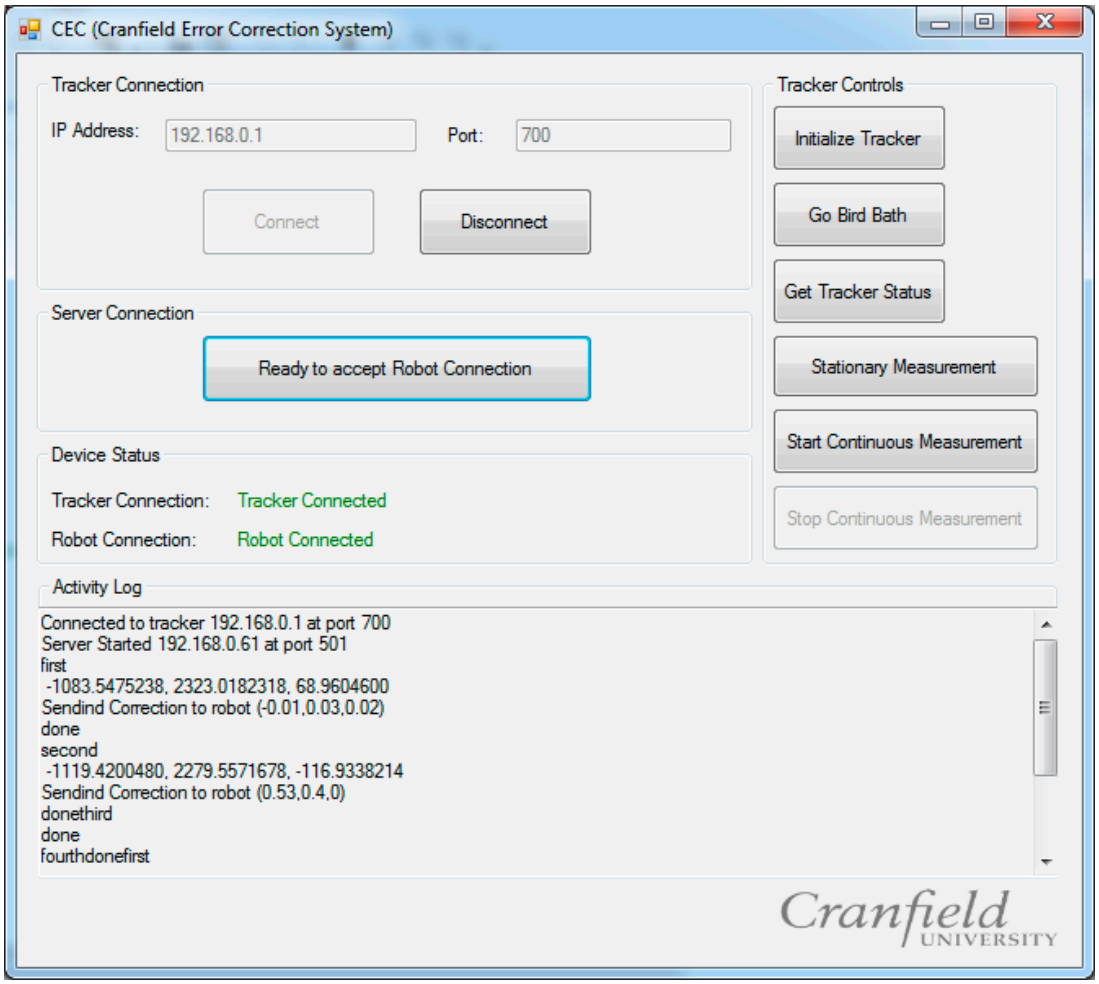


Figure 4. CEC for UR5 for static error correction.

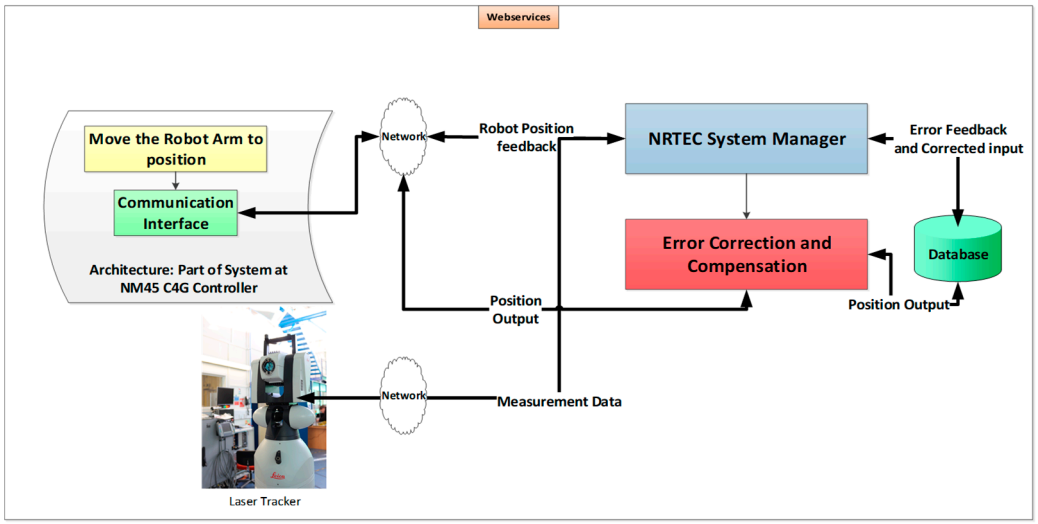


Figure 5. NRTEC basic architecture.

