

LUT UNIVERSITY  
LUT School of Energy Systems  
LUT Mechanical Engineering

*Narayan Panthi*

**POSITIONING OPTIMIZATION OF MOBILE ARTICULATED ROBOT  
RELATIVE TO TASKSPACE**

Supervisor: Associate professor Huapeng Wu  
Associate professor Francois Christophe  
Dr. Ming Li

Examiner(s): Associate Professor Huapeng Wu  
Dr. Ming Li

## **ABSTRACT**

LUT University  
LUT School of Energy Systems  
LUT Mechanical Engineering

Narayan Panthi

### **Positioning optimization of mobile articulated robot relative to taskspace**

Master's thesis

2021

45 pages, 20 figures and 3 table

Examiners: Professor Huapeng Wu  
Dr. Ming Li

Keywords: positioning, Mobile Manipulator, Optimization, Inverse reachability,

This thesis describes an optimization approach for positioning a mobile-articulated robot in a task-space. This thesis dives into the first approach, of preplanned location with certain dimension constraints. In the second approach, an objective function is created, which can be used in any dimension of circular or rectangular task space. The method was tested by experiments carried out in the lab to control the hybrid robot to fulfill the assembly task of drilling on a Sheetmetal panel used in roofs.

## **ACKNOWLEDGEMENTS**

Firstly, I would like to express my sincere gratitude towards my supervisor, Huapeng Wu, and Dr. Ming Li from (LUT University) for their continuous guidance throughout the thesis. I would like to extend my great appreciation towards D.Sc.(Tech) Francois Christophe (HAMK Tech) for his suggestion and feedback during the entire thesis process.

Finally, I would like to thank my family and friends for their support and encouragement.

*Narayan Panthi*

Narayan Panthi

Lappeenranta 30.03.2021

**TABLE OF CONTENTS**

<b>ABSTRACT</b> .....	<b>1</b>
<b>ACKNOWLEDGEMENTS</b> .....	<b>2</b>
<b>TABLE OF CONTENTS</b> .....	<b>5</b>
<b>LIST OF SYMBOLS AND ABBREVIATIONS</b> .....	<b>6</b>
<b>1 INTRODUCTION</b> .....	<b>7</b>
1.1 Background.....	8
1.2 Motivation.....	15
1.3 Objective.....	15
<b>2 METHOD</b> .....	<b>16</b>
2.1 Applied research .....	16
2.2 Robots .....	19
2.2.1 Design and architecture .....	26
2.2.2 Transformation matrix .....	27
2.3 Optimization .....	29
2.3.1 Differential Evolution (DE) .....	29
2.3.2 Teaching learning-based optimization(TLBO).....	31
2.4 Implementation and experiment .....	35
<b>3 RESULTS AND ANALYSIS</b> .....	<b>39</b>
3.1 First test.....	39
3.2 Second test .....	40
<b>4 CONCLUSION AND FUTURE WORK</b> .....	<b>43</b>
<b>LIST OF REFERENCES</b> .....	<b>45</b>

## LIST OF SYMBOLS AND ABBREVIATIONS

### List of symbols

A, B	assigned area for the sectors [sq m]
$D$	clearance distance required by the robot to approach any object [m]
reach	radius of the articulated manipulator [m]
width	width of the table [m]
$x$	the horizontal distance from the edge of the table [m]
$\alpha, \beta, \theta$	corresponding angles in radians [rad]

### Abbreviations

AGV	Automated guided vehicles
AMR	Automated mobile robots
API	Application Programming Interface
CAGR	Compound Annual Growth Rate
C-space	Configuration space
DE	Differential Optimization
DH	Denavit-Hartenberg
DOF	Degree of freedom
EE	End effector
GA	Generic algorithm
GUI	Graphical user interface
HRC	Human-Robot Collaboration
HTML	Hypertext Markup Language
MIR	Mobile industrial robot
MMR	Mobile manipulator robots
ROS	Robot operating system
SLAM	Simultaneous localization and mapping
SMEs	Small and medium-sized enterprises
TLBO	Teaching Learning Based Optimization
UR	Universal robot
URDF	Universal Robot Description Format
XML	Extensible Markup Language

## 1 INTRODUCTION

Whether it be small and medium-sized enterprises (SMEs) or large enterprises, the manufacturing industries are leaning to adapt to a new era of industrialization. The introduction of different technologically advanced systems and innovative manufacturing methods are eliminating traditional practices. Industry 4.0, the revolutionary concept of intelligent networking of machines and processes for the industry with information and communication technology, has completely changed the game for manufacturing. The introduction of Smart factories with Cyber-physical systems, autonomous machines, advanced robots, connected to the Internet of things, cloud computing, bridging communication are targeting to create 'automatization.' It is expected, over 50 billion machines are going to work connectively worldwide in the next five years. (Wisskirchen *et al.*, 2017, p. 12.)

The growth in robot installation sales, pre-trade tensions 2019 (International Federation of Robotics, 2017, p. 13), and increase in demand for the collaborative robot (Bogue, 2016, p. 10) could indicate how factories are evolving towards the fourth industrial revolution. It is estimated within 2020, more than half of European companies plan to automate at least ten processes via robot process automation (Siderska, 2020 p. 24). The industries are searching for the most effective automated robot to use in a dynamic environment, which can provide the flexibility of handling multipurpose in multiscale manufacturing processes. The third generation's industrial robots have low-level controllers, which are carried out with latency and proved to be less responsible for a real-time environment (Schrimpf, 2013, p. 19). It is not swift in decision-making as the sensors are not entirely reliant. The variations in sensor reading and sensor fusion techniques produced poor decisions that could not be guaranteed. Even with expert level of root programming for robots' functionality always meant several if-then or else conditions for the robot to make decisions. Ultimately, it cannot anticipate all scenarios. Therefore the only viable solution was to use Human-Robot Collaboration (HRC). It is evident that automation has started shifting gears in overall dynamics on the need for HRC use. (El Zaatari *et al.*, 2019, p. 178.)

This thesis aims to present a viable solution using a robot's reachability map to position the hybrid mobile manipulator robot (MIR200 and UR16e) around the task-space. The robot

was assembled at the HAMK Tech Research unit in HAMK University of applied science, Riihimäki, Finland. The positioning configuration uses an external program on a Python 3.8 platform to update a mobile robot's positioning sequence, and the optimization algorithm was used in MATLAB 2020a to generate the values. The project focuses on using two different approaches to making adjustments and guiding the unit. Further objectives for this project are as follows.

- How to improve the navigational space of the MIR200 platform
- How to plan MIR's path movements so that UR16 gets to its comfortable working zone for each step of its mission?
- How creating an objective function of the workspace using the geometrical approach of the table's dimension can be purposed in optimizing the algorithm for positioning the unit?

### 1.1 Background

A combination of joints with rigid body links is a building block of a robot. Following these joints constrain the links of the robot can move about in space. The configuration space (C-space) of a robot is the n-dimensional space containing all possible configurations for the robot. Similarly, a naturally expressive C-space for a particular task can be expressed as task space; on the other hand, the cloud of points achieved with all possible set of configuration of joint angle values to reach a pose with the tool center point while abiding by robot constraints is Robot's workspace (Lynch and Park, 2017, p. 29). Any pose  $p \in SE(3)$  which lies inside the workspace of a robot must give a joint configuration  $q \in \mathbb{R}^n$  to validate the pose  $p$  of the end-effector (EE). The process of determining a joint configuration from a pose  $p$  is known as inverse kinematics. The number of configurations for a particular pose  $p$  could vary depending upon the number of joints; however, as the number of joints increases, it becomes challenging to form a geometrical deduction and obtain joint-values for the pose of the EE. (Vahrenkamp, Asfour and Dillmann, 2013, p. 1974.)

A rigid body (object) in a C-space can be defined with six sets of variables, three of which represent the body's position in space, and the remaining three represent the orientation of the object (Lynch *et al.*, 2017, p. 3). Using this information and inverse kinematics of the robot for a particular pose  $p$ , a point cloud, or the reachability space can present a simplistic model to determine the robot's positioning from the grasping object's pose. In other words,

the robot's reachability map can be inverted to generate an object-centered reachability map for potential robot base when a pose is known. (Vahrenkamp *et al.*, 2013, p. 1974.)

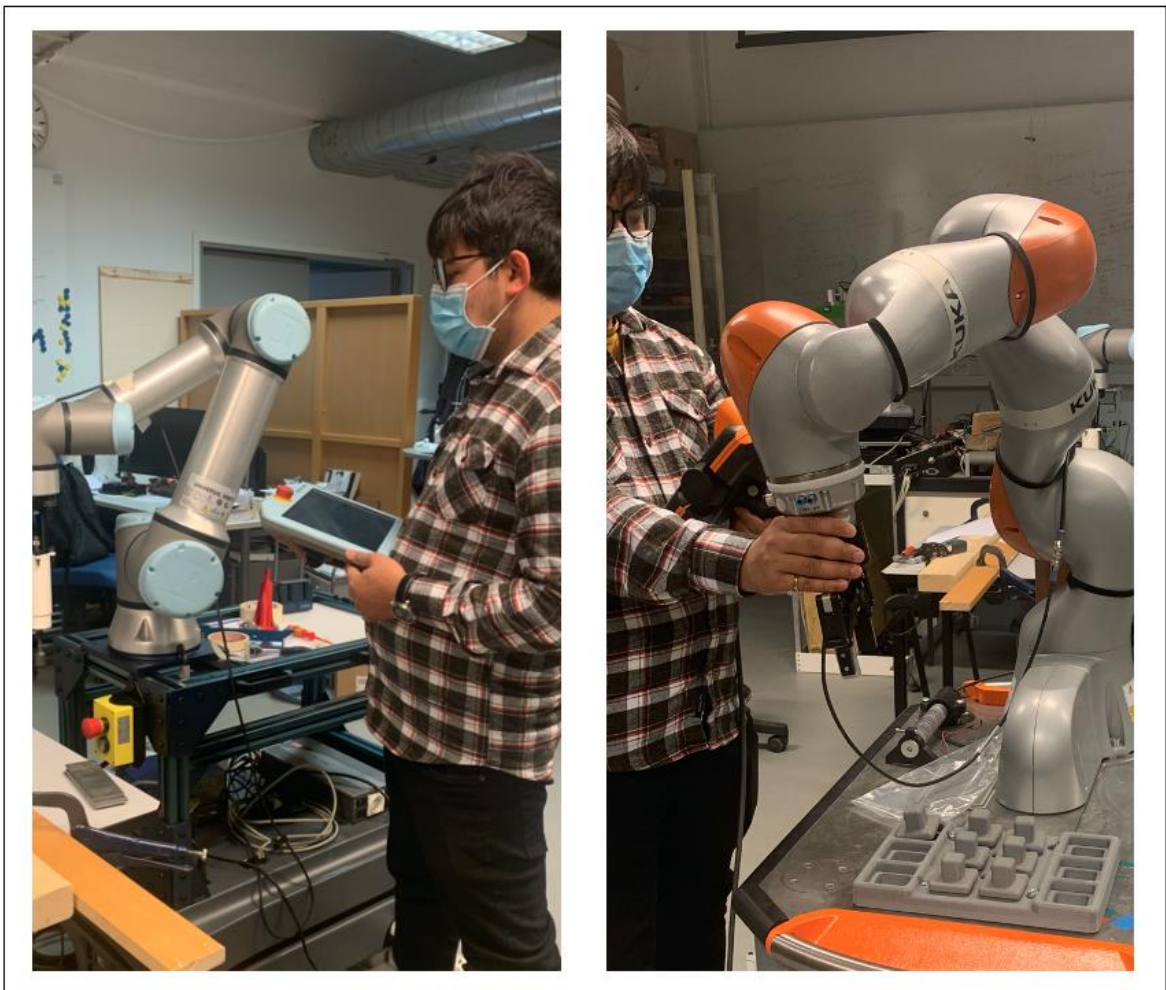
The efficient use of inverse reachability techniques for positioning of robot base was proposed in Zacharias *et al.*, Diankov *et al.*, and Reuleaux, and comprehensive analysis on the computational time was done in Makhal *et al.*, Vahrenkamp *et al.*, presented a generic approach to generate a distribution in  $SE(2)$ . In contrast, Kalawoun *et al.*, extended the coverage by proposing multiple robots to complete the task using a hybrid optimization algorithm. Similarly, Forstenhausler *et al.*, used a torus-based workspace map to find the robot's base's optimal position. It can be observed that all approaches for positioning were based on inverse kinematics, and various models of optimization techniques were implemented to achieve a viable solution.

