High Precision Robotic Manipulator for Bluelining at MAX IV

Vinay Venkanagoud Patil

Lund University
Department of Automatic Control

MSc Thesis
TFRT-6173
ISSN 0280-5316

Department of Automatic Control
Lund University
Box 118
SE-221 00 LUND
Sweden

Printed in Sweden by Tryckeriet i E-huset
Lund 2022

Abstract

MAX IV is a synchrotron facility currently running 14 beamlines hosting about 1000 users per year. It is one of the best of its kind; producing high quality X-rays at the forefront of synchrotron science. To have a beamline or a synchrotron functional, every equipment used needs to be placed in the workspace with high precision. In the current setup, this is supported by bluelining, a technique to transfer the points from the 3D CAD diagram to the real space. This is done by marking points on the floor at places where the equipment needs to be placed. The position measurement is provided by a Leica Laser Tracker. The positioning precision expected in a bluelining task is about 60 micrometers. However, the marking precision expected is about a millimeter due to the thickness of the pen. The manual process of marking these points is a tedious and time-consuming task for the measurement team. Automating this task would ease the work for the measurement team and free up time and resources for them. Not to mention the reduced back pain.

Large-scale infrastructures, such as the MAX IV Laboratory, rely on technology that must conduct extremely accurate operations. The position of the components in space has a significant impact on their performance. The purpose of this Master's thesis is to create a sophisticated robot that will increase the precision of equipment positioning well beyond what is now possible. MAX IV, as well as its colleagues and consumers from academia and business, will profit tremendously from the project's outcomes. In addition, the robot can be employed in other similar locations and has the potential to be used in industrial settings.

Acknowledgements

This Master's thesis was conducted in 2022 at the Department of Automatic Control, Lund University and MAX IV Laboratory.

I would like to thank Anders Robertsson and Gunnar Lindstedt, my academic supervisors, for their support and their faith in my work. I would also like to extend my gratitude towards my teammates from the project courses who helped me gain the knowledge and confidence I needed to bring this thesis to a state of completion. And last but not least, I would like to thank Alina Andersson, research engineer at MAX IV, for her constant support and encouragement through out the thesis.

Sincerely,
Vinay Venkanagoud Patil

Contents

1

Introduction

Equipment installation in a synchrotron dictates the quality and results of experiments conducted. These experiments conducted at MAX IV, are initially designed in a CAD software which gives the measurement team an idea on how the beamline needs to be set up. The team projects these CAD diagrams onto the MAX IV beam lines. MAX IV is equipped with a large network of measured points through out the building that helps the Measurement team keep track of the relative positions of the new equipment with respect to the whole structure. Aforementioned measured points are kept track of by regularly measuring them to update the network.

The goal of the Measurement team is to place the new equipment in the right positions using the network of known points as reference. Since the experiments conducted in a synchrotron are designed around using a very narrow beam of laser, the positional accuracy dictates the quality of results. An error of 60 micrometers is considered tolerable but the team always aims to improve the precision as much as possible.

The measurement is facilitated by a high precision Leica Laser tracker. The Leica Laser tracker comes equipped with a Laser and a set of reflectors that have a 60 degree field of vision. A PC, with a software, Spatial Analyzer[New River Kinematics, 2019] is used to visualize the measurement. Most laser trackers can detect up to 15 micrometers displacement of the reflector from a distance of 1 m.

1.1 Background

The SAM (Survey, Alignment and Mechanical Stability team) is responsible to align and place the equipment at the desired positions in the beamline. The initial step of equipment installation is *Bluelining*. This is when the SAM team projects all the mounting points on the floor. Here they use the reference network that already exists.

Traditionally, this is done by positioning the reflector placed on the floor using a reflector seat [MAXIV Laboratory, 2020]. The position of the reflector; see Fig. 2.2b is fine-tuned while checking the computer screen set up to display the offset from the required position. When the offset is acceptable, the reflector is removed from the reflector seat; see Fig. 2.2a, that allows the user to mark the point on the floor using a marker. This task is tedious, monotonous and time-consuming. Marking as few of 30 points on the floor can easily take hours and involves sitting uncomfortably on the concrete floors.

This current manual process poses several disadvantages that are being addressed in this thesis.

- The process is time-consuming
- Users need to work on the floor in a very uncomfortable manner
- Prone to human error while marking the point on the floor
- Needs assistance from another person other than the one who marks on the floor to keep track of the data stream of measurements

1.2 Problem Formulation and Goals

An important aspect when doing high precision positioning lies in the robot hardware, the control algorithm and the sequence and logic that control the robot. The goal of this thesis is to use the feedback from the Leica laser tracker previously used for bluelining and to control the robot to position its end effector as close as possible to the desired position. This project is an extension of previously completed Master's thesis at the Automatic Control department [Klinghav, 2021]. The Cartesian robot used in the project was designed as part of the Applied Mechatronics and Mechatronics and Industrial Product design courses at LTH [Shahin et al., 2020]. However, the software to control the robot is designed as part of this master's thesis project. In future, a 3-DOF omnidirectional mobile robot will be added to the robot as a base to enable the robot to operate with a larger workspace.

1.3 Limitations

The Laser tracker used in the project was only accessible at MAX IV which was a limiting factor during the project. In order to control the position of the end effector with high accuracy, the robot construction is critical. While testing, it was found that the axes of the robot were not perfectly orthogonal which had its effects on the bluelining performance by some margin. The working range of the robot is limited to a 300 mm square but plans to increase the range of the robot using a mobile robot are in action.

The Leica reflector used in the robot has a 60 degree field of vision, which implies the reflector needs to be facing the tracker in order to receive reliable position data.

1.4 The structure of the report

Chapter 2 covers the theoretical aspects that encapsulate the thesis and the methodologies used are presented in Chapter 3. Results in Chapter 4 are categorised into system repeatability and bluelining performance. Discussions on the results can be found in Chapter 5 and the conclusions can be found in Chapter 6.

2

Theory

2.1 Bluelining

The substantial progress in the field of particle accelerators has made it feasible to make discoveries in comprehending the origins of the universe, elements of matter, medical research, and so on. For effective operation, all particle accelerators, regardless of their scientific use, require exact positioning of numerous components. In particle accelerators, survey and alignment operations involve precisely locating multiple components within micro-metric tolerances across the workspace in relation to a reference point set by beam physicists in any plane.

Before installation of components along a beam-path, the CAD model of the setup is projected on the workspace and positional data of all equipment is generated to meet the requirement of relative alignment along the horizontal plane. This provides the measurement team with the coordinates of the footprints of all the components on the floor. The team then uses a laser tracker to aid the marking of these points on the floor.