1. **Networked-Based Control System:** This includes the control system (MPC: model-based predictive controller) to form the feedback loop in the system. The major advantage it gives in NRTEC is its compatibility with realtime systems. PID is the simplest controller, to begin with, and was used until stage 1 for the static error correction. MPC is a multivariable controller [19], which means multiple variables can be controlled by implementing MPC in the system. It helps to control multiple variables at the same time and can apply adjustments to them. In our case, that is compensation. MPC is needed for specifically controlling the error in each variable. There are six variables involved in the positional and orientation error correction: X, Y and Z for the position and rotation around that axis RX, RY and RZ. MPC is implemented to control the error in each axis and its rotation.
2. **Calibration Calculation:** The main purpose of the system is to correct robot's kinematical errors, which can occur because of its geometry, weight, noise or temperature. The mathematical calibration model for kinematic positional error correction is used. The kinematics error correction is chosen to correct the geometrical errors in the robot system. The errors are calculated using the DH (Denavit–Hartenberg) model.
3. **Robot Control Programs:** The robot controller has a set of programs that help to provide data to the other modules such as NCS (networked control system) and the calibration calculation module. It holds the program for:
    i. Position feedback, which gives the current robot position—in the robot controller coordinate system.
    ii. Sensor Tracking enables the robot controller to change the position in case of error. It helps to change the current robot position whilst the robot is moving. It helps to run the dynamic error correction in realtime [18].
    iii. Communication Interface passes the information from the controller to the requested module and also receives the requests from the modules.
4. **Communication through the Network:** The NRTEC system is based on the concept of communication, which is performed through the network. TCP/IP is used as a protocol to perform this communication. Web services are used to communicate to the outside world system, such as FLEXA [20] (FP7 project, 2010–2012).
5. **Metrology System:** The metrology system is needed to determine the current position of the robot. Its feedback is crucial to determine the robot kinematical error. The Leica Tracker was chosen as the device for metrology work because of its proven accuracy.
6. **Database:** The database is used to store the steps performed by the system on each stage. It stores the robot position, metrology measurements, calculated errors and noise. All of this information is stored along with the timestamp details.

Figure 6 details the algorithm for the error correction and compensation side in the NRTEC system. The main error correction and compensation is run through the Compact RIO project. The MPC is implemented to control the complete process along with a Kalman filter. The system starts with the trajectory calculation for the robot path. The system provides the path to the robot and the robot start moving along the trajectory. The system then starts the concurrent process of running the error correction and compensation mechanism while the robot is moving. There is a concurrent loop inside the system in the Compact RIO, which handles the timesteps for error correction phases. These timesteps can be increased or decreased according to the requirement of the application. The system asks the measurement device to provide the measurement based on the timestep. The system then performs the calculations to transform the measurement into the robot coordinate system. Once the measurement is in the robot, the coordinate system for the forward kinematics is performed to determine the TCP/IP value, which is then compared with the robot controller's TCP (tool centre point) values. If there is a difference between values, then inverse kinematics is calculated to make certain the robot is aligned to the measured position. The inverse kinematics provides the robot join angles that are provided to the robot controller through sensor tracking, which makes certain that offsets are applied while the robot is moving. The output is also supplied to the neural network, which helps to

learn the system and its errors. Once the system learns its errors, the offset drops and there is no need to take measurements for the trained path.

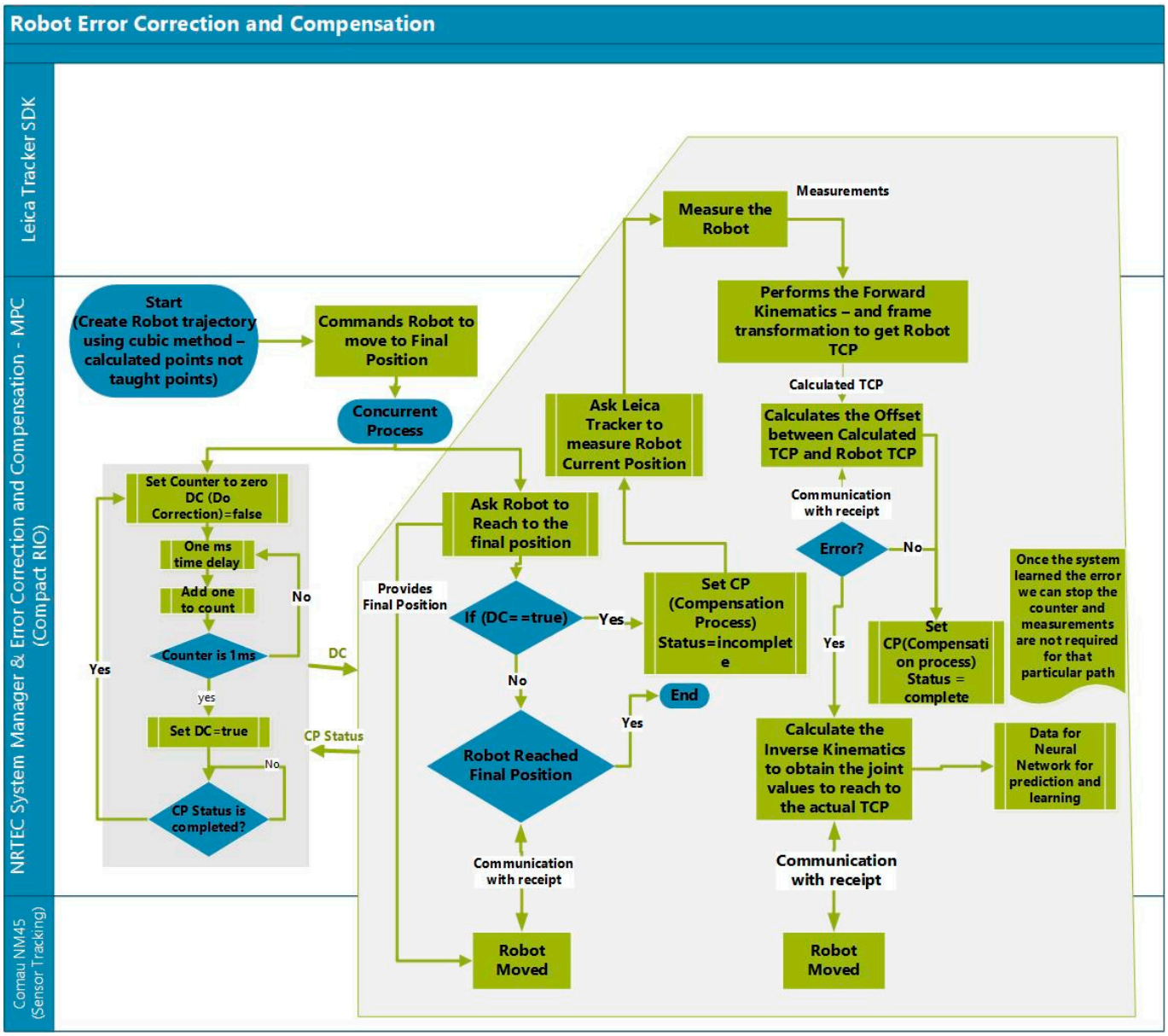


Figure 6. Algorithm (flow chart) for error correction and compensation.