### Cobots

Collaborative robots are the next addition to Industry 4.0 as industrial manufacturing robots. Between the years 2016 - 2020, there has been a growing interest in research on the collaborative robot, with 969 academic publications in 2020 retrieved from a search analysis with keyword collaborative robot in the Scopus database. Countries like United States, China, Germany are among the top contributors to the research on innovation and development of these robots. (International Federation of Robotics, 2017.) The industry hopes to transform entirely from the traditional safety cage robots to more on shared human and robot co-working environment as envisioned by El Zaatari et al (El Zaatari *et al.*, 2019, pp. 162-164). According to Research and Markets Ltd, (2020), there will be significant growth in the market share of Cobot by a CAGR of 30.37% between the period from 2020 to 2025. The robot market, in essence, will create new value, business models with smart factories (Wisskirchen *et al.*, 2017, p. 12.) Collaborative robots designed with human-robot interaction have already been implemented in numerous manufacturing assembly lines in production environments worldwide.

The biggest robot consumers, auto industries of 28% (International Federation of Robotics, 2017), are implementing cobots from Kuka, UR, ABB in their factories. The advantages over third-generation industrial robots, such as fast setup and easy interface along with affordable price, make cobots a highly lucrative product for competitive SMEs. These SMEs

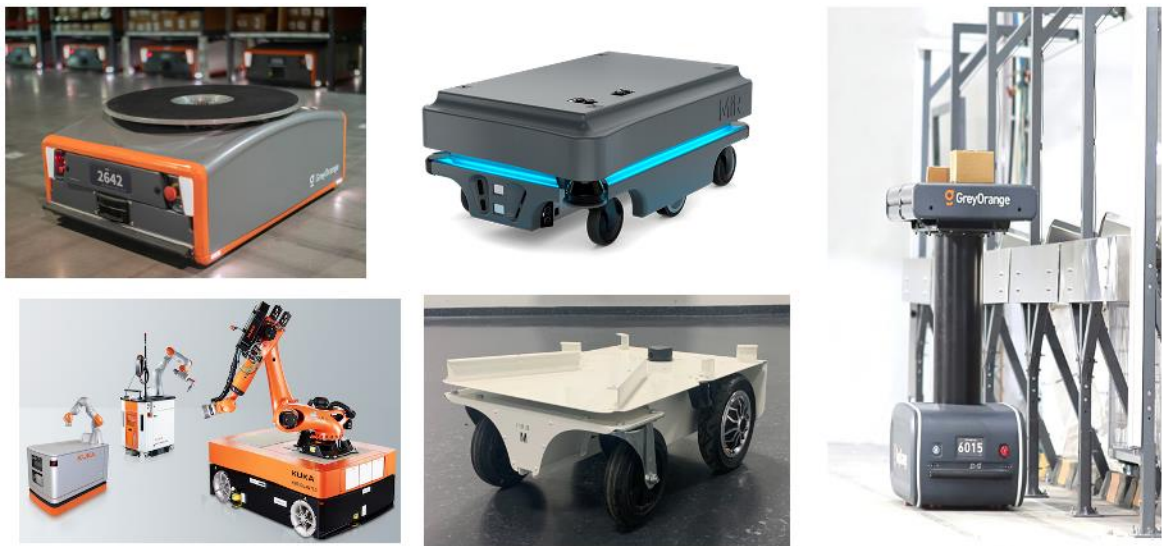
account for about 90% of all the enterprises in various countries (Pech and Vrchota, 2020, pp. 1-2). In an effort to adapt to modernization, small-scale manufacturers are also updating to robots and automating with the digital transformation of processes in manufacturing and production (Siderska, 2020, p. 25). Although the industry is thriving with the introduction of cobots, robot manufacturers are still focused on developing robots with higher safety and swift response to work in a dynamic environment. The current generation of cobots is still operating under strict speed and maintain distance under regulations EN ISO 10218-1 and EN ISO 13855 (Unger, Markert and Müller, 2018, p. 255). However, even with regulation, these robots outperform human counterparts in tedious tasks. In the forthcoming future, more industries are liable to incline towards the use of cobots, from food processing to manufacturing, construction, healthcare, the chemical industry even hotel and restaurants. Some of the cobot purposed for industrial applications for pick and place and drilling are presented in **Figure 1** Collaborative robots (Universal Robots UR16e, Kuka LBR iiwa 14 R820).



**Figure 1** Collaborative robots (Universal Robots UR16e, Kuka LBR iiwa 14 R820)

## Mobile robots

According to Rubio, Valero and Llopis-Albert (2019, p. 1) "Mobile robots are categorized as robots possessing the ability to move autonomously with enough intelligence to anticipate and make a decision based on the perception it receives from the environment." Although current process lines have been seen using mobile robots, they differ in the category of intelligence. Automated guided vehicles (AGV), operates in a central hierarchical structure using GPS or guidance devices to compute or plan a predefined path when placed in a controlled space. In contrast, Automated mobile robots (AMR) operate autonomously in a rather dynamic real-time data for routing and scheduling. (Fragapane *et al.*, 2020, p. 5.) The latest generation of mobile robots is designed using new techniques, implementing machine learning and image recognition as a function for visual SLAM algorithms to carry out tasks in the dynamic environment. The capability to navigate "autonomous" without the need for physical or electro-mechanical guidance devices has brought a considerable advantage for exploratory missions in inspection and maintenance. Some of the applicable mobile robots for logistics environments are shown in **Figure 2** Autonomous Mobile Robots (MiR, 2019c; GreyOrange, 2020; Kuka, 2021). below.



**Figure 2** Autonomous Mobile Robots (MiR, 2019c; GreyOrange, 2020; Kuka, 2021).

Mobile robots are gaining a massive market share with a valuation of about USD 1.9 Billion in 2019 (Grand View Research, 2021) as a solution for service robots for agriculture, medical care, mining, surveillance, focusing primarily on intralogistics 4.0. They have become more common in commercial services, mainly due to the 2019 pandemic; however, they still have

massive significance among military applications and industrial settings. Depending on the robot's terrain and potential use, whether inland air or water, the robots are designed accordingly with legs, wheels, wings, or fins (Rubio *et al.*, 2019, p. 2). Implementing mobile robots, whether in a commonplace for as simple as food delivery services or in process lines to move materials from stocking shelves to order fulfillment zones in factories, has significantly improved efficiency. The current industrial application for mobile robots is to systematically organize the flow of goods between static production lines and production cells. It is also used to handle warehouse logistics speeding up the task and workflow between stations. (Rubio *et al.*, 2019, pp. 1-3.) IKEA, GEODIS, Amazon are few companies using mobile robots to speed up and facilitate their warehouse workflow.

#### Mobile Manipulator Robots (MMR)

MMR is a hybridization between two different robots AMRs and articulated manipulators. The drawback of articulated manipulators has a fixed base to the ground, which limits the workspace. Mobile robots can act as a conduit to transport articulated manipulators from one point to another. The development of hybrids autonomous mobile manipulators has focused on robotics' research fields for the past two decades. Although the concept is not revolutionary; however, its implementation in a dynamic environment while coaxially existing to execute tasks with accuracy and safety is innovative and challenging. These would bring a substantial number of benefits mentioned for large and small-scale manufacturing industries, as stated in Unger *et al.*,

- Higher flexibility in processes
- Improved ergonomic scoring of close by workstations
- Improved degree of capacity utilization for robots
- Higher economic efficiency and process stability
- Automatization of currently not automatable processes. (Unger *et al.*, 2018, p. 258)

Additionally, autonomous mobile robots also paved an increase in research covering various studies from visual recognition, sensor fusion, control mechanics, SLAM algorithms, and optimization. The mobile manipulators can be reviewed as interactive robots designed to assist humans as professional service robots applicable for flexible and cognitive manufacturing applications in factories and, hopefully, commercially available for home applications. A human variable that accounts for unpredictability in tedious routine functions

is one element that affects the level of efficiency in low-skilled work. It would be reasonable to assume that the use of robots will decrease low-skilled jobs with industrial automation.

There are almost 26 companies that have developed their version of MMRs and are testing its flexibility. KUKA youBot is an open-source mobile manipulator specifically designed to bridge knowledge into understanding robotics in research and education. One example of automated application in research is a fully functional autonomous KUKA cobot assembled with a mobile unit functioning at the University of Liverpool. It operates at its chemist lab using an AI optimization algorithm to make decisions, assist, navigate and safely work around humans.

### Trajectory planning

Depending upon a type of robot manipulator, it can move in numerous ways to retain a point inside its workspace. Robot movements in space also rely on physical limiting factors such as the robot's arms reach, the number of joints, and the types of joints available in the robot. For an articulated manipulator, the typical design consists of 5 to 7 degrees of freedom. Generally, for a 6 DOF robot, three DOF are used for the position and the remaining three for the end-effector orientation. The joint variable (angle  $\nu$ ) for rotation and the (displacement  $d$ ) for translation is a part of the kinematic model to describe the pose of the robot. When going from one pose to another, the robot should be ensured with a motion of flexibility and smoothness. The goal of trajectory planning is to generate a smooth change in values of joint variable and translation values to avoid jolting when moving the end-effector to the desired position. As DOF increases for a system, the final assigned posture and the position can be achieved using a different combination of  $\nu$  and  $d$  values. The multiple possible combinations for different routes which follow a trail of points in a joint space can be categorized as trajectory paths. Typically the interpolation of the trajectory between the two points is designed to have the minimal time required for a manipulator to move from an initial pose to an acquired pose. The fundamental process for such a trajectory depends on few factors. The motion should be such that the actuators do not exceed generalized force (manipulator dynamics) and should not exert the joints to saturation, and the motion should be suitably smooth. The trajectory is planned with constant acceleration followed by a parabolic transition to a constant velocity and finally again a parabolic transition to constant deceleration at the end phase of the path.