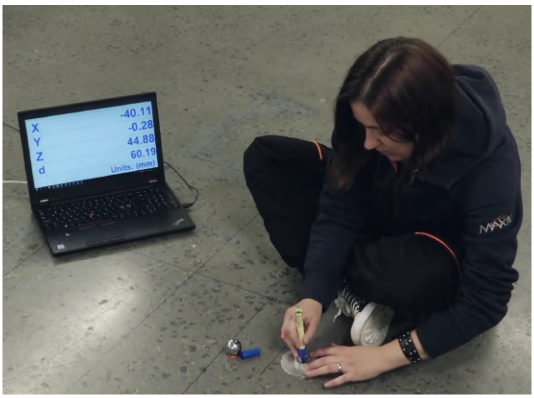
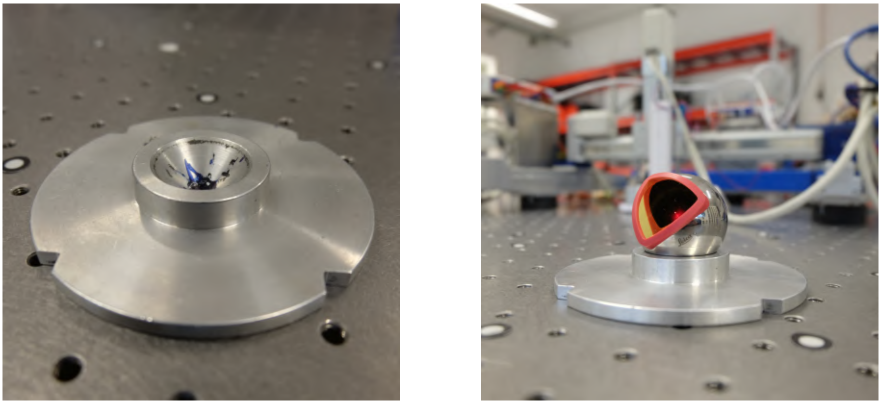


Figure 2.1 Traditional method of Bluelining



(a) (b)

Figure 2.2 (a) Reflector seat, (b) Leica Reflector

2.2 Laser Tracking

Laser tracking methods were previously employed in performance measurement of Industrial Robots at the National Institute of Standards and Technology back in 1986 [Vikas and Sahu, 2021]. Since then there has been an extensive growth in need for high precision measurement in places such as particle accelerators. Positioning of components in particle accelerators, like many other practical applications of precise positioning, requires 3D positioning. These positioning requirements are fulfilled using a portable coordinate measurement device. The Laser tracker is one of many such portable measurement devices which uses a reflector. The Leica Laser tracker; see Fig. 2.3, has a unit that emits a laser beam and uses a tracking mechanism to follow the reflector when it is moved. In order to obtain the 3D Cartesian coordinates, the reflector is moved to the points that need measurement.

The Leica Laser tracker connects via Ethernet / Wi-Fi to a computer that obtains the positional data of the reflector from the Laser tracker and transmits it over UDP using the Spatial Analyzer application. The IP configuration needs to be verified before the operation of the Leica Laser tracker.

2.3 Cartesian Robots

Robots can be classified in many ways based on their mechanical structure [Valin, n.d.] .

- **Linear Robots / Cartesian Robots**
  It is one the most commonly used type of robot. Due to its linear movements along each axis, the robot has a cube-shaped workspace and works best for pick-and-place applications.

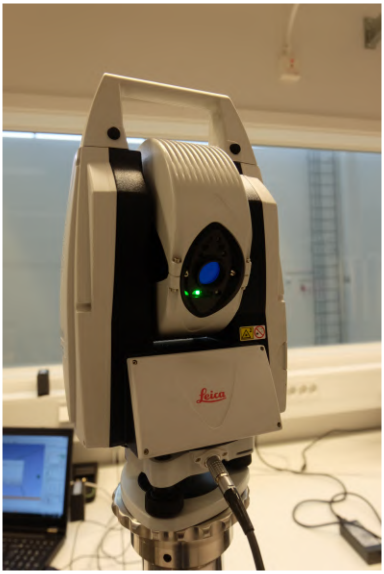


Figure 2.3 Leica Laser tracker

- **Cylindrical Robots**
  Cylindrical Robots are similar to Cartesian robots. They are made of two types of moving elements: rotary and linear actuators. This makes up a Cylindrical Workspace, hence the name cylindrical Robot.

- **SCARA**
  The SCARA robots have a working envelop similar to that of the cylindrical robot. This type of robots are well known for its high-speed performance in pick-and-place applications.

- **Articulated Robot**
  These robots are designed to replicate the motion of a human arm. Typically contain more that 4 rotary joints where every joint enables a degree of freedom. They are very common in the manufacturing industry, where 6 degree of freedom robots are dominating.

- **Delta Robots**
  Delta Robots are one of the fastest actuating robots. Consequently they are also one of the most expensive robots. Delta robots have a dome shaped working envelop. These types of robots are usually employed in high-speed pick-and-place applications.

Due to the simplicity of movements in the workspace, the Cartesian Robot was chosen for this application.

A Cartesian system moves along 3 orthogonal axes: X, Y and Z. The points in this system are represented using 3 values (x, y, z). A Cartesian robot; see Fig. 2.4, as the name suggests uses the Cartesian frame of reference and whose principle axes of control are linear (i.e., they move along a straight line rather than rotate). This mechanical structure, among other things, makes the robot control arm more straightforward.

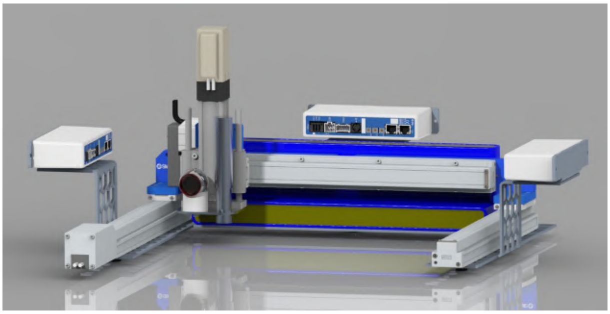


Figure 2.4 Cartesian Robot

Coordinate Transformations

A Linear transformation is a function that takes in an input vector and outputs another vector. The input vector can be a simple coordinate of a point in 3D space. The transformation helps understand how a vector in one frame of reference can be transformed in another frame of reference. Although, one might argue as to why call it a transformation rather than a simple function. In order to simplify this, transformation can be thought of as a movement. It moves an input vector to an output vector. In essence, a transformation moves every input vector to the corresponding output.

A coordinate transformation is used to denote position or movement in a certain coordinate frame. It is a linear relation between the rigid body frame and the world frame which can depict, rotation, translation and a combination of both.