4. Physical Cell

The physical cell was assembled with a Comau NM45 robot and its C4G controller, Central Control PC, Leica Tracker Control PC, Compact RIO, Switch, and the Leica Tracker. Figure 7 shows the layout of the physical cell.

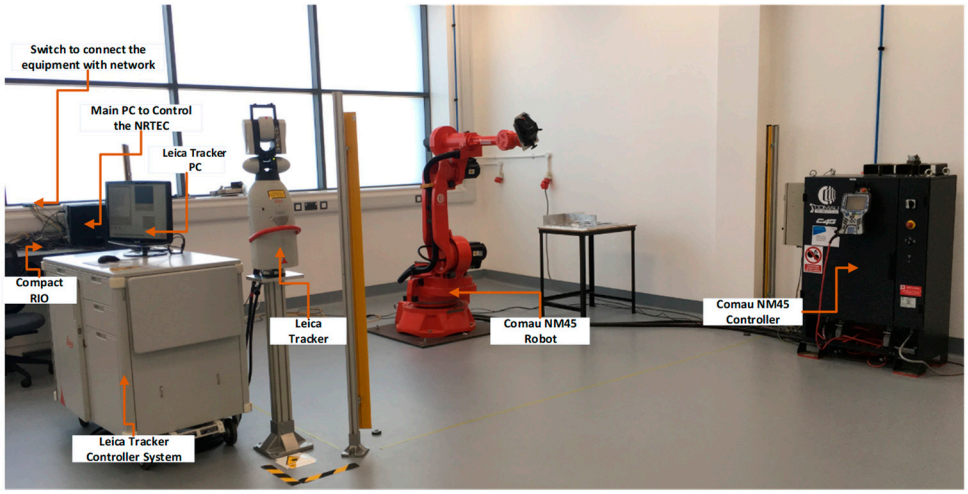


Figure 7. NRTEC physical cell.

5. Results

Repeatability tests have been performed as part of system testing. The standard repeatability of the Comau NM45 robot is ±0.06 mm [21].

Figure 8 shows the repeatability tests that were performed for the NRTEC system. Five cycles were performed and all of them repeated the points in the selected path. The path contained 18 points and each point is different from the others. The graph on left shows the path formed with 18 points is quite repeatable. The graph on right shows the histogram of the difference between these paths. The mean value by which they differ from each other is 0.005 mm. The use of the NRTEC system (Seemal Asif, Cranfield University, 2017–2018) improves the repeatability from 0.06 to 0.005 mm.

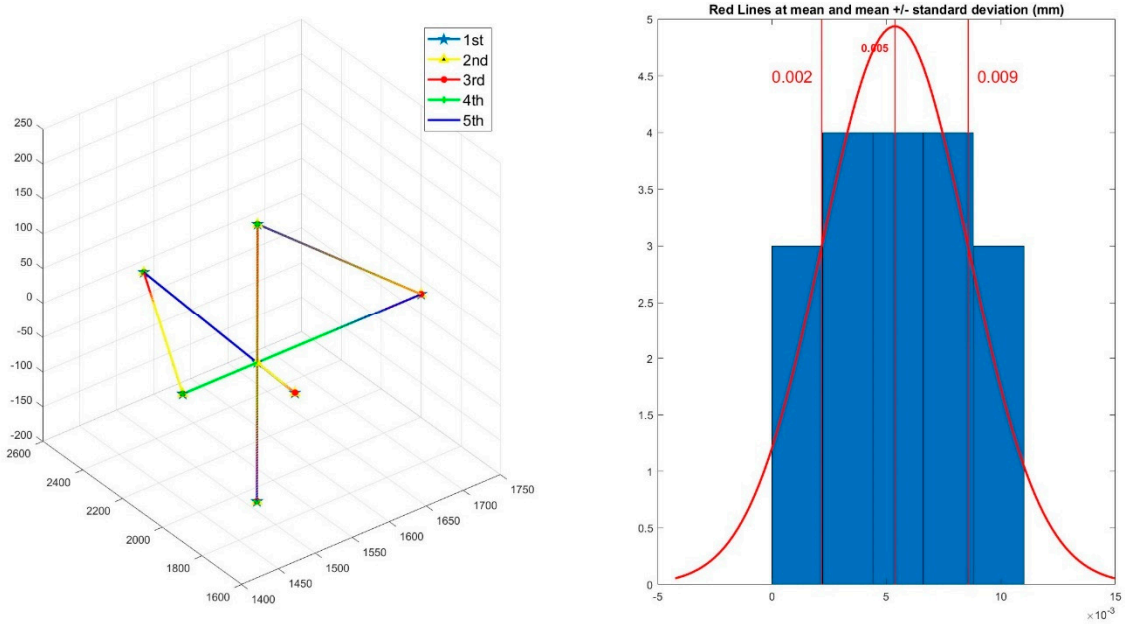


Figure 8. NRTEC repeatability test.

6. Point Correction Verification

The point's information was taken through the AT901 Leica Tracker, which was used to develop the NRTEC system. The graph in Figure 9 shows that the system was able to match the planned path between two points.