As the path's complexity increases, especially in curved surfaces or obstacle avoidance, the end-effector can be subjected to move across a set of points. The interpolation is done based on trapezoidal velocity profiles consisting of linear segments and parabolic segments representing a transition in between.

### Optimization

It is a technique to deduce the best choice based on some criteria from a set of available alternatives. The optimization problem identifies the decision variable, which can be continuous, semi-continuous, discrete, or sets. The range of values is usually within the bound for physical systems. The objective function rationalizes the optimization by giving it a specific relation to decision variables. According to Hillier and Hiller (2003), "It can be linear or nonlinear, differentiable or nondifferentiable, concave or convex". It is a criterion concerning which the decision variable is to be optimized. Every solution in variable space is mapped to an objective. The objective function is usually minimized or maximized depending upon the problem and finally filtered with constraints. The constraints layout restrictions or produce limitations to the result. It can be inequality or equality constraints. (Hillier and Hillier, 2003, p. 11.)

Based on constraints, a solution can be classified as a feasible solution that satisfies all the constraints or an infeasible solution that cannot satisfy at least one constraint. There are possibilities that due to constraints, a solution cannot be achieved, and therefore, the constraints have to be neglected or omitted. Depending upon the system's flexibility, some can be observed as hard constraints that must be satisfied to accept the solution and others as soft constraints that can be allowed to relax to some extent to accept the solution.

Optimization can be classified into three main techniques mathematical programming, Metaheuristic techniques, and hybrid techniques (Rodríguez *et al.*, 2018, p. 1). The mathematical techniques are mainly based on the problem's geometrical properties where the algorithm is designed depending on the problem's nature. It can be used explicitly for linear program problems using a simplex algorithm as it will have a linear objective function and linear constrain problem (Hillier *et al.*, 2003, p. 11). Similarly, Metaheuristic techniques are nature-inspired techniques that do not necessarily exploit the nature of the problem. The

techniques found under metaheuristic techniques consider a black box problem. (Yang, 2011, pp. 21-22.)

## 1.2 Motivation

Although industries are improving by increasing automation to secure safety and decrease lead time for manufacturing a product, the projected level of machine intelligence for dynamic changes in the production line does not compete with the interactive human intelligence element. There is always a need for a human interface to navigate the situation. The only model that can surpass this scenario is predicting and utilizing algorithms to generate a simulated model for navigation of workspace. This level of sophistication is achieved only with the mobility and trajectory planning of the robot. Thus a hybrid system of combining a mobile robot with an articulated manipulator has a vast potential to solve the challenges faced in automation. Furthermore, these hybrid setups have been used in disaster situations to work with safety personals for rescue missions.

## 1.3 Objective

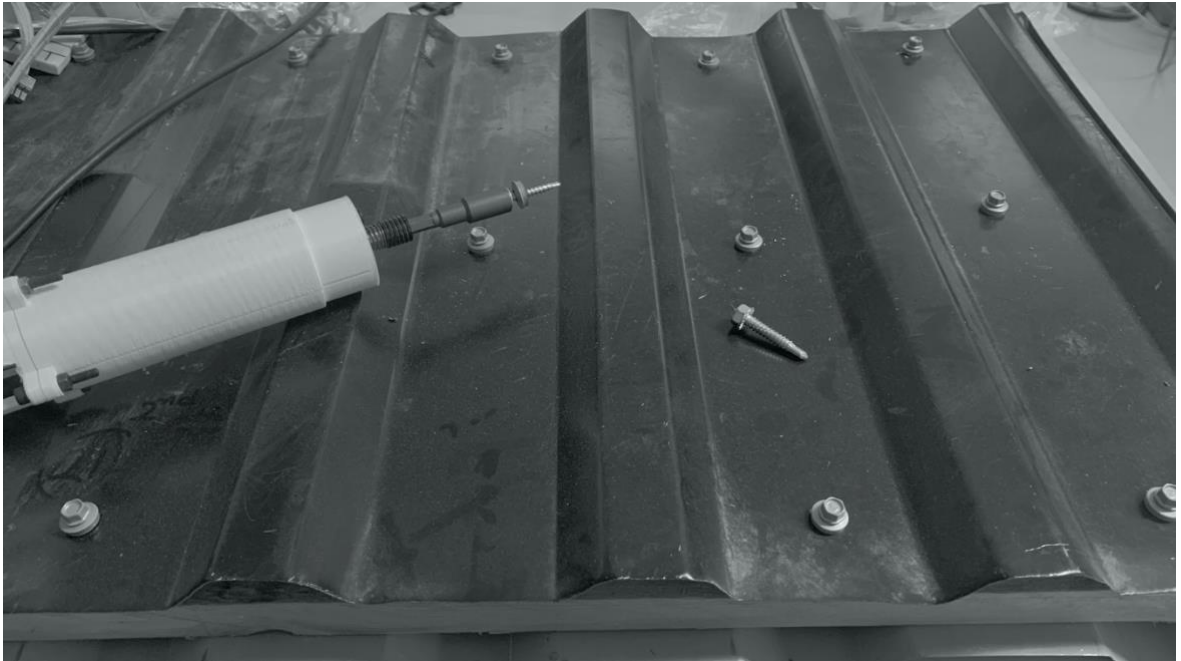
Obtaining an interactive robot mechanical system that can facilitate performing tasks while working in a dynamic environment has been a long-time challenge. The industries are always perusing to reduce cost and time to maximize profit. The new advancement in force sensor manipulators and autonomous robots has been building up designs to produce a simplistic android bot that can assist humans in the workplace. The mobile robot's case is specific to locomotion, confined to 2D space while having no significance to work on the z-axis. On the other hand, a cobot has limitations to its maximum reach to a confined workspace. This project aims to utilize this assistive robot as a single unit and implement it to reposition the hybrid system depending upon the task in drilling on the roof sandwich panel in the workplace. The concept of using such a hybrid to work and assist alongside humans is trending. The system can also be made to work continuously throughout the day to carry out missions without human interference. This thesis aims to combine both robot MIR200 and UR16e to create a new workspace and relate a method to utilize positioning techniques via the geometrical or algorithmic objective function for optimizing mobile robot positioning in this workspace.

## 2 METHOD

### 2.1 Applied research

It is generally observed that when working in a large assembly, the person needs to move around the workspace to complete the task, for instance, when cleaning a table with a cloth. It is generally good to move the body around the table rather than extend the arm far wide. Similarly, the limitation of articulated manipulator needs to be repurposed via mobile to correct and efficiently carry out functions. The locomotive parameter of a mobile platform increases the workspace of the robot arm. An extensive volume application can be resolved using such a system. During this research, an autonomous mobile manipulator system was developed to demonstrate experimentally to an application, whereby a mobile manipulator unit would move around a table with a sheet metal plate of (1.2 x 1.5)sq m placed over it for drilling.

The system was integrated between two independent units, collaborating and working coincidentally to perform a repetitive task of drilling around a fixed shape of sheet metal panel. These panels are used as a roofing material to insulate the house from rain and snow. The panel shown in Figure 3 Sheet metal roof panel (Ruukki T20-29W-1095) has a groove profile and needs to be drilled in different locations. Since the sandwich panel is placed on patterned wooden columns, the placement of drill points is always equal. The drill placement could range between 400mm to 1000mm apart, depending upon the panel's profile. The concept of making these drill points as a function in a workspace could be utilized as the root equation to preplan the position for the mobile robot. Initially, a sticker will be tagged on the panel. The sticker could be defined as a location for one of the drill holes. Using the sticker's location and the grooves' spacings, a pattern of coordinates can be generated for the panel's surface. The table's center's location can act as a reference in a global frame mapped in the mobile robot.

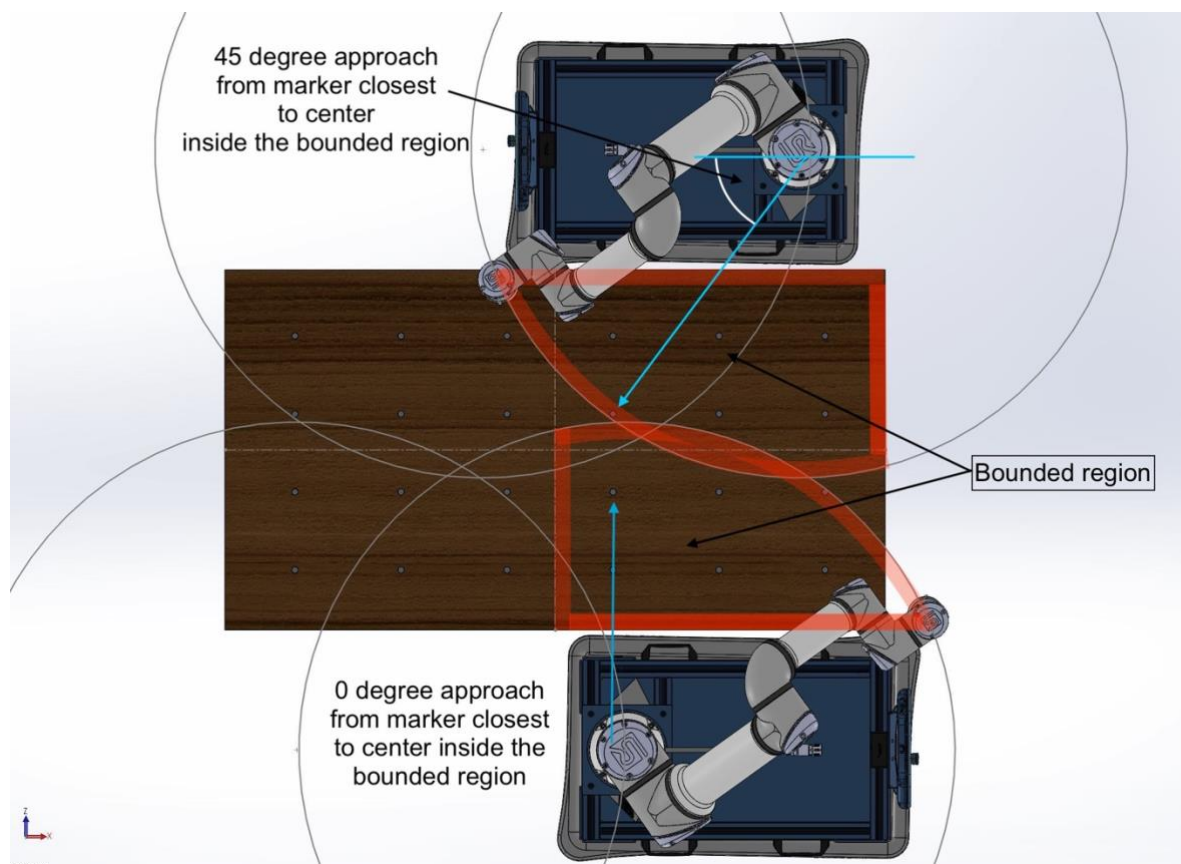


**Figure 3** Sheet metal roof panel (Ruukki T20-29W-1095)

This project's concept is to utilize all the input information and determine the best suitable position for the robot to be located to carry out drilling while covering maximum numbers of drill points. When the marker's location is received from the overhead camera, it will describe the first drill's hole position in the local map. The other drilling positions are positioned in some equidistance in the x and y direction plane, referencing the marker position. These imaginary drill positions are set in the local frame relative to the marker frame. Using the local frame and drill positions, the taskspace can be sectioned into workspaces, or the whole taskspace can be categorized into sections irrelative to the drilling positions. There have been two approaches to conduct this process.