A homogeneous transformation can be described using the following matrix

$$T_{global\ to\ resulting\ frame} = \begin{bmatrix} R_{global\ to\ resulting\ frame} & P_{displacement} \\ 0 & 1 \end{bmatrix} \tag{2.1}$$

where $R_{global\ to\ resulting\ frame}$ represents the net rotation of the resulting frame with respect to the global frame. $P_{displacement}$ is the Cartesian displacement of the frame's origin from the global origin.

In Fig. 2.5, we can see that the robot frame (X',Y',Z') has rotated and translated with respect to the global frame (X,Y,Z). The coordinate transformation will help transforming points in global frame to robot frame.

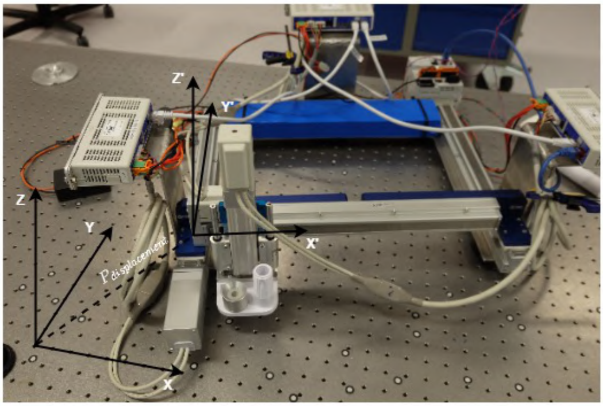


Figure 2.5 Global frame(X,Y,Z) and Robot frame(X',Y',Z')

2.4 System Design

The Robot system used in the project can be realised in the following block diagram.

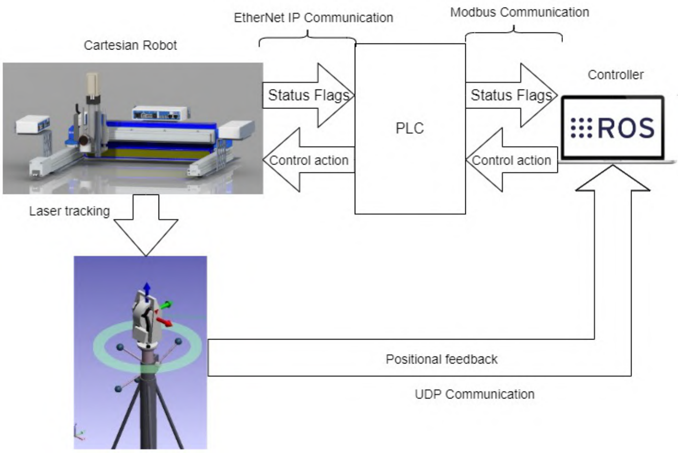


Figure 2.6 System Design

2.5 Robot Operating System

Robot Operating System [Open Robotics, 2021] or ROS is a free and open source robotic middle-ware that defines the components, interfaces and tools for building advanced robots. It provides services designed for a heterogeneous cluster such as hardware abstraction, low-level device control, implementation of common functionality, message passing between processes and package management. ROS helps users develop and connect the actuators, sensors and the control system with ROS tools such as topics and messages. These messages can be recorded using ROS Bags and logs for easy testing, training and quality assurance. Messages can be sent to various tele-operation tools such as Gazebo where the users can work on simulated robots with ease. ROS's modular architecture allows users to build interfaces for just any component with a software interface.

ROS Nodes

A ROS Node [Open Robotics, 2018] is a *process* that performs computations. Nodes are combined together into a graph and communicate with one another using streaming topics, and services. These nodes are meant to operate at a fine-grained scale; a robot control system will usually comprise many nodes. For example, one node controls a laser range-finder, one node controls the robot's wheel motors, one node performs localization, one node performs path planning, one node provides a graphical view of the system, and so on.

The use of ROS Nodes while programming a robot has several benefits. There is additional *fault tolerance* as crashes are isolated to individual nodes. Code complexity is reduced in comparison to monolithic systems. Implementation details are also well hidden as the nodes expose a minimal API to the rest of the graph and alternative implementations, even in other programming languages, can easily be substituted.

Topics Topics are named buses over which ROS Nodes exchange messages [Open Robotics, 2019b]. In general, nodes are not aware of whom they are communicating with. The Nodes are interested in the data from a certain topic they subscribe to. And similarly, the nodes that generate data are interested in topics they publish to. There can be multiple publishers and subscribers to a topic.

Each type of each Topic is dictated by the type of message it is used to transport. A publisher can only publish a matching type of message to a topic and a subscriber can only receive a message with a matching type. Additionally, ROS topics come with semantics that allow the users to keep a track of all the topics in the system and also monitor the messages through the topics. ROS Nodes can be categorised into:

Publishers Publishers are nodes that are used to publish ROS messages to the corresponding topics.

Subscribers Subscribers are nodes that are used to subscribe to ROS topics. Subscribers obtain the data from the topic they subscribe to when a publisher publishes a new message to the topic.

ROS Services

Although the publish/subscribe model is a fairly flexible communication architecture, its many-to-many one-way transport is not ideal for Remote Procedure Call request/response interactions, which are common in distributed systems [Open Robotics, 2019a]. Request/Response is done via a ROS Service, which consists of 2 messages: one for the request and one for the response. The ROS Node that offers the service is registered using a string and the client ROS node that calls for the service sends a request message of the request message type and awaits a response of the reply message type.

2.6 PLC

PLCs are micro-processor based controllers which can be found in most industrial setups. They are constructed in a rugged fashion and are more durable when compared with micro-controllers. PLCs are equipped with programmable memory which stores the instructions and the functions, digital I/O and multiple communication interfaces. Among many advantages, a PLC can be programmed using multiple methods such as Ladder Logic, Structured Text, Function Block Diagram and so on. The user has all the freedom to choose the method of programming and at times a project can support all the different types of programs.

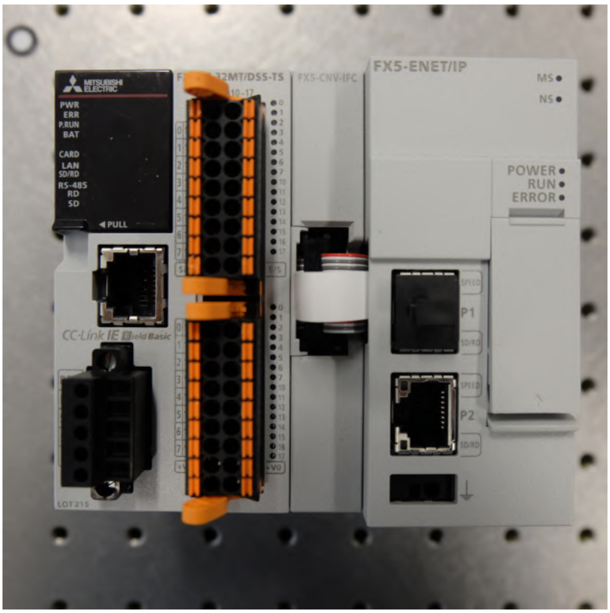


Figure 2.7 Mitsubishi PLC FX5UC-32MT DSS/TS used to control the robot axes

The FX5UC PLC [Mitsubishi Electric, 2017] which was chosen to control the robot is supplied by Mitsubishi Electric. This PLC belongs to the iQ-F series. GX Works3 [Mitsubishi Electric, 2020] was the development environment used to program and configure the PLC for the use case. In order to actuate and control the robot actuators, the robot needs to communicate with all the components consistently.