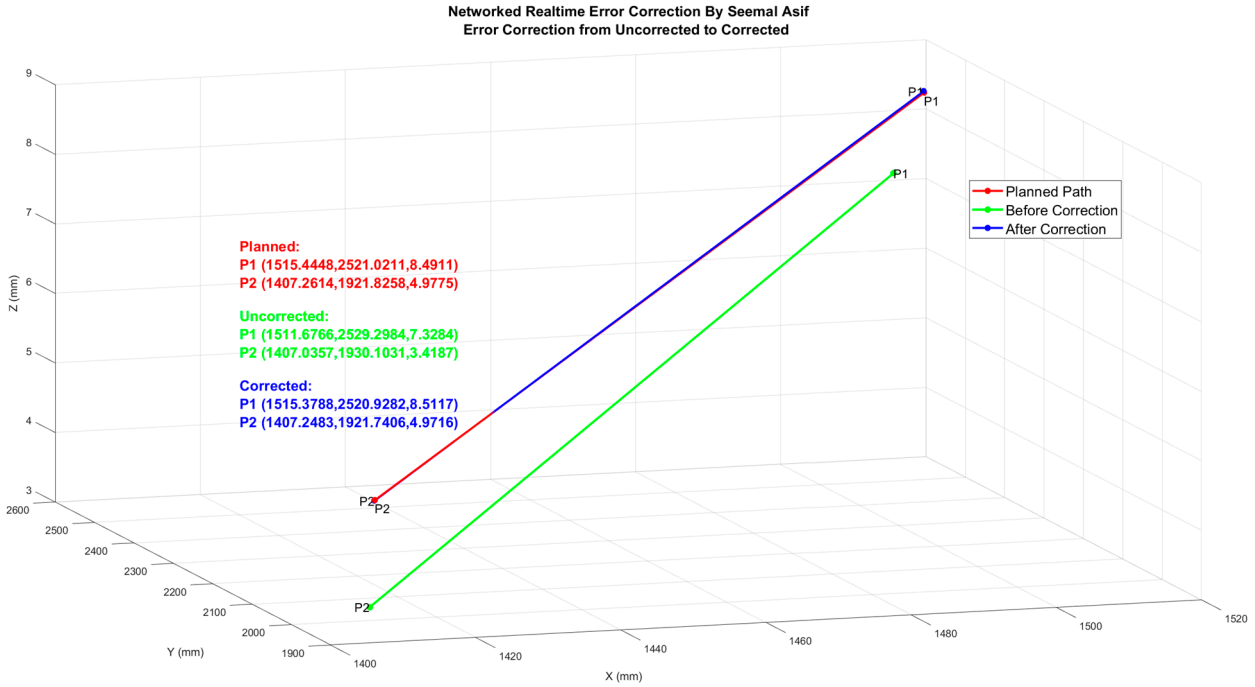


Figure 9. Error correction between two points.

Points p1 and p2 are the same points as reported by the robot, measured by the tracker, and corrected by NRTEC. "Planned" in red indicates the planned path that forms the robot trajectory, "Uncorrected" are the points where the robot reaches and "Corrected" are the same points which were corrected by NRTEC after receiving the information from the Leica Tracker (AT901) and sent back to the robot. The point selected is on the first instance where the system is able to be in the close proximity to the planned path. The system is further able to correct itself and achieve the correction within 0.02 mm.

For experimental purposes, 18 different positions were chosen to form the path, as shown in Table 1. There were positions in +ve and –ve of each axis (X, Y, Z) and +ve and –ve in rotation around each axis (RX, RY, RZ). The following table shows one of the samples of the positions in the robot reference frame that forms the path for the experiment.

Table 1. Set of positions (in robot reference frame) to form the path.

X (mm)	Y (mm)	Z (mm)	RX (deg)	RY (deg)	RZ (deg)
0	−1000	1600	−49.07	90	90
200	−1000	1600	−49.07	90	90
400	−1000	1600	−49.07	95	90
600	−1000	1600	−49.07	85	90
−200	−1000	1600	−49.07	90	90
−400	−1000	1600	−49.07	95	90
−600	−1000	1600	−49.07	85	90
0	−1100	1600	−49.07	90	90
0	−1200	1600	−49.07	95	90
0	−1300	1600	−49.07	85	90
0	−920	1600	−49.07	90	90
0	−840	1600	−49.07	95	90
0	−780	1600	−49.07	85	90
0	−1000	1800	−49.07	90	90
0	−1000	1800	−49.07	95	90
0	−1000	1800	−49.07	85	90
0	−1000	1400	−49.07	90	90
0	−1000	1400	−49.07	95	90

The graph in Figure 10 shows the improved correction over the period, which was achieved due to the prediction and learning capability in neural network-based MPC (model predictive control). The system was able to achieve an accuracy of 0.02 mm. Figure 10 shows the output of the NRTEC system at the end of the calibration process.