In the first approach, few constraints were assigned. The robot could not approach head-on to the taskspace due to the borderline placement of UR16 on the mobile platform. Whenever the robot arm rotated towards the heading direction, the mobile robot would lose balance, and the whole unit would roll over forward. The robot could only approach the taskspace sideways along the horizontal length of the table. The tagged marker position was used to generate a list of values that were sets of x and y coordinates. The values were stored in a list. The table's dimension was known; the table was divided into four quadrants similar to the graphical quadrants with positive and negative X and Y values. The geometrical origin became the center of the table. The coordinates were filtered and sectioned into different list

specific to the points which lie in the respective quadrants. These four sets of the coordinates list were prepared to be targeted from different locations expressively. Each quadrant was filtered for one specific coordinate, which was the closest to the table's center. Taking this coordinate as the reference point, two modes were implemented, a cosine angle of 45 degrees with maximum arm reach length was calculated on an outward diagonal position away from the center, and a zero degree approach, as shown below in **Figure 4**. The resulting robot's location was almost at the four corners of the table, which meant it was more accessible for the robot arm to reach the targeted position. The positioning of the mobile robot was cover the maximum taskspace that laid inside the quadrant. This method enabled to satisfy the required criteria to accomplish the given task. This process also considered the targeting of all the positions that could be reached from a single position and obeyed the constrain.



**Figure 4** Positioning of MIR via angular approach in reference to marker position

The second approach was taken into consideration to a limitation that arose in the first approach. In the beginning, the approach to the problem fixated on numerous constraints, such as the shape of the table was a rectangle with a fixed dimension of 1.2m x 1.5m; The

manipulator could only be actuated in sideways movement as there was not sufficient counterweight to balance the unit. The mobile robot's positioning was almost predefined, limiting to four corners of the table, minimizing overlap while maximizing taskspace. Since the whole table was sectioned into four parts, the program was limited to only four locations and would fail if the table was round or with greater length.

The mobile unit's position needed to be calculated in a more novel method to accomplish the goal for a variable length of the table. Depending upon a taskspace, a suitable method of dividing a table was selected, independent of both the table's dimension and shape. In the second method, the taskspace was overlaid with circles that represented the UR16 robot workspace. The circles could be consecutively placed along with each other with some overlap to cover the taskspace. The level of detail for the coverage of taskspace by the workspace of the UR relied on the ratio of reachable to the unreachable area of taskspace. If the coverage area outside the workspace had no drill position near the intersection of the three workspaces, the overlap could be smaller. Similarly, as the first position and the last position along the length can be determined using this technique, the in-between position along the length can be obtained with a minor modification to the same overlap technique. The  $(2^{nd}) \dots (n^{th}-1)$  position along the length will be governed with an overlap of four consecutive circles of the robot's workspace in balance with unreachable space in between them.

An objective function was created using this technique. The area of the sectors or the area of overlap was taken and compared to the area bounded by the three circles' intersection. A second objective function was created to govern the in-between positioning of the robot using the four circles. Both functions were fed to the Differential Evolution and Teaching Learning Based Optimization algorithm; each algorithm used the object function and deduced the minima value that could satisfy the objective function.

## 2.2 Robots

### MIR200

A mobile robot model is always based on one principle consisting of three essential factors sensing, planning, and locomotion (Rubio *et al.*, 2019, p. 2). Generally, mobile robots use wheels as the locomotion mode since it can be represented with the simplest kinematic

model. MIR is built on a single chassis integrated with nonholonomic differential drive wheels subjected to a single Pfaffian velocity constraint  $A(q)\dot{q} = 0$ . The configuration space of a mobile robot frame  $\{\text{mir}\}$  relative to global frame  $\{G\}$  can be represented as,  $T_{\text{mir}}^G = SE(2)$ . The wheels' configuration is represented  $q = (\phi, x, y)$ , and the velocity can be represented as the time derivative of  $q$  as stated in eq 2 below. (Lynch *et al.*, 2017 pp. 441-449.) In addition, to the kinematic model of the mobile robot from wheels, there are other components that need to be incorporated for path planning. The depth camera sensors, which provide a detailed description of the environment, the accelerometer, gyroscope, ultrasonic sensors, and Laser scanners placed in front and the rear of the robot to determine the presence of obstacle are used to enhance the sensory parameters that may have been overlooked from the vision camera. The wheels motor contains an encoder that is used for odometry and trajectory following. The data from all the sensors are intelligently fused into the algorithms. It is analyzed and processed most productively and stored locally in the robot, in the cloud, or even in remote, for integrating into a map that can be used for navigation and path planning. The robot system program is powered with artificial intelligence working with deep learning and image recognition that provide additional intelligence data for the robot to anticipate needs and proactively adapt its behavior. (MiR, 2019a.)

All mobile robots with nonholonomic wheels are governed by the form  $q' = G(q)u$  that can be related to the robot's position and orientation in the map. The two differential-drive wheels that are independently driven about the same axis are connected to the MIR chassis. The controller is connected to brushless motors, gear, and wheel assembly, directly attached to the modular MIR frame. The robot turns by varying the speed of the left and right motors of the wheel. The extended caster wheels at the four corners are to stabilize the platform and are driven wheels. The relation of the robot's different frames when it moves from one position to another can be obtained using a straightforward method. If the distance between the drive wheel is taken as  $2d_l$  and the MIR position in-plane space is represented by  $(x_l, y_l)$  at the center between wheels. The configuration can be stated as  $q = (\phi, x_l, y_l, \theta_R, \theta_L)$ . The rolling angles of right and left wheels are represented by  $\theta_R$ , and  $\theta_L$ . The kinematic equation of MIR can be given by eq 1 as follows. (Lynch *et al.*, 2017, p. 450.)

$$\dot{\mathbf{q}} = \begin{bmatrix} \dot{\phi} \\ \dot{x}_1 \\ \dot{y}_1 \\ \dot{\theta}_L \\ \dot{\theta}_R \end{bmatrix} = \begin{bmatrix} -r_1/2d & r_1/2d \\ 0.5r_1\cos\phi & 0.5r_1\cos\phi \\ 0.5r_1\sin\phi & 0.5r_1\sin\phi \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_L \\ u_R \end{bmatrix} \quad eq 1$$

Where, the angular speed of the left and right wheels are represented by ' $u_L$ ' and ' $u_R$ ' and ' $r_1$ ' is the radius of the diff-drive wheel. After excluding the wheels' rolling angles, the last two rows of the matrix can be eliminated to form a somewhat simplified control system given in *eq 2*. (Lynch *et al.*, 2017, p. 450.)

$$\dot{\mathbf{q}} = \begin{bmatrix} \dot{\phi} \\ \dot{x}_1 \\ \dot{y}_1 \end{bmatrix} = \begin{bmatrix} -r_1/2d & r_1/2d \\ 0.5r_1\cos\phi & 0.5r_1\cos\phi \\ 0.5r_1\sin\phi & 0.5r_1\sin\phi \end{bmatrix} \begin{bmatrix} u_L \\ u_R \end{bmatrix} \quad eq 2$$



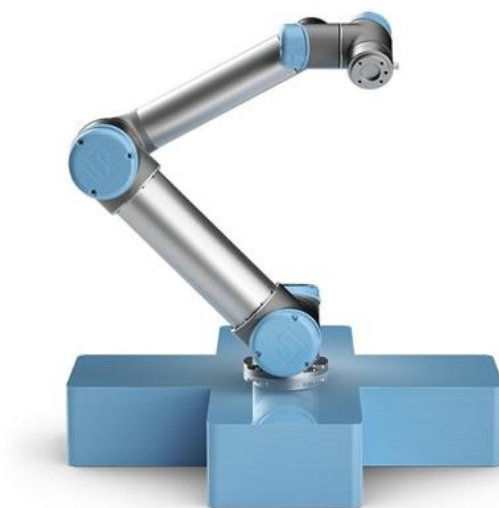
**Figure 5** MIR200.(MiR, 2019c).

The mobile robot MIR200 shown in **Figure 5** MIR200.(MiR, 2019c). uses dual laser scanners placed on the front and rear corners to live-detect obstacles and two 3D cameras in front to detect obstacles up to 1,700 mm height for localization and mapping. (MiR, 2019b). The 3D point cloud created from all the information gathered by every pixel, color, and depth values is applied to the localization algorithm based on observation models to filtered cloud points projected to the 2D vector map. The data from the full sampled point cloud and sensor

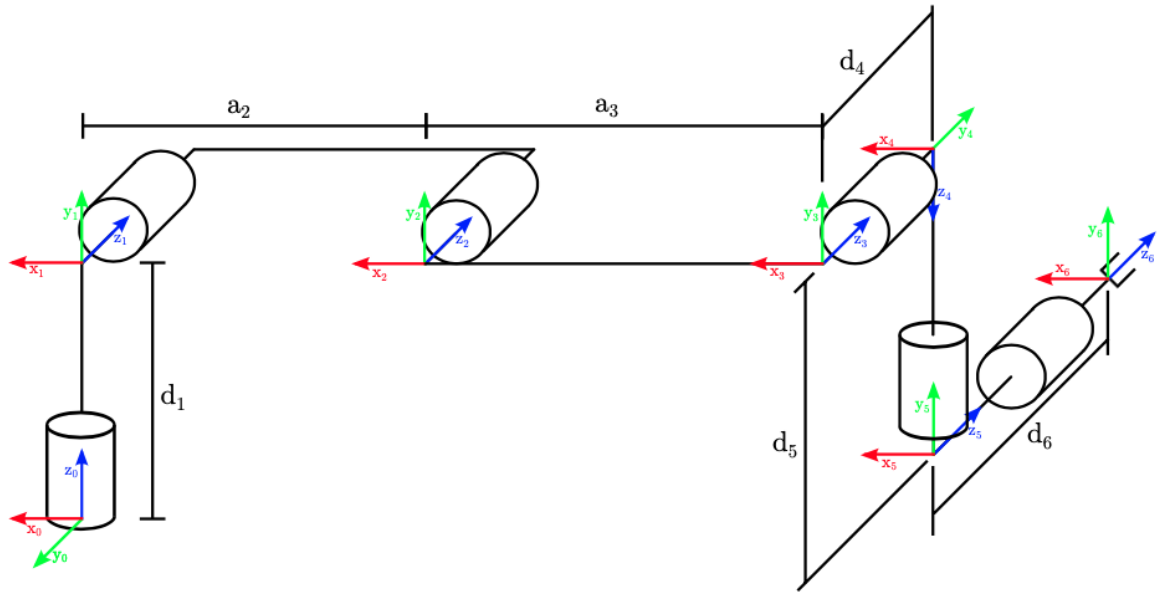
reading from other sources ultrasound, odometry is processed with sensor fusion for obstacle avoidance in autonomous navigation. (Biswas and Veloso, 2012, p. 1699).