PLC Communication

This section will discuss the different communication protocols used in this project.

ModBus ModBus communication protocol was first introduced by Modicon (now Schneider Electric) in 1979 [Schneider Electric, 2013] in order to establish communication between PLCs and Industrial Electronic equipment. It uses a master-slave configuration where, in our scope, the PLC will be the master, and the Computer connected will be the slave. More importantly, the ROS Nodes that connect to the PLC via ModBus are the individual slave connections to the PLC.

The FX5UC PLC allows up to 8 ModBus slaves to connect which means the Computer application can use up to 8 ROS Nodes [Section 2.5] that are directly connected to the PLC. The application running on the PLC exposes all the control variables to the ModBus defined registers and coils. The ROS Nodes that are connected to the PLC can then read and write to these registers and coils to control and monitor the robot.

Ethernet/IP Ethernet/IP [Wikipedia, 2013] where IP stands for Industry Protocol is an industrial network protocol. It is one of the fastest growing communication protocols used in industrial automation.

The Servo drivers used in the robot are interfaced with the PLC using the Ethernet/IP protocol.

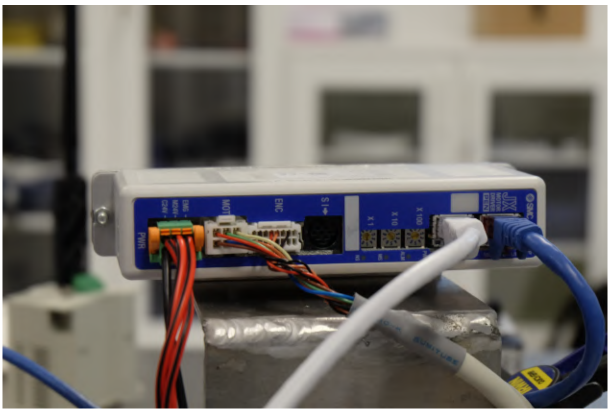


Figure 2.8 Servo driver connected over Ethernet/IP

3

Methodology

Here we will discuss the various parts of the implementation.

3.1 Understanding the task

In this section we close the loop between the multiple components of the robot. The robotic system can be depicted in Fig. 3.1.

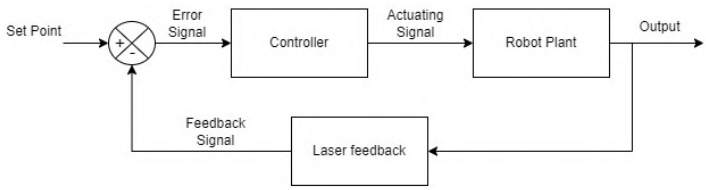


Figure 3.1 Control loop

The system consists of 3 subsystems i.e., a Computer that generates the control signals, a PLC that acts as communication layer between the Computer and the actuators and the sensory feedback that is obtained from the Laser tracker. The software implementation for these subsystems can be broken down as follows:

- Communication between Laser tracker and Computer.
- Communication between Computer and PLC.
- Communication between PLC and Servo Driver.
- Initialization and Calibration.
- Positioning task.
- Marking the floor.

3.2 Communication between Laser tracker and Computer

As previously discussed in Section 2.2 the positional data of the Leica reflector is transmitted using UDP packets. Two types of positional information are extracted as part of the feedback loop which are:

- Global Position of the reflector with respect to the Leica Laser tracker
- Distance of the reflector from the Target

The computer receives these sensory data packets over 2 different UDP ports. 2 ROS services are assigned to acquire these UDP packets and respond with corresponding information upon request.

Global Position of tracker

Upon initialization of the Laser tracker, a coordinate frame needs to be defined. The tracker will then locate the reflector; see (Fig. 2.2b) with respect to this coordinate frame. The global position of the reflector is used to perform the calibration sequence on the robot and also allows the robot to obtain the homogeneous transformation between the Robot frame and the Laser tracker frame.

Distance of tracker from the Target

We know from Section 2.2 that the laser tracker has the ability to track the position of a moving reflector. The laser tracker also has the ability to transmit distance of the reflector from a given target coordinate in the working frame. This data-stream is used to obtain the offset while performing the fine positional corrections.

Measurement sampling

The Laser tracker is one of the more accurate equipment to localize a stationary point. It can detect relative distances of about 15 micrometers from a distance of 1 meter. However, it is not one of the best when it comes to tracking a moving target. This is due to the fact that the laser needs to be pointing at the center of the reflector (Fig. 2.2b). The laser tracker makes small adjustments in order to focus the light at the center of the reflector. While performing such small adjustments, the real-time position estimation varies a lot and renders the measurement data from the laser tracker unreliable.

To tackle this, it was important to sample data when the tracker was absolutely stable. So the measurement was sampled only at times when the robot was stationary.

3.3 Communication between Computer and PLC

As previously discussed in Section 2.6, the Computer communicates with the PLC over ModBus. The PLC being used in the robot can support up to 8 Modbus clients. This implies that it is possible to use 8 ROS Nodes on the computer that can connect to the PLC and read and write registers and outputs.

The ModBus connection between the PLC and the Computer is established using the Python PyModBus Library [PyModBus, 2017]. It is built with the semantics to initialize and connect a ModBus client to the target PLC. It is also possible to read and write to the register and coils of the PLC using the built-in read and write semantics.

Fig. 3.2 shows the different ROS Nodes (Section 2.5) used in the robot application and the direction of flow of ROS messages Ṫhe different topics [Open Robotics, 2019b] are depicted in the rectangular boxes.