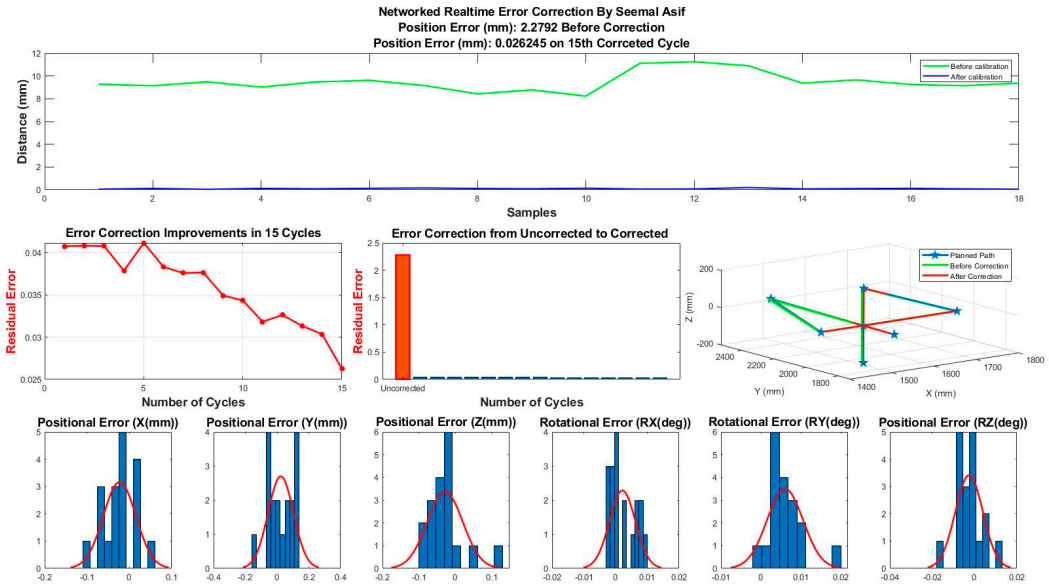


Figure 10. Output of the NRTEC system.

The residual error was 2.2792 before correction, which was reduced to 0.0262; the graph in middle shows that information.

The graph at the top of Figure 10 shows the positional distance between the corrected and uncorrected points. It shows the distance between all 18 points before and after correction. The corrected version of points has been taken after the completion of the 15th cycle. Each cycle contained the path that was formed by the 18 points.

The graph in the middle left shows the error correction improvement from the first cycle to the 15th cycle. The achievement of correction after completing the first cycle of the path was 0.04 and the error was reduced to 0.0262 after the 15th cycle.

The graph in the middle shows the comparison between the errors before starting the process and after finishing each cycle. The error before starting the process was 2.2795, which reduced to 0.0262 after finishing the process.

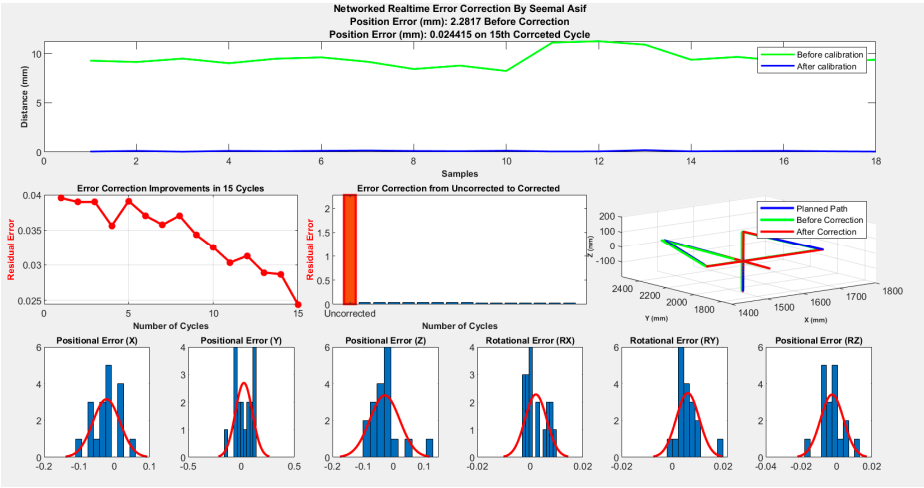
The graph in the middle right shows the path that was planned for the process. The blue line shows the planned path, the green line shows the path that was taken before correction and the red line shows the path that was taken by the robot after the correction and compensation.

The graphs on the bottom show the histogram of the errors in each axis (X, Y, Z) and the rotation (RX, RY, RZ). The histograms are presented for all the 18 points at the end of the process, which means they show the data from the last cycle. It means that the calibration model developed was able to achieve the accuracy of individual components X, Y, Z and RX, RY, RZ within the range of ± 0.15 mm and $\pm 0.02°$, respectively.

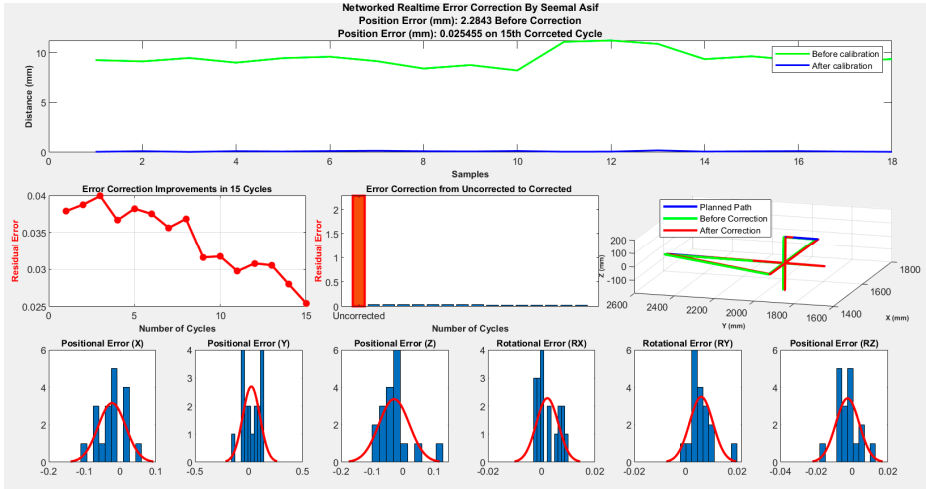
The experiment was performed on different data sets that were collected in different robot positions. Each data set has several parameters; for example, it should have variable positions in +ve and $-$ve direction of each axis and should have variable reachable rotations RX, RY, RZ.

Figure 11 illustrates the performance of the system on eight different robot paths. The 15 cycles of error correction were performed for each robot path to achieve the correction. Graph A in the set of figures below shows the achievement of correction was about 0.0244, graph B shows the achievement of correction was about 0.254, and similarly for other datasets shown in graphs from C to H, the achievement of correction was in the range 0.022 to 0.025, which verifies the claim of the project for the achievement of 0.02 mm correction.