### UR16e

The articulated manipulator is designed with a 6R configuration. The joints are directly driven by a brushless motor. Due to its 100:1 zero-backlash harmonic drive gearing, it is safe to handle a high torque at reduced speeds. It is equipped with force sensors. It is among the companies strongest cobot with a heavy-duty payload of 16 kg, thus able to handle vibrations along with increase safety features. The cobot operates on Robot Operating System (ROS) with industry-standard Universal Robot Description Format (URDF) is an XML (extensible Markup Language) file format. (Lynch *et al.*, 2017, p. 125). It has inbuilt safety functions, including customizable safety functionalities. The cobot has intuitive as well as script programming. Data can be sent and received from systems to the robot via a socket connection. The UR16e has the safety certification of EN ISO 13849-1PLd category 3 and EN ISO 10218-1 used in a commercial application. (Universal Robots, 2020a). **Figure 6** Cobot UR (Universal Robots, 2021a).shows the esthetics of the UR16e, and the kinematic diagram of the articulated UR robot series is presented in **Figure 7** The kinematic diagram of the UR16e robot .



**Figure 6** Cobot UR (Universal Robots, 2021a).



**Figure 7** The kinematic diagram of the UR16e robot (Keating, n.d.).

**Table 1** DH parameter table below shows the DH parameter of the universal robot, utilized to derive the forward kinematics of the universal robot.

**Table 1** DH parameter table (Universal Robots, 2021b).

Kinematics	$\theta(\text{rad})$	$\alpha(\text{rad})$	$a(\text{m})$	$d(\text{m})$
Joint 1	0	$\pi/2$	0	$d_1$
Joint 2	0	0	$a_2 = 0.4784$	0
Joint 3	0	0	$a_3 = 0.3600$	0
Joint 4	0	$\pi/2$	0	$d_4 = 0.17415$
Joint 5	0	$-\pi/2$	0	$d_5 = 0.11985$
Joint 6	0	0	0	$d_6 = 0.11655$

The inverse kinematics of the articulated manipulator robot can be obtained using a geometrical approach whereby each joint angle can be related to the positional frame. The inverse kinematics of the UR16e is designed in reference to inverse model presented by Ryan Keating for UR5.

### Singularity

Singularity is a common problem that is found with open-chain robots. The postures at which the end effector loses Jacobian rank are known as singularity. One should always avoid such posture as it is challenging for the robot to move to the intended trajectory during singularity (Lynch *et al.*, 2017, p. 163). There are two types of singularity conditions: boundary where the robot arms are stretched to move outside the robot's workspace and the internal singularities due to the robot's two or more axis's alignment. Mathematically, during singularity Jacobian  $J(\theta)$  fails. The geometrical Jacobian is shown in the equation below. (Craig, 2004, pp. 151.)

$$v = \begin{bmatrix} \dot{p} \\ \omega \end{bmatrix} = \begin{bmatrix} J_P \\ J_\omega \end{bmatrix} \dot{\theta} = J_G \dot{\theta} \quad eq\ 3$$

### Modelling of MIR and UR kinematics

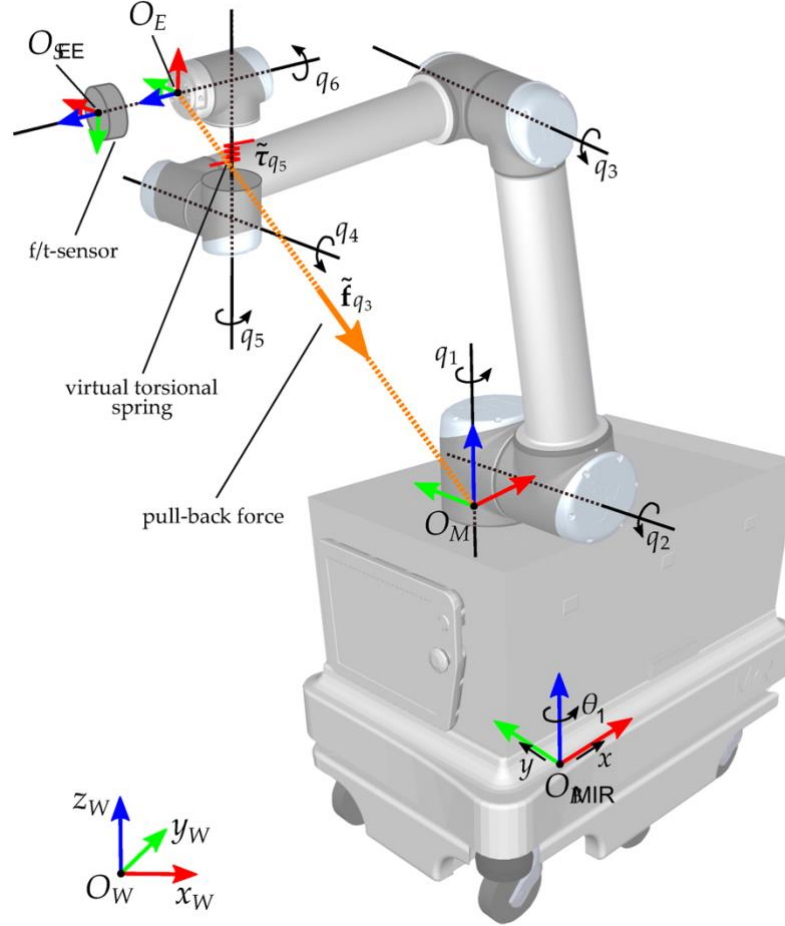
Although it is quite natural for humans to move both body and arm together to reach an object, it is not as precise in mobile manipulator robots. Both the movement has to be working relative to one another. The complexity of the system increases, and as a function of time and it is better suited to divide the motions into separate parts. The arm's motion is more precise; the mobile unit is first deployed to the nearest position, and the manipulator activates to perform a precise task. According to Lynch *et al.*, (2017, p. 125) the configuration of end-effector frame  $\{O_{EE}\}$  with a base frame  $\{O_M\}$ , chassis frame  $\{O_{MIR}\}$  in the fixed global frame  $\{O_G\}$  can be defined by the following expression

$$X(q, \theta) = T_{O_G O_{EE}}(q, \theta) = T_{O_W O_{MIR}}(q) T_{O_{MIR} O_M} T_{O_M O_{EE}}(\theta) \in SE(3) \quad eq\ 4$$

Where  $\theta \in \mathbb{R}^n$  is the set of arm joint positions for the n-joint robot. (Lynch *et al.*, 2017, p. 125).

The base frame of the UR is in a fixed placement position to the central axis of the MIR robot. The overall guidance of the UR robot is dependent on the positioning and orientation of the MIR. Based on forward kinematics, the relation of UR base changes relative to MIR; therefore, an initial transformation matrix is introduced that represents the rotation and the translation of the robot. The rotation about the Z-axis  $ROZ(\theta_i)$  can represent the rotation

matrix of the MIR, and the forward kinematics matrix can be post-multiplied to relate the overall transformation matrix of the system from the ground to the end-effector of the UR robot as seen in the figure below.



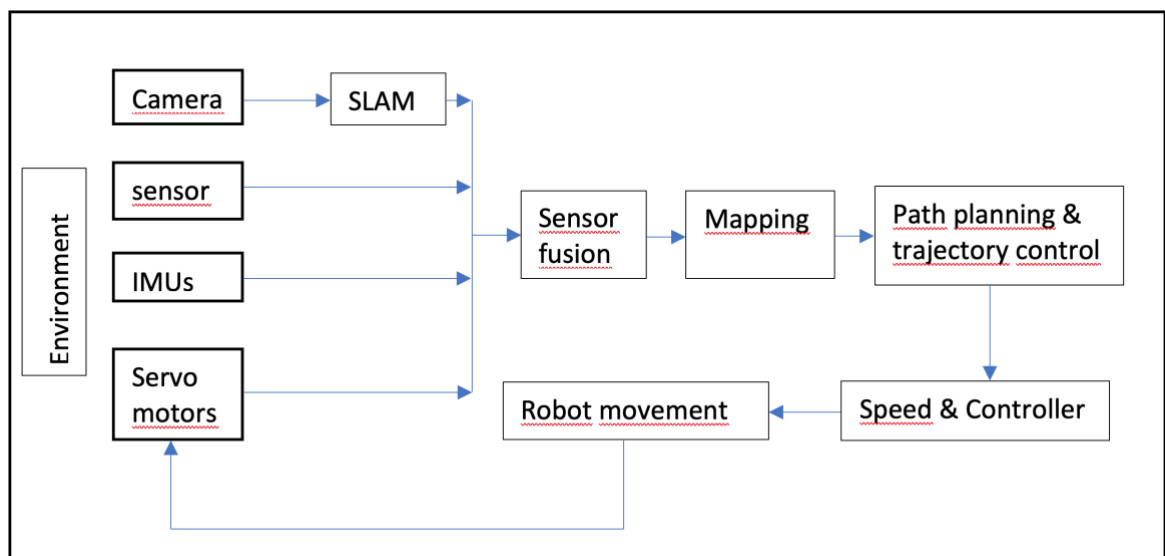
**Figure 8** Frame representation of the MIR200 and the UR16e (Weyrer, Brandstötter and Husty, 2019, p. 14).

The homogenous transformation matrix of the given system can be represented mathematically,

$$T_{UR}^{MIR} = \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 & P_x \\ \sin(\theta_1) & \cos(\theta_1) & 0 & P_y \\ 0 & 0 & 1 & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} n_x & o_x & a_x & P_x \\ n_y & o_y & a_y & P_y \\ n_z & o_z & a_z & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad eq\ 5$$

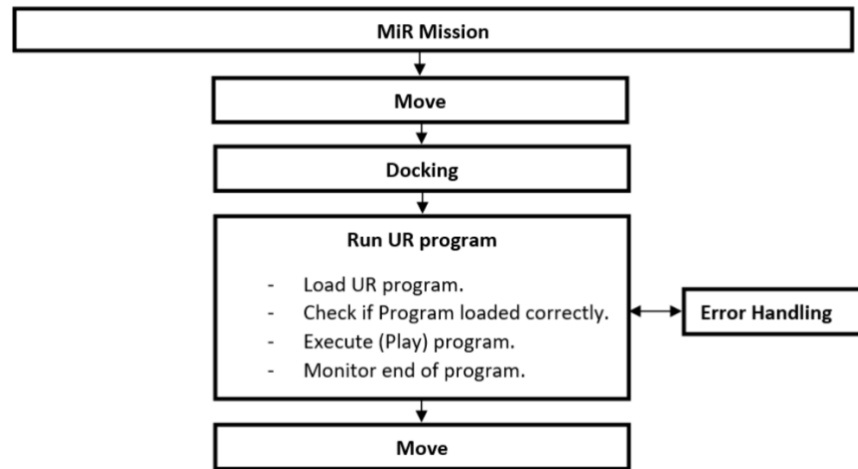
### 2.2.1 Design and architecture

The Mobile Manipulator is designed using MIR200 with a Universal robot UR16e. The chassis developed by Vention specific to MIR200 is fitted on the top of the mobile robot with four M16 bolts. The chassis houses a control box, an inverter, and connectors between the two systems. The articulated manipulator UR16e is fixtured on the top of the chassis. The power supply to the UR16e is deployed from the mobile robot battery via an inverter. The Serial robot is connected to a mobile robot via a Local area network (LAN) which sends commands to UR16e from the Mobile robot interface. The graphical user interface (GUI) of the mobile robot is set up with UR16e via an IP connection. The UR is set on remote mode. Similarly, the UR is also set up on the mobile robot GUI where the assigned IP address is stored in the MIR-UR configuration settings. The following **Figure 9** Framework of the mobile manipulator shows the framework of the system.