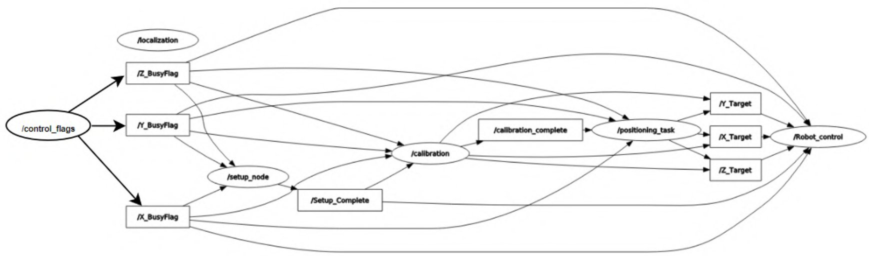


Figure 3.2 A graph that displays all the existing ROS Nodes in the system and how they communicate with each other

3.4 Communication between PLC and Servo Driver

The Servo driver (JXC-91) [SMC AB, 2022a] used in the Robot is supplied by SMC AB. The PLC communicates with the Servo driver over Ethernet/IP. The PLC and the Servo drivers are interfaced by memory mapping the variables on the Servo driver to the PLC. The PLC is then programmed using the application GX-Works3 that uses these memory mapped variables to initialize a communication with the drivers. This part of the application development was supported by Mitsubishi Electric, Lund. They provided the Function block that performed the memory mapping and initialized the communication between PLC and one driver. Further development of the application to connect to the other 2 drivers was done as part of this project. Several Control variables such as the servo enable, target position and so on were later exposed to the Computer that was connected over ModBus in order to control the servos using the computer application.

3.5 Initialization and Calibration

Initialization

As seen in Fig. 3.3, this step involves the powering on of the system and initialization of the Encoders on each Servo. As discussed in Section 3.3 several control variables are exposed to the ROS network. Some of the Control flags utilized are *connected*, *setup_flag*, *SVRE (Servo Enable)*, and *Busy flags* for each axis.

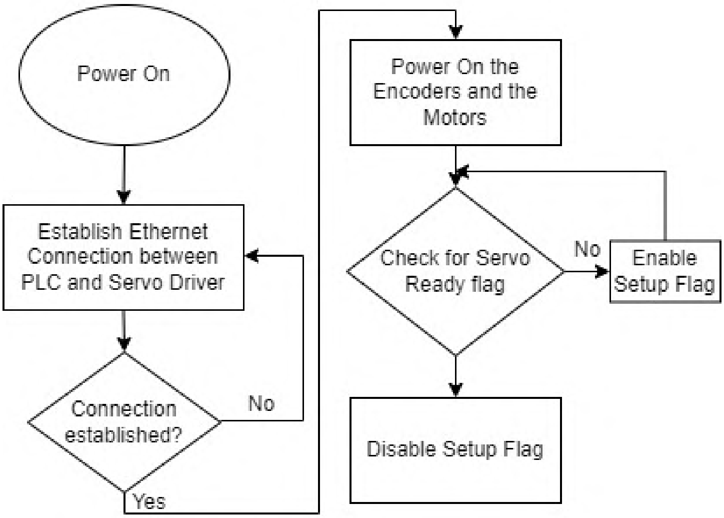


Figure 3.3 Initialization step

Calibration

This step involves the calibration sequence. The robot needs to know its position and orientation in the space in order to perform the positioning of the end effector. This is done by performing a sequence of motions along the 3 orthogonal axes of the robot. The flow chart in Fig. 3.4 explains the sequence.

Upon Calibration, the transformation matrix can be obtained. This transformation matrix is essential to transform the coordinates of the target point which is defined in the Laser Tracker's frame of reference to the robot frame of reference. Using this transformation matrix, the robot will be able to position the end effector to the desired position.

Computation of the transformation matrix

We know,

$$T_{global\ to\ robot}\, P_{global} = P_{robot} \tag{3.1}$$

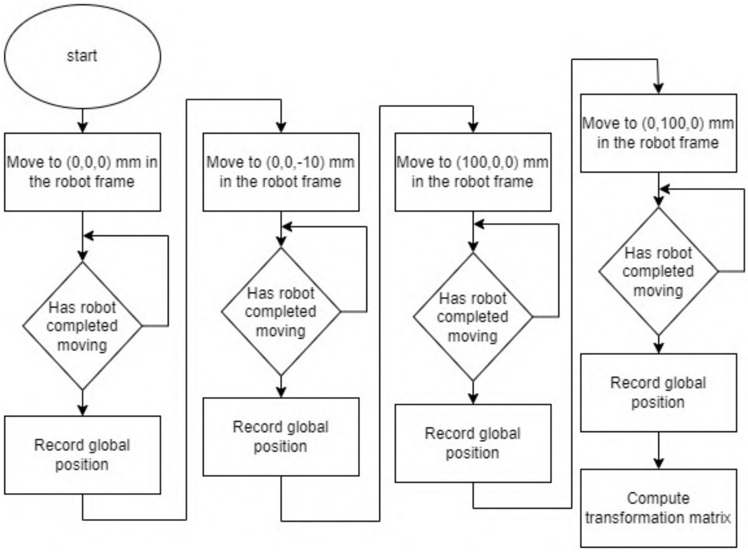


Figure 3.4 Calibration step

where $T_{global\ to\ robot}$ is the transformation matrix that needs to be computed.

To start off, we assume an arbitrary transformation matrix $T_{robot\ to\ global}$ and compute the elements of the matrix using the coordinates of the points acquired during the calibration step calibration.

$$T_{robot\ to\ global} = \begin{pmatrix} a & b & c & X_0 \\ d & e & f & Y_0 \\ g & h & i & Z_0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{3.2}$$

In the arbitrary transformation matrix a, b, c, d, e, f, g, h and i are the elements of the rotation matrix and X_0, Y_0 and Z_0 are the translation part. Also, it is important to note that a, b, c, d, e, f, g, h and i are **not** independent for them to form a rotation matrix.

As mentioned in Fig. 3.4 we record global coordinates of 4 points which are:

- Global coordinates at Robot Origin
- Global coordinates at (0, 0, -10)mm in Robot Frame
- Global coordinates at (100, 0, 0)mm in Robot Frame
- Global coordinates at (0, 100, 0)mm in Robot Frame

$$T_{robot\ to\ global}\, P_{robot} = P_{global}$$

$$\begin{pmatrix} a & b & c & X_0 \\ d & e & f & Y_0 \\ g & h & i & Z_0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X_{robot\ frame} \\ Y_{robot\ frame} \\ Z_{robot\ frame} \\ 1 \end{pmatrix} = \begin{pmatrix} X_{global} \\ Y_{global} \\ Z_{global} \\ 1 \end{pmatrix} \tag{3.3}$$

Eq. (3.3) is a linear system of equations. The values of the elements of the matrix can be computed by substituting the robot frame coordinates in P_{robot} and solving for the unknowns.

This gives us the transformation matrix $T_{robot\ to\ global}$. In order to obtain $T_{global\ to\ robot}$, we need to invert $T_{robot\ to\ global}$.

$$T_{global\ to\ robot} = T^{-1}_{robot\ to\ global}$$

From Eq. (2.1), we have

$$T_{robot\ to\ global} = \begin{bmatrix} R_{robot\ to\ global} & P_{displacement} \\ 0 & 1 \end{bmatrix}$$

Here, $R_{robot\ to\ global}$ is a 3x3 matrix and $P_{displacement}$ is a 3x1 matrix

$$T_{global\ to\ robot} = \begin{bmatrix} R^T_{robot\ to\ global} & -(R^T_{robot\ to\ global}\ P_{displacement}) \\ 0 & 1 \end{bmatrix} \tag{3.4}$$

This transformation matrix is then used to compute the coordinates of the target point in Robot frame.