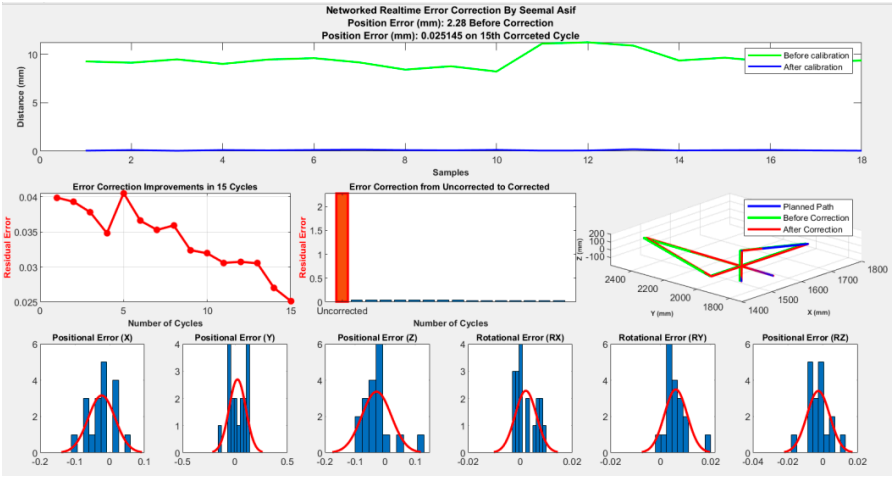
A



B



C



D

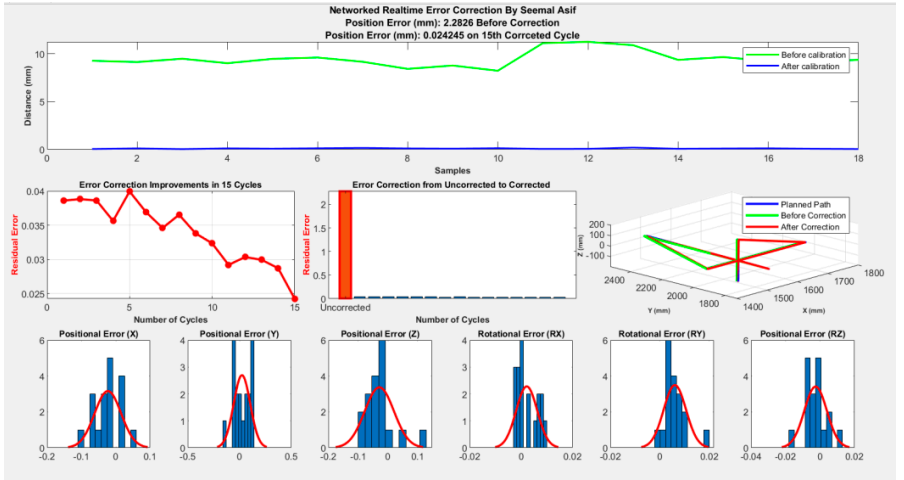
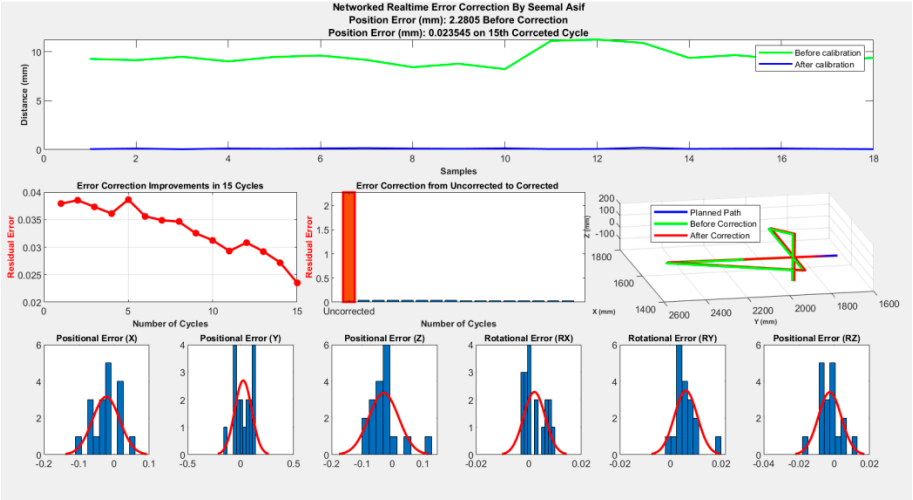
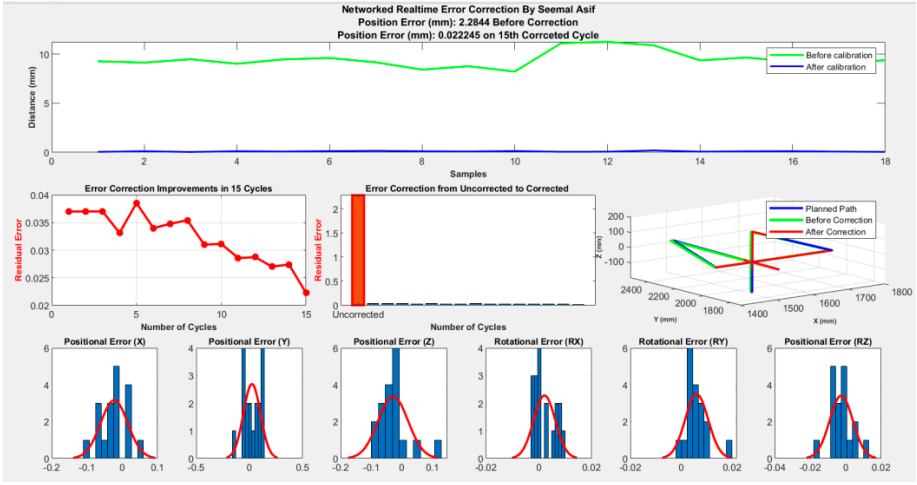