**Figure 9** Framework of the mobile manipulator

The overall control of the system always remains in the MIR firmware system. The general program can be designed at the MIR GUI interface, and program control flow can be sent to the UR system. The MIR dominates the UR system as it controls the initial steps of the program. The MIR and the UR work in a master-slave relationship. The following **Figure 10** Control flow between MIR and UR (MiR, 2018). shows the control flow of the system between the two robots.



**Figure 10** Control flow between MIR and UR (MiR, 2018).

In theory, a client can program individual task depending upon the sophistication of programming for both MIR and UR independently and use individual flow control to make a functional program for the system to work simultaneously but, due to access restriction posed by the MIR system, there has been a limitation to access all the keys. A preprogrammed script can be added to the UR system, and a MIR system can command the UR to run a script file from its MIR mission control.

The program was developed in python version 3.8. LAN connector of Universal robots(UR), as connected to the router inside the MIR system. The wireless network (Wi-Fi) of mobile robot MIR200 and its GUI are structured together with remote access mode from UR is set to send and receive data from both robots via REST API.

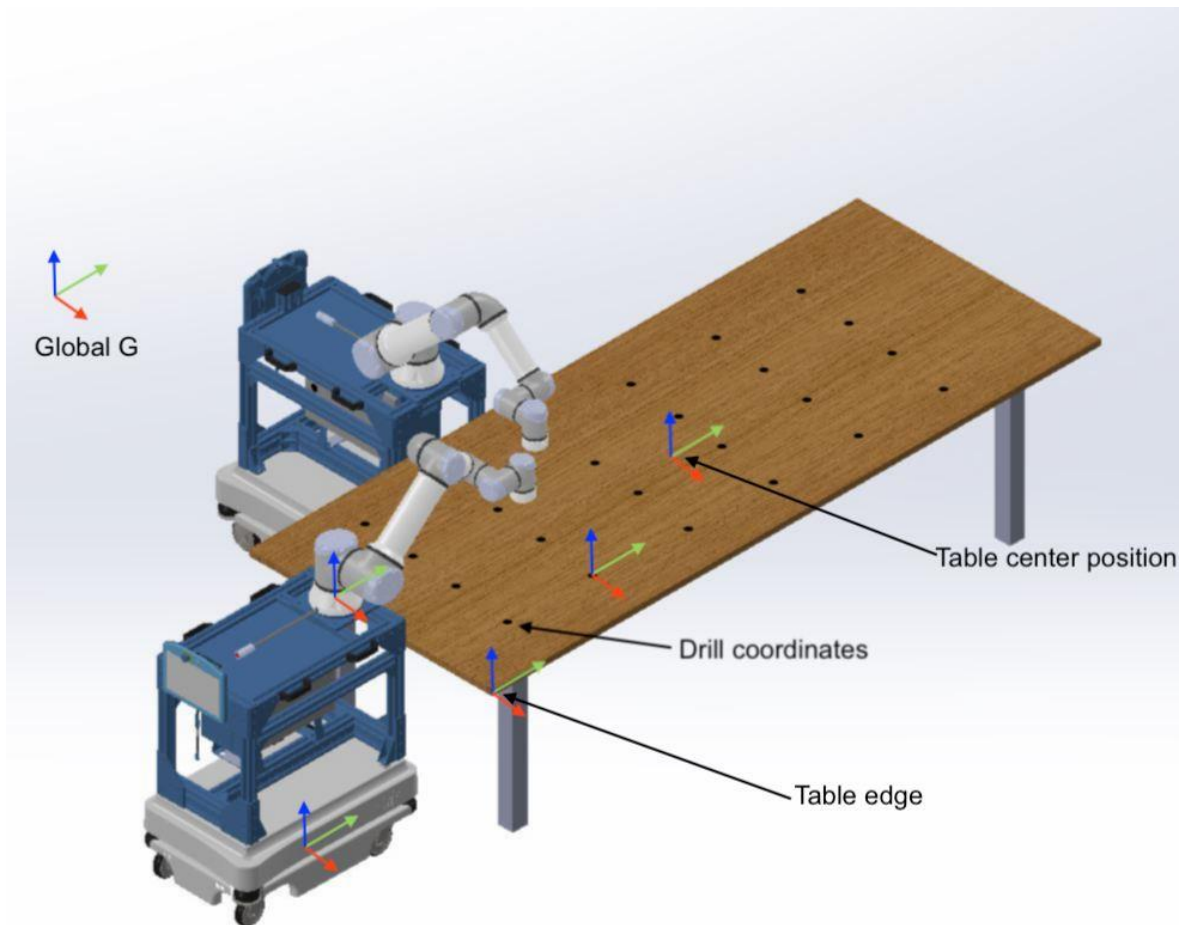
### 2.2.2 Transformation matrix

The transformation matrix provides a reference vector to relate drill positions to the tool position of the EE. The task of drilling at different locations using a mobile manipulator can be done sequentially by distributing drill positions into different sections of the workspace. The drill gun can target all locations which fall inside the workspace from one suitable position. The drill position is represented into the transformation matrix, which can be related in reference to the global frame. With the following information, the inverse kinematics of UR tool center can be strategically used to find the position of the UR base frame and the position of MIR in the global frame. After reaching the mobile robot's location,

the manipulator (UR16e) activates and requests drill coordinates. The client program running on python takes one set of coordinates from the list and converts to bits format. The bits value is sent to the UR program, and the UR EE is directed to the drill position. After the mobile robot has reached the location, serial robot (UR16e) activates and requests drill coordinates. Client program running on python, transforms the location of drill coordinate relative to UR base frame and sends the pose values in the form of bits. The UR end-effector moves to the position, sequentially targeting all the holes that fall inside that quadrant. The process is in a loop until the task is completed. **Figure 11** Graphical representation of frame transformation a graphical representation of the frames for the robot and the object relative to global frame.

Mathematically,

$$T_{UR}^{Drill\_pose} = T_{UR}^{MIR} T_{MIR}^G T_G^{Table} T_{Table}^{Table\ edge} T_{Table\ edge}^{Drill\_pose} \quad eq\ 6$$



**Figure 11** Graphical representation of frame transformation

## 2.3 Optimization

The evolutionary algorithms (metaheuristics) is a mathematical computation to use a higher-level procedure to find and generate a solution based on a generic population. The model follows the biological evolution on how recombination with mutation can evolve to reproduce a better selection. The idea is applied in different variations to optimize problems that often perform reasonable approximation solutions. There are different series that underly with the principle of evolutionary algorithms such as Genetic algorithm, Differential evolution, Particle swarm optimization, Evolutionary programming, Neuroevolution, and others which follow a path of evolving generation. (Xiao, 2009, p. 410.)

The implementation of these models usually follows a random selection of individuals from a population taken from the first generation. The fitness of each individual is evaluated breed with new individuals through crossover and mutation operation to reproduce a new offspring, and they replace the least fit individuals in the population. The process is followed depending on termination until conditions such as time limit or sufficient fitness are achieved. (Xiao, 2009, p. 411.)

### 2.3.1 Differential Evolution (DE)

The differential evolution is a very powerful algorithm for optimization to find a minimum of a given function. The DE algorithm works on regenerating a population of a viable solution to an existing initial population of all the candidate solutions in its basic form. These solutions are randomly selected and under relevant mathematical formulae are rearranged in the search-space. It combines the existing candidate from the population and reconfigures a new position in space. Depending upon the improvement, it is either added to the new list of the population or simply discarded. The process is repeated in a loop, hoping to find a satisfactory solution eventually; however, it is not guaranteed that a solution will eventually be discovered. (Huang and Chen, 2013, p. 2.)

As mentioned by Kok and Rajendran, and introduced by Storn and Price, the optimization model is based on a Genetic Algorithm(GA), which involves three distinct operations, mutation crossover, and selection processes. The population is randomly chosen within a search space and undergoes mutation, and an individual  $v_i^{G+1}$  is generated.

$$v_i^{G+1} = x_{r_1}^G + F(x_{r_2}^G - x_{r_3}^G), \quad r_1 \neq r_2 \neq r_3, \quad i = 1, 2, \dots, NP, \quad eq 7$$

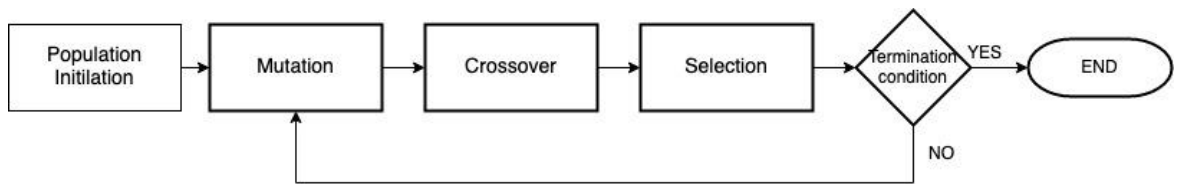
Where  $x$  is an individual from population  $r \in (1, NP)$ , and the  $G$  represents the iteration,  $NP$  is the overall population size, and  $F$  is the differential weight. Depending upon the crossover probability, only selective will be used in the next operation. The new population of individuals with crossover is produced as  $u^{G+1}$  over the following condition.

$$u_{ij}^{G+1} = \begin{cases} v_{ij}^{G+1}, & rand_{ij} \leq crossover \\ x_{ij}^G, & otherwise \end{cases} \quad j = 1, 2, \dots, D, \quad eq 8$$

Where  $rand_{ij}$  is a random value between the interval  $[0, 1]$  for the  $i^{th}$  individual at the  $j^{th}$  particle. The population goes through the selection process while comparing with different iteration.

$$x_i^{G+1} = \begin{cases} u_i^{G+1}, & f(u_i^{G+1}) \leq f(x_i^G) \\ x_i^G, & otherwise \end{cases} \quad eq 9$$

The process continues until the termination condition is reached. The control flow of the method is given below in the flowchart **Figure 12** Differential algorithm flowchart (Kok and Rajendran 2016).. (Kok and Rajendran, 2016.)



**Figure 12** Differential algorithm flowchart (Kok and Rajendran 2016).

The differential evolution control parameter is used primarily for numerical optimization. The objective function in the following path planning was programmed using a graphical approach of the workspace, as shown in **Figure 14** Optimization ratio of overlap to the uncovered triangle area. The area of coverage of the workspace by articulated manipulators from a different location was taken into account. The taskspace was primarily sectioned into

a rectangle that had three possible placements of the robot arm, first at the center position of one side on the rectangle, which would cover half a circle of the rectangle similarly, following two positions would be at the corresponding edges furthest away and opposite to one another.