3.6 Positioning task

Now that the transformation matrix has been computed, positioning of the end effector can be performed. As mentioned in Section 1.3, the mounting of the robot axes were found to be not perfectly orthogonal. This poses a problem when it comes to perform correction as the positional error grows as the end effector moves further from the robot origin.

When the robot application is started, it is provided with a list of all the points that need to be marked in the workspace. Hence, the robot iterates over all the points in the target list, positions the end effector, and places marks on those points. The positioning is done over 2 steps: one initial correction on the $Z = 0$ plane. And a smaller correction close to the floor. The correction in the second step is done iteratively until the error along X and Y axes in the global frame is within 60 micrometers.

This positioning task is depicted in the flowchart of Fig. 3.5.

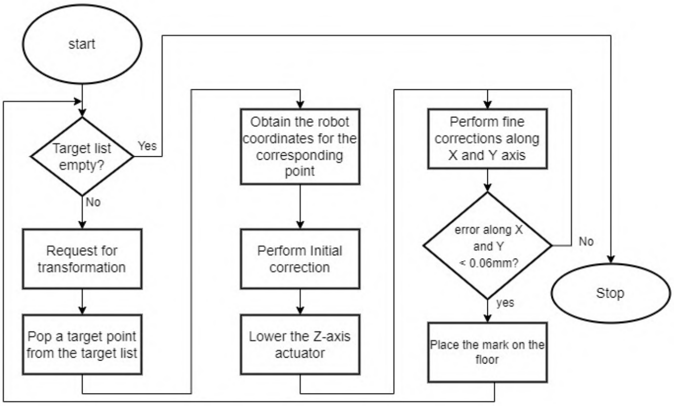


Figure 3.5 Positioning Task

3.7 Marking the floor

Upon completion of positioning of the end effector the next step is to place the marker on the floor. It is important to note that the tip of the pen is placed with an offset along the X, Y and Z axe s from the reflector in the robot frame of reference.

The first step is to move the robot's end effector along X and Y corresponding to the pen-reflector offset. The offset correction will bring the pen right above the target on the floor. Following that the pen is slowly lowered until the pen has placed the mark. The pen is equipped with a limit switch which stops the Z-actuator when the pen is lowered enough to place the mark on the floor; see Fig. 3.6.

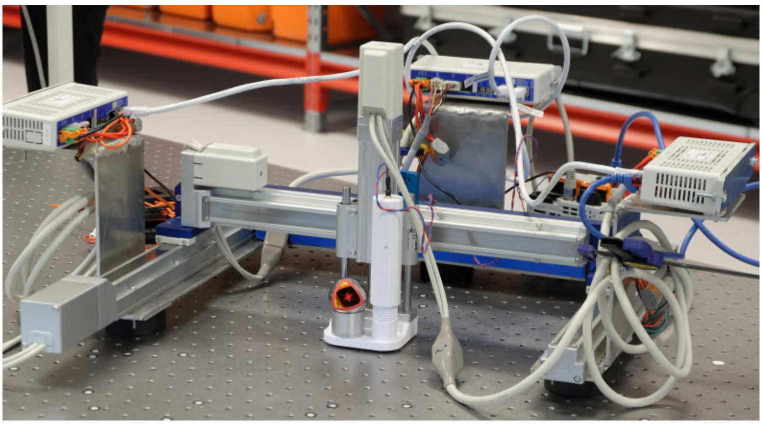


Figure 3.6 Robot placing mark on the floor

4

Results

This chapter summarizes all the results obtained throughout the project.

4.1 System

In order to perform bluelining, a Cartesian robot was designed as part of the Applied Mechatronics and Mechatronics and Industrial Product Design courses at LTH [Shahin et al., 2020]. The robot was controlled using a combination of PLC and a Computer running ROS. A major part of the thesis was involved in developing the different components of software that operates the robot. The different components are the numerous ROS nodes that keep a track of all control flags in the robot and actuate them, the PLC software that receives the command signals from the Computer and reacts to it and the positional feedback to the entire system.

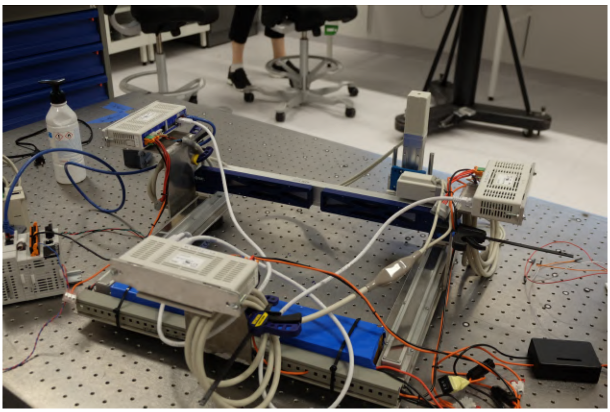


Figure 4.1 Bluelining robot

The Bluelining Robot

The Bluelining robot used in this project, see Fig. 4.1, is designed to mark points from a given list of targets on the floor. The main parts of the robot are the PLC, the 3 SMC drivers and actuators and the computer that communicates with all the components of the robot.

Robot performance

This section discusses the robot performance and its effects on the Bluelining performance.

Actuator Repeatability The actuators used in the robot are listed in Table 4.1.

Table 4.1

Axis	Actuator used
X axis	LEFS16LB-300-R1C6171
Y axis	LEFS16B-300-R1C6171(master) and LEFG16-S-300(slave)
Z axis	LEYG16LDB

The actuators used in the robot have a documented repeatability of ±0.02 mm [SMC AB, 2022b]. The results for this will be discussed in Section 4.2 of the results.

Testing Robot Orthogonality In order to control the robot and use the coordinate transformation matrix we obtained in Section 3.5 we need the robot axes of motion to be orthogonal. To verify this, the robot actuator was moved with the reflector and the corresponding position was recorded at several points along the robot axes on the Leica Spatial Analyzer software.

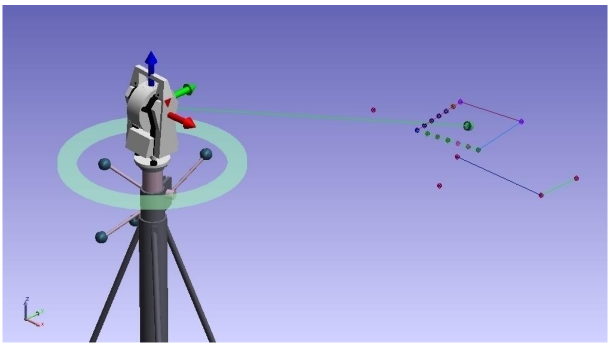


Figure 4.2 Points recorded in the SA Kinematics software. It also shows the Leica Laser tracker's frame of reference

The Spatial Analyzer software has the functionality to fit a line through all the recorded point. With this, we can find the angle between all the different axes along which the points were recorded. Following results were obtained.