Figure 11. *Cont.*

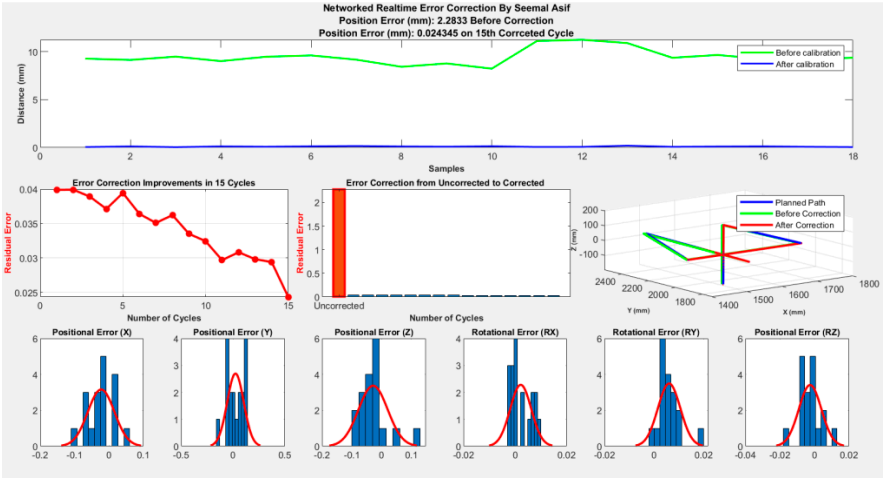
E



F



G



H

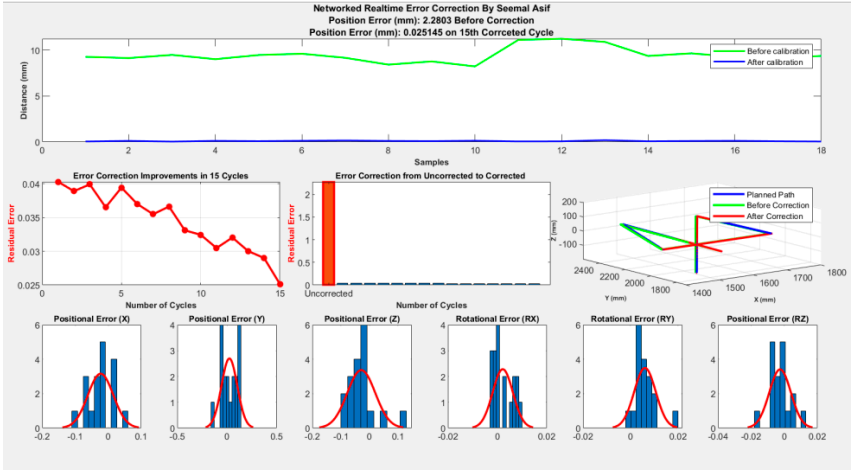


Figure 11. Achievement of correction with variable data sets (**A**–**H**).

The number of cycles to run the system before reaching the correction threshold of 0.02 mm is 15. After 15 cycles, there is no improvement in the error correction. Each cycle takes 225 min to complete and it takes around 6 weeks to finish the experiments and collate the data. The little or no improvement after certain cycles shows that the system has learning capability. The accuracy improves as long as the path of the robot is repeated. This is achieved through the implementation of neural network-based MPC. The use of neural network-based MPC helped the system to learn the error on each point in the path during these cycles. After 15 cycles, the Leica Tracker can be unplugged and physically removed to be used by the other systems where it is required.

The system was tested and verified to perform error correction and compensation in realtime through a network. The flexibility of performing the operation through the network allows the tracker to be used by the other cells, if needed.

Table 2 shows the robot positions before and after calibration. The efficiency of the approach can be seen from Tables 2 and 3. The measurements are in the tracker frame. Figure 12 shows the error result based on the 15 poses in the fourth cycle. One of that pose is noted in Table 2.

Table 2. Positions (in tracker reference frame) before and after calibration.

Nominal Pose					
1507.033	1929.328	3.556133	90.29549	0.220567	−49.5132
Measured Pose					
1507.429	1920.198	5.132199	90.17301	0.067902	−50.026
Corrected Pose					
1507.363	1920.271	5.13706	90.28236	0.068421	−50.0529

Table 3. Error between desired and measured pose on fourth correction cycle.

Error Type	Positional Error	Orientation(°)
Before Correction	9.481775251	0.15
After Correction	0.06	0.034050821

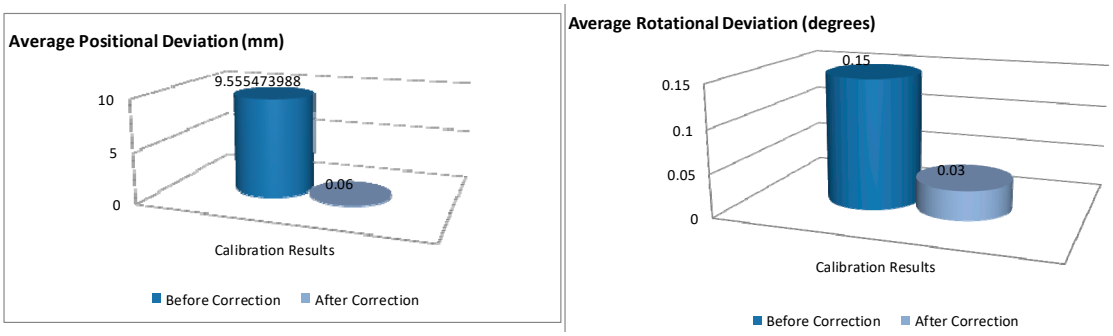


Figure 12. Error correction results (on fourth correction cycle).