### 2.3.2 Teaching learning-based optimization(TLBO)

The TLBO is an algorithm based on a model where a subject (student) is taught a content in two different modes from two different source teacher and other students in the class. This strengthens the value of content and, when a subject is evaluated, would surpass on the level of knowledge in comparison to a subject, based only on one mode of learning as described by Rao et al. Initially, there exist a teacher phase whereby a learner will acquire the best possible knowledge to some extent level of the teacher. (Rao, Savsani and Vakharia, 2012,. p. 4).

Let  $M_i$  be the mean value and  $T_i$  be the teacher at any iteration  $i$ . The teacher will move mean value as a result of his teaching in the class. Let  $M_{new}$  be the new mean value  $r_i$  represent a value between the interval  $[0,1]$  and  $T_F$  can be the teaching factor. The difference in mean value can be represented as follows.

$$diffrence\_in\_Mean_i = r_i(M_{new} - T_F M_i) \quad eq\ 10$$

Due to this difference, it changes the solution with the following expression.

$$X_{new,i} = X_{old,i} + difference\_in\_Mean_i \quad eq\ 11$$

In the learner phase, the subject's knowledge increases due to random interaction with other subjects (learners). There can be gain in level of overall knowledge as presented in the following process.

$$X_{new,i} = \begin{cases} X_i + r * (X_i - X_j) & \text{if } f(X_i) < f(X_j) \\ X_i + r * (X_j - X_i) & \text{otherwise,} \end{cases} \quad eq\ 12$$

For  $i = 1 : P_n$

Randomly select two learners  $X_i$  and  $X_j$  where  $i \neq j$

If  $f(X_i) < f(X_j)$

$$X_{new,i} = X_{old,i} + r_i(X_i + X_j)$$

*Else*

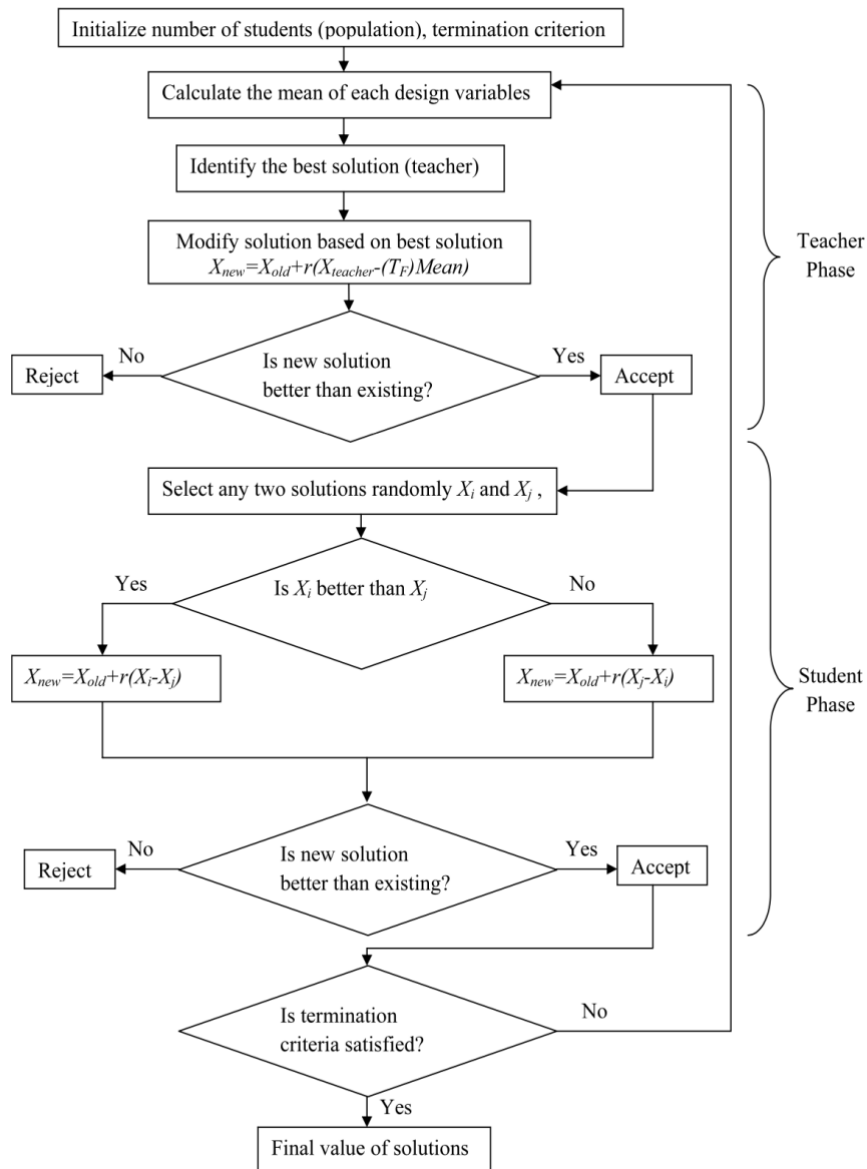
$$X_{new,i} = X_{old,i} + r_i(X_j + X_i)$$

*End If*

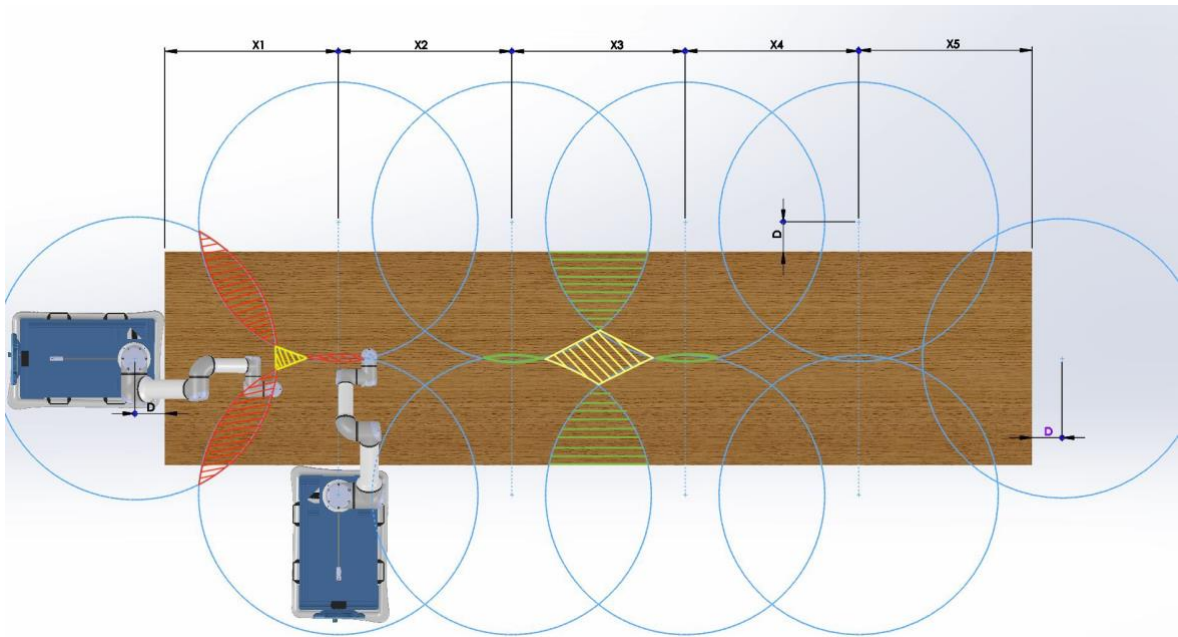
*End For*

Accept  $X_{new}$  if it gives better function value.

Based on this model of acquiring knowledge on from quality of teaching from teachers as well as from other learners leads to giving better results. (Rao *et al.*, 2011, p. 305). The method for elimination and iterations is presented in the flowchart below **Figure 13** Flowchart of TLBO optimization (Rao et al., 2011, p. 305)..

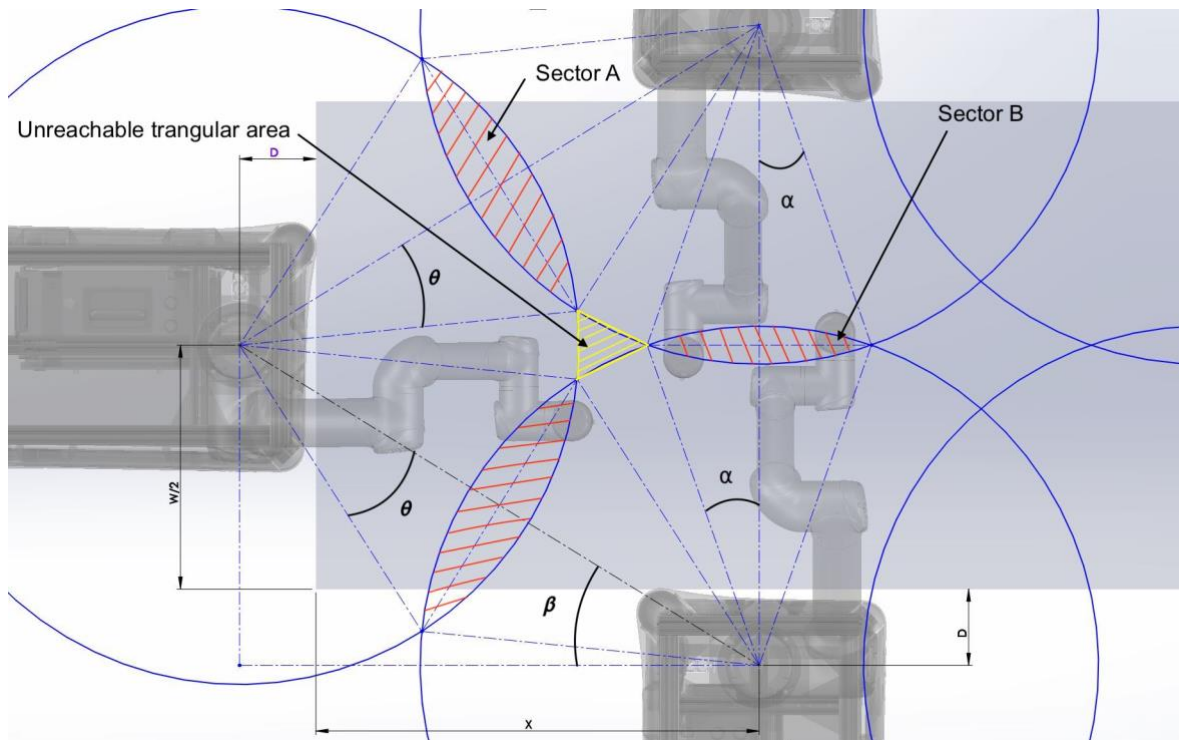


**Figure 13** Flowchart of TLBO optimization (Rao *et al.*, 2011, p. 305).



**Figure 14** Optimization ratio of overlap to the uncovered triangle area

The three circles form almost a triangular bound at the center, and three sectors of overlaps are shown in **Figure 14** Optimization ratio of overlap to the uncovered triangle area. The objective function was formulated to find the difference between total overlap areas created by the first three circles, the circle's sector, and the triangle formed inside the three circles. When the total area of overlap and the area of the triangle is almost equal, it can be deduced as a balance between the overlap and the unreachable area. The minimization of the resulting value between the difference in overlap can be used as a model for the optimization fitness function. The variable that formulates the equation for the sector and the triangle are correlated; therefore, each value links simultaneously to the resulting overall value. When fed to the optimization algorithms, this equation minimizes the objection function and results in the possible position of the articulated manipulator distance  $X1$  and  $X5$ . In the second step, the yellow rhombus formed at the center shown in **Figure 14** Optimization ratio of overlap to the uncovered triangle area was compared to the corresponding green circular overlap but with the exception of only the overlap area which covered over the table. The fitness function was modeled based on these premises to utilize the minima function and deduce the value for  $X2$  and  $X3$ .