Table 4.2

Lines	Angle (degrees)
X axis and Y axis	89.9252
Y axis and Z axis	90.0004
X axis and Z axis	89.9870
Y master and Y slave	0.0466

We can see from Fig. 4.1 that the Y Axis of the robot has 2 rails. One of them is a master rail that drives the actuator and the other one follows the master. It is also important to have these lines parallel in order to achieve the best results for Bluelining.

4.2 Bluelining

This section discusses the bluelining performance of the robot. For the bluelining task, 3 points were chosen at random in the workspace. The robot was placed with random orientation while making sure the reflector is facing towards the Laser tracker.

The points chosen for the Bluelining verification were:

Table 4.3

Point	Coordinates in Laser tracker's frame of reference (mm)
P1	(-704.84, 2474.93, -922.78)
P2	(-576.62, 2501.99, -922.63)
P3	(-624.92, 2438.59, -922.68)

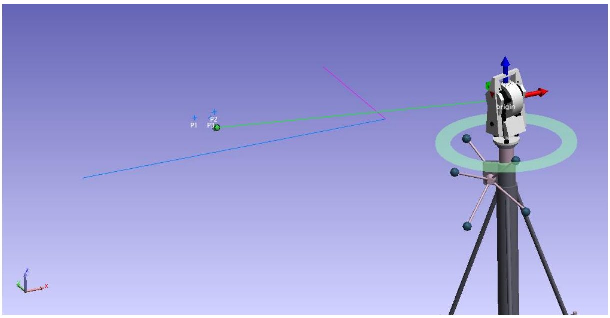


Figure 4.3 Chosen points for bluelining in the workspace

The coordinates of the chosen points in Table 4.3 are given with respect to the Leica Laser frame of reference, where the origin lies in the center of the laser as shown in Fig. 4.3.

Accuracy and Precision Observational error can be quantified using accuracy and precision. Accuracy is the measure of how far off is a given measurement from the actual value. On the other hand, precision quantifies the dispersion of the measurements. The following picture explains the accuracy and precision.

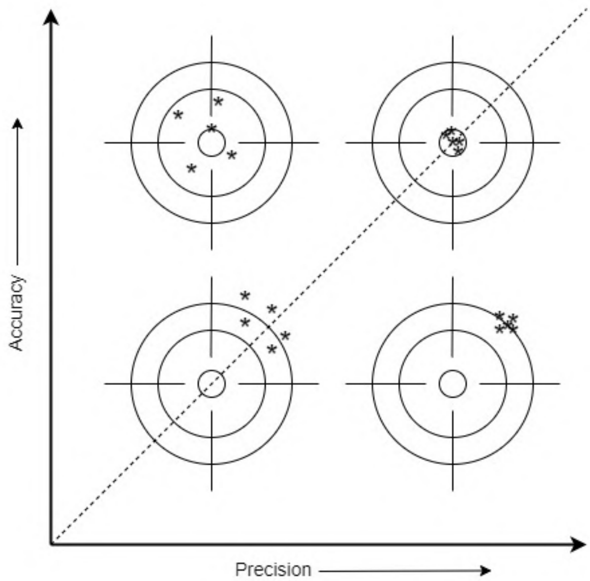


Figure 4.4 Precision vs Accuracy

The goal of the Bluelining task is to achieve as high accuracy as possible. In order to verify the results the tool was used to manually place marks on the workspace.

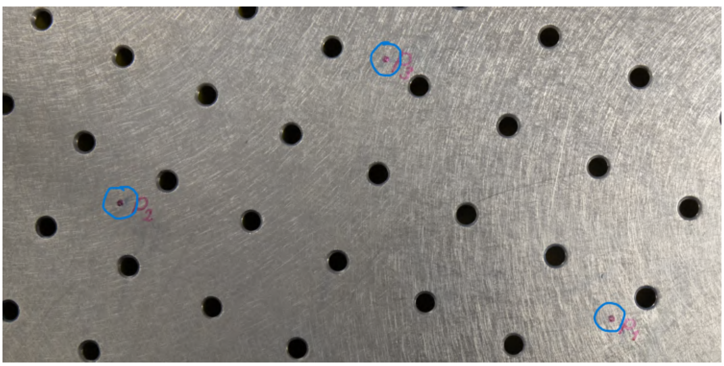


Figure 4.5 Target points marked using the tool in Fig. 2.2a

The robot was then tasked to place a mark on these points seen in Fig. 4.5 for 5 times each.

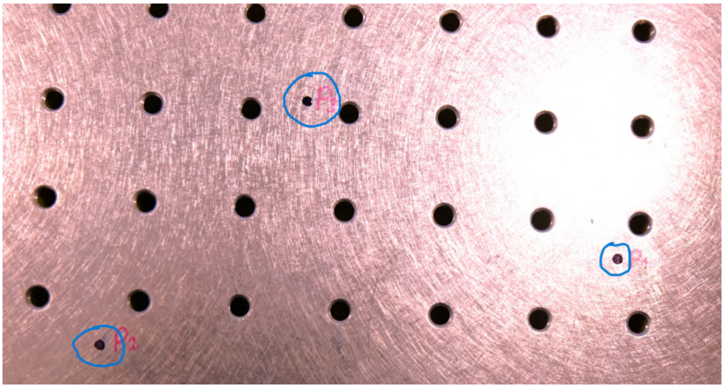


Figure 4.6 Target points marked using the robot **??**

From testing Bluelining When taking a close look at Fig. 4.6, we can see that the robot was able to place the markers with very high accuracy at the point. Five rounds of tests were conducted to mark the points in Table 4.3 in order to verify repeatability. The same tests were conducted with different robot orientation in the workspace to ensure proper functioning of the coordinate transformation calculations.

5

Discussion

This chapter summarizes all the results obtained through out the project and gives an overview of the main topics and the objectives. This section also provides insight into future development in this project and possible improvements that could further improve the current solution.

Performance evaluation

The Laser tracker was only accessible at MAX IV which was a limiting factor through out the project. Different sensor options were explored such as the HTC VIVE lighthouse were explored. Although positional precision was promising, it could never live up to the critical precision requirements at facilities like MAX IV. Some parts such as the PLC code were given relatively short amounts of time due to late arrival of the solutions from Mitsubishi Electric. However, the targeted requirements were achieved and the robot was controlled in the intended manner. Looking back at the project, the method to use the linear system of equations instead of manually calculating the angles for the transformation matrix in Section 3.5 proved time-saving and computationally light. In addition, the usage of ROS proved very helpful with monitoring and managing the different applications running on the robot. Due to the modularity of the software development in the ROS environment, it allows for future updates to the design of the robot easy to accommodate.