7. Conclusions

The system was aimed to perform the robot/machine error correction in a realtime (dynamic) setup through the network. The work was carried out to achieve this aim and developed the NRTEC in a way that it performed the error correction and compensation on an open network. The major achievement of the work is the cost-effectiveness, as the system does not need hard-wire connectivity to the metrological device. The NRTEC can achieve the realtime error correction with an accuracy achievement of 0.02 mm. The system can be used to accurately drill holes on the fixture, which can then be verified by the CMM. Another challenge can be added to have a tool changer as an end-effector to demonstrate the versatility of the system. The tool changer would be changing the T-MAC (as an end-effector). To generalise the use of the robot, the database needs to hold the kinematics specifications of different robots. The current resource database is capable of switching

between resources but cannot hold the kinematic specifications of these resources. Further modifications need to be completed in the database to cater to this need.

Author Contributions: Conceptualization, S.A.; Formal analysis, S.A.; Investigation, S.A.; Methodology, S.A.; Resources, P.W.; Software, S.A.; Supervision, P.W.; Validation, S.A.; Visualization, S.A.; Writing—original draft, S.A.; Writing—review & editing, S.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All data were presented in the main text.

Conflicts of Interest: The author declares no conflict of interest.

References

1. *ISO 9283*; Manipulating Industrial Robots–Performance criteria and Related Test Methods. International Standards Organization: Geneva, Switzerland, 2020.
2. Mooring, B.W.; Roth, Z.S.; Driels, M.R. *Fundamentals of Manipulator Calibration*, 1st ed.; John Wiley & Sons: New York, NY, USA, 1991.
3. Smith, G.T. Machine Tool Metrology. In *An Industrial Handbook*; Springer: Cham, Switzerland, 2016. [CrossRef]
4. Mu, Y.H.; Ngoi, B.K.A. Dynamic Error Compensation of Coordinate Measuring Machines for High-Speed Measurement. *Int. J. Adv. Manuf. Technol.* **1999**, *15*, 810–814. [CrossRef]
5. Ni, J. CNC Machine Accuracy Enhancement Through Real-Time Error Compensation. *J. Manuf. Sci. Eng.* **1997**, *119*, 717–725. [CrossRef]
6. Chen, J.S.; Yuan, J.; Ni, J. Thermal error modelling for real-time error compensation. *Int. J. Adv. Manuf. Technol.* **1996**, *12*, 266–275. [CrossRef]
7. Wang, Z.; Maropolous, P.G. Real-time error compensation of a three-axis machine tool using a laser tracker. *Int. J. Adv. Manuf. Technol.* **2013**, *69*, 919–933. [CrossRef]
8. Lei, W.; Hsu, Y. Accuracy enhancement of five-axis CNC machines through real-time error compensation. *Int. J. Mach. Tools Manuf.* **2003**, *43*, 871–877. [CrossRef]
9. Mooring, B.W. *Fundamentals of Manipulator Calibration*; Wiley: New York, NY, USA, 1992; Volume 32.
10. Gharaaty, S.; Shu, T.; Joubair, A.; Xie, W.F.; Bonev, I.A. Online pose correction of an industrial robot using an optical coordinate measure machine system. *Int. J. Adv. Robot. Syst.* **2018**, *15*, 1729881418787915. [CrossRef]
11. Yi, J.; Du, Z.; Yu, H.; Li, S.; Yang, W. Real-Time Error Compensation Strategy Based on BP Neural Network for Master-Slave Control. In Proceedings of the 2018 IEEE 8th Annual International Conference on CYBER Technology in Automation Control, and Intelligent Systems (CYBER), Tianjin, China, 19–23 July 2018; pp. 267–270. [CrossRef]
12. Yang, X.; Liu, D.; Bai, Y.; Cong, M.; Liao, Z. Kinematics calibration research based on the positioning error of the 6-DOF industrial robot. In Proceedings of the 2015 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), Shenyang, China, 8–12 June 2015; pp. 913–917. [CrossRef]
13. Khaled, T.A.; Akhrif, O.; Bonev, I.A. Dynamic Path Correction of an Industrial Robot Using a Distance Sensor and an ADRC Controller. *IEEE/ASME Trans. Mechatron.* **2020**, *26*, 1646–1656. [CrossRef]
14. Asif, S.; Webb, P. Networked Control System An Overview. *Int. J. Comput. Appl.* **2015**, *115*, 26–30. [CrossRef]
15. Li, A. *Comparison between Model Predictive Control and Pid Control*; University of Pittsburgh: Pittsburgh, PA, USA, 2010.
16. Universal Robot. Remote Control Via TCP/IP-16496. 2019. Available online: https://www.universal-robots.com/articles/ur/interface-communication/remote-control-via-tcpip/ (accessed on 20 September 2022).
17. Universal Robot. Real-Time Data Exchange (RTDE) Guide-22229. 2019. Available online: https://www.universal-robots.com/articles/ur/interface-communication/real-time-data-exchange-rtde-guide/ (accessed on 20 September 2022).
18. Comau. MOTION PROGRAMMING System Software Rel. 3.0. 2006. Available online: https://www.pmdcorp.com/resources/type/articles/get/motion-control-software-setup-the-basics-article (accessed on 29 September 2022).
19. Zhang, K.; Huang, H.; Zhang, J. MPC-Based Control Methodology in Networked Control Systems. In Proceedings of the 6th International Conference, SEAL 2006, Hefei, China, 15–18 October 2006; pp. 814–820. [CrossRef]
20. Webb, P.; Asif, S.; Hogger, S.; Kosche, T.; Kiernan, P. Advanced Flexible Automation Cell Control for Aerospace Manufacturing. *Aircr. Eng. Aerosp. Technol.* **2015**, *87*, 156–164. [CrossRef]
21. Comau Robots. Coamu NM45 Specifications. *Industrial-Robots.* 2015. Available online: http://industrial-robotics.co.uk/comau/nm_spec.htm (accessed on 20 September 2022).

https://dspace.org/

Realtime calibration of an industrial robot

Asif, Seemal; Webb, Phil

2022-09-30
MDPI
https://doi.org/10.3390/asi5050096
Asif S, Webb P. (2022) Realtime calibration of an industrial robot. Applied System Innovation, Volume 5, Issue 5, October 2022, Article number 96

Downloaded from a DSpace repository. https://dspace.org/