6

Conclusion

This chapter summarises the results and also talks about possible future development in the project.

Conclusions

The goal of this thesis was to use the feedback from the Leica laser tracker and controlling the robot to position its end effector to the desired position and accurately mark on the floor. To conclude the thesis, it should be stated that the goals of the thesis have been reached to a great extent. Although the range of motion of the robot is quite limited to a 300mm square, the robot's ability to position the end effector with very good accuracy was achieved. The robot can accept a list of task points that need to be marked. The robot is able to find its position and orientation in its work space. The robot's movements are repeatable with high degree of precision.

Future development

As mentioned in Section 1.3 the robot is limited by its range of motion. In order to increase its working range, the robot will be coupled with a mobile robot. This opens up to many more applications other than Bluelining such as management and maintenance of the reference network used at MAX IV. In fact, a mobile robot that extends the range of the current robot is already in development at the Industrial Engineering department and the Automatic Control department, LTH. It will be very interesting to follow their progress.

Bibliography

Klinghav, L. (2021). *Mobile Floor-Marking Robot, utilizing Feedback from Laser Tracker*. Master's Thesis TFRT-6123. Department of Automatic Control, Lund University, Sweden.

MAXIV Laboratory (2020). *"Magnus Malmgren and Alina Andersson - SAM team at Max IV Laboratory"*. URL: https://www.youtube.com/watch?v=JueNflX_VW4.

Mitsubishi Electric (2017). *MELSEC iQ-F FX5U User's Manual (Hardware)*. Mitsubishi Electric. URL: https://eu-assets.contentstack.com/v3/assets/blt5412ff9af9aef77f/bltd518732e1d328970/617235820c7ad207787e4806/a04f3bfc-557f-11e7-a133-b8ac6f83a177_JY997D55301-F.pdf.

Mitsubishi Electric (2020). *GX Works 3*. URL: https://www.mitsubishielectric.com/fa/products/cnt/plceng/smerit/gx_works3/programming.html. (accessed: 10.05.2022).

New River Kinematics (2019). *Spatial analyzer kinematics*. URL: https://www.kinematics.com/index.php. (accessed: 20.05.2022).

Open Robotics (2018). *Ros nodes*. URL: http://wiki.ros.org/Nodes. (accessed: 25.05.2022).

Open Robotics (2019a). *Ros services*. URL: http://wiki.ros.org/Services. (accessed: 25.05.2022).

Open Robotics (2019b). *Ros topics*. URL: http://wiki.ros.org/Topics. (accessed: 25.05.2022).

Open Robotics (2021). *Robot operating system*. URL: https://ros.org/. (accessed: 25.05.2022).

PyModBus (2017). *PyModBus*. URL: https://pymodbus.readthedocs.io/en/latest/. (accessed: 15.06.2022).

Schneider Electric (2013). *Modbus*. URL: https://www.se.com/us/en/faqs/FA168406/#:~:text=Modbus%20is%20a%20serial%20communication,lines%20between%20electronic%20devices.. (accessed: 20.05.2022).

Shahin, A., F. Bille, K. Henriksson, L. Hansson, and M. Malekpour (2020). “Concept development to product production of a Cartesian Robot for high precision Bluelining”. *Mechatronics Course, Dept. of IEA, LTH*. Student Paper.

SMC AB (2022a). *JXC91 Ethernet/IP operation manual*. URL: http://content2.smcetech.com/pdf/manuals/SFOD-OMT0006-A.pdf. (accessed: 01.06.2022).

SMC AB (2022b). *SMC ball and screw actuators*. URL: https://www.smcpneumatics.com/pdfs/LEF_1.pdf. (accessed: 01.06.2022).

Valin (n.d.). *Industrial Robots*. URL: https://www.valin.com/resources/articles/five-types-of-industrial-robots-and-how-to-choose-the-best-fit#:~:text=A%20simpler%2C%20more%20complete%20definition,their%20speed%2C%20size%20and%20workspace.. (accessed: 15.06.2022).

Vikas and R. K. Sahu (2021). “A review on application of laser tracker in precision positioning metrology of particle accelerators”. *Precision Engineering* **71**, pp. 232–249. ISSN: 0141-6359. URL: https://www.sciencedirect.com/science/article/pii/S014163592100101X.

Wikipedia (2013). *EtherNet/IP*. URL: https://en.wikipedia.org/wiki/EtherNet/IP. (accessed: 15.06.2022).

Lund University
Department of Automatic Control
Box 118
SE-221 00 Lund Sweden

Document name
MASTER'S THESIS

Date of issue
June 2022

Document Number
TFRT-6173

Author(s)
Vinay Venkanagoud Patil

Supervisor
Gunnar Lindstedt, the Division of Industrial Electrical Engineering and Automation (IEA), Lund University, Sweden
Alina Andersson, MAX IV Laboratory, Lund University, Sweden
Anders Robertsson, Dept. of Automatic Control, Lund University, Sweden
Tore Hägglund, Dept. of Automatic Control, Lund University, Sweden (examiner)

Title and subtitle
High Precision Robotic Manipulator for Bluelining at MAX IV

Abstract

MAX IV is a synchrotron facility currently running 14 beamlines hosting about 1000 users per year. It is one of the best of its kind; producing high quality Xrays at the forefront of synchrotron science. To have a beamline or a synchrotron functional, every equipment used needs to be placed in the workspace with high precision. In the current setup, this is supported by bluelining, a technique to transfer the points from the 3D CAD diagram to the real space. This is done by marking points on the floor at places where the equipment needs to be placed. The position measurement is provided by a Leica Laser Tracker. The positioning precision expected in a bluelining task is about 60 micrometers. However, the marking precision expected is about a millimeter due to the thickness of the pen. The manual process of marking these points is a tedious and time-consuming task for the measurement team. Automating this task would ease the work for the measurement team and free up time and resources for them. Not to mention the reduced back pain.

Large-scale infrastructures, such as the MAX IV Laboratory, rely on technology that must conduct extremely accurate operations. The position of the components in space has a significant impact on their performance. The purpose of this Master's thesis is to create a sophisticated robot that will increase the precision of equipment positioning well beyond what is now possible.MAXIV, as well as its colleagues and consumers from academia and business, will profit tremendously from the project's outcomes. In addition, the robot can be employed in other similar locations and has the potential to be used in industrial settings.

Keywords

Classification system and/or index terms (if any)

Supplementary bibliographical information

ISSN and key title
0280-5316

ISBN

Language
English

Number of pages
1-35

Recipient's notes

Security classification