**Figure 15** Graphical evaluation

From **Figure 15** Graphical evaluation a geometrical approach can be used for evaluating the shaded region.

Mathematically,

The above figure can be formulated into a function as follows

$x_d$  = the horizontal distance from the edge of the table

width = width of the table

$D$  = represents the clearance 0.179 m required by the robot to approach any object

$r$  = radius of the articulated manipulator (arm reach)

$\alpha$ ,  $\beta$ ,  $\theta$  = corresponding angles in radians

$A$ ,  $B$  = assigned area for the sectors

*objective function = Area of overlap sector – Area of triangle*

$$\text{hypoteneous} = \sqrt{(x_d + D)^2 + (0.5 * \text{width} + D)^2}$$

$$\text{Area of sector A} = \frac{1}{2} r^2 (2\theta - \sin 2\theta)$$

$$\text{Area of sector B} = \frac{1}{2} r^2 (2\alpha - \sin 2\alpha)$$

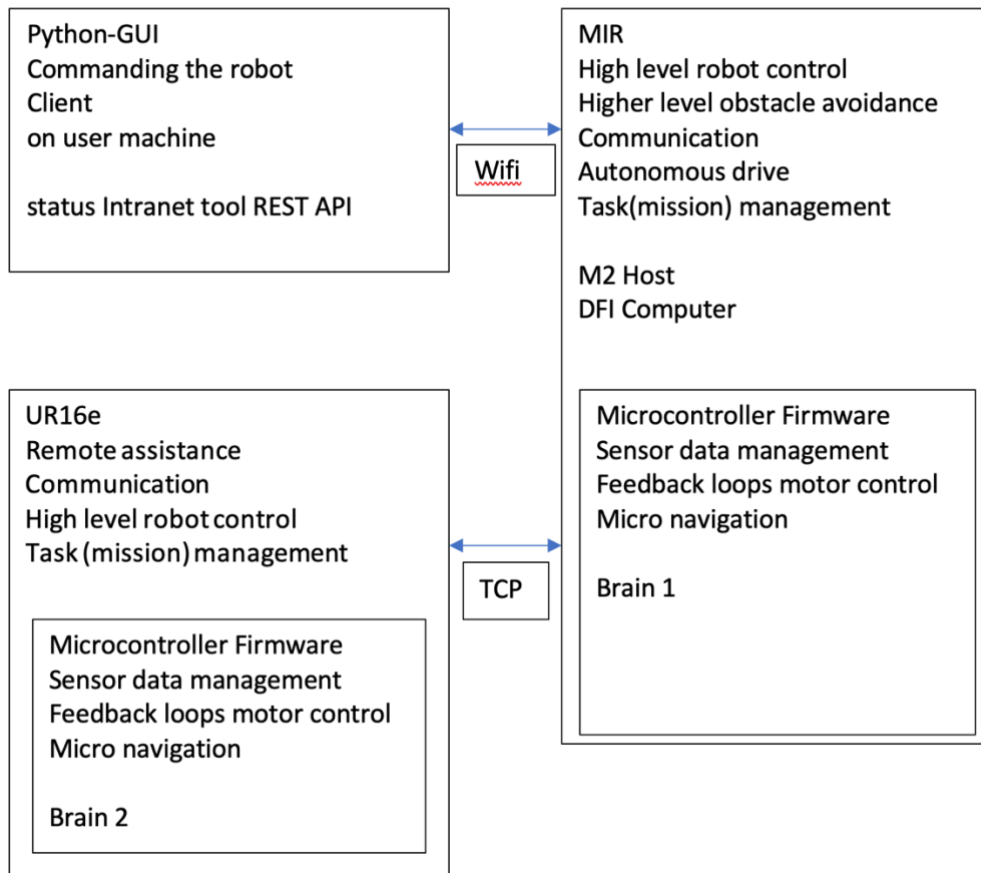
$$\text{Area of center triangle} = 0.5 * (x_d + D - r - (r * \sin \alpha)) * r * \sin(\beta - \theta)$$

$$\text{objective function} = 4 * \text{area of sector A} + 2 * \text{area of sector B} - \text{Area of center triangle}$$

Using this function for optimization and submitting the value of the table's length and the reach of the robot arm, positioning of the robot can be achieved along the length axis.

#### 2.4 Implementation and experiment

The autonomous mobile robot MIR200 is used as locomotive assistance to the articulated manipulator UR16e. The MIR has its own range of sensors to detect the surrounding dynamic environment. The UR robot is equally equipped with its force sensors to detect impact and reduce the motion to stop. The mobile robot has its own router to produce a WIFI network. The wireless network can be incorporated into another network, such as office networks or personal networks. The mobile robot provides an HTML based graphical user interface GUI for the end-user. Any device which can connect to its Wi-Fi and run an HTML page in the web browser can be used as a GUI device. A commercial tablet, phone, or even a computer can be used as a GUI device from this system. The device needs to be connected to system's WIFI, and a web browser is opened using mir.com. The HTML page provides a simple graphical interface to access different actions, missions, settings, configurations, terminals etc. Through this interface and an ethernet connection, the UR robot can be connected to the MIR network. The networking is a simple process of setting the UR robot to remote assistance mode and the MIR robot to accept the UR interface mode. The IP address of the UR needs to be manually set into the MIR system, and the system begins to assimilate the UR robot to the MIR system. A brief overview of the architecture of the robot can be seen in the **Figure 16** Architecture of the system below.

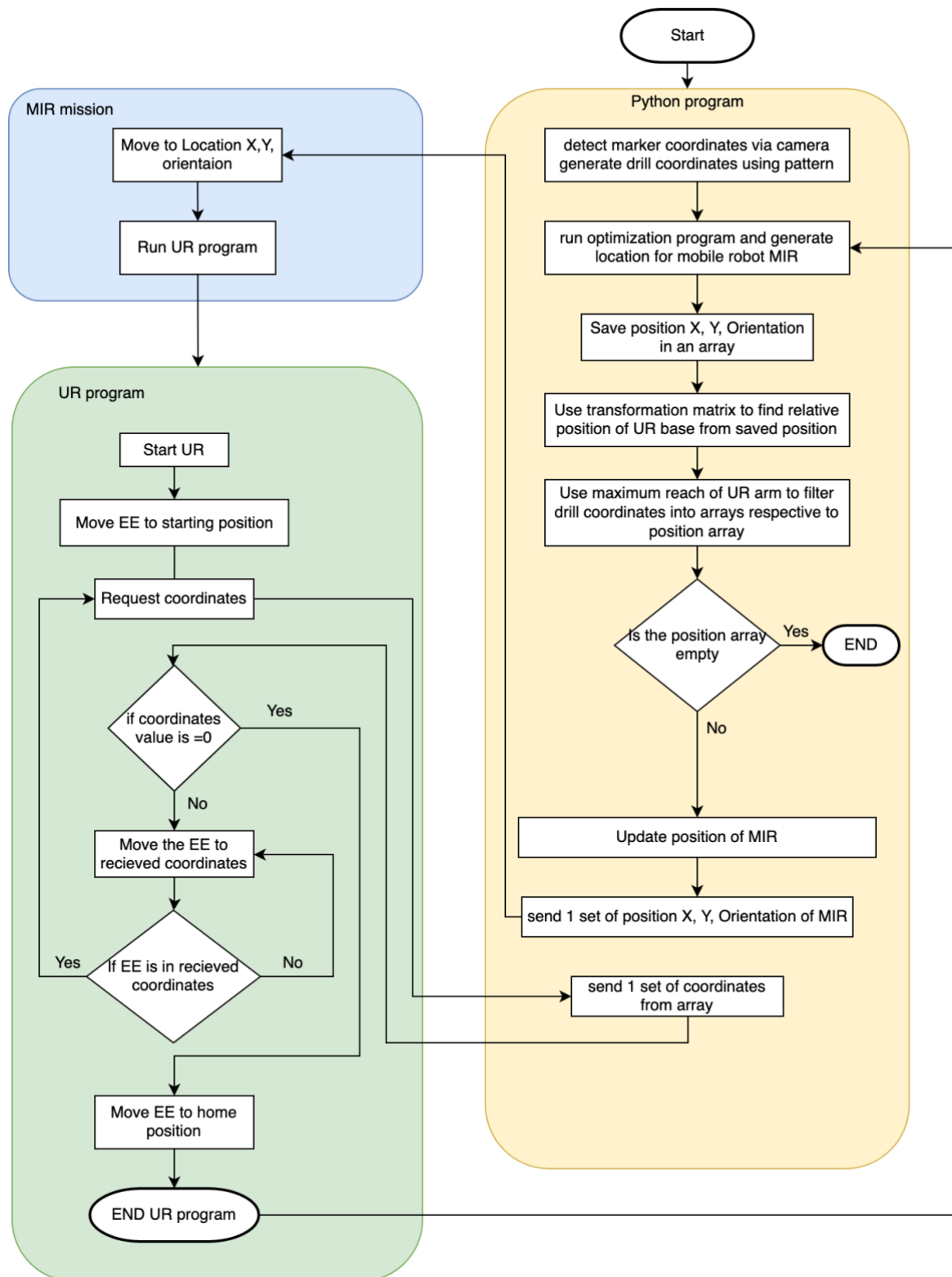


**Figure 16** Architecture of the system.

Depending upon the structure's physical assembly, it can be seen that the UR base's position is at specific values of  $X$ ,  $Y$ , and  $Z$  taken to the base frame of the MIR robot. Although the robot is taken as a single unit of a mobile articulated robot, the working of the robot is always functionalized into two parts, reach a location and perform the task.

In an example of this project, the robot needs to reach the table and perform the drilling. Similarly, the task is divided into two parts which different robots perform sequentially. The mobile robot is always the master as it controls the GUI interface, and therefore the UR robot is always the dominant robot. The power source, the dynamic detection, the mission control is all running on the mobile robot. Therefore it is imperative that the MIR robot can run its functionality without any errors. The sequencing of controls and activation of different robots depends upon an independent program running on python. The python program running on the computer connects to MIR Wi-Fi using a terminal or REST API. This includes the security bypass and the command flow to run the mobile robot program limited

to GUI access. The program can only be used to execute a mission with actions and change the parameters in the actions. Thus a mission parameter with an action move and a UR program was preprogrammed and saved to MIR's mission log in the GUI. The python program would run this specific mission in a loop command and change parameters during each loop to function like a sequential robot. The python program consisted of different functions with different objectives. Firstly, it needed to mathematically design a pattern of drill location depending upon the location of the marker position. Using this knowledge, the program needs to run a differential evolution optimization on an objective function characterized by the table variables' dimension. The values obtained from the optimization give the UR base frame's positioning relative to the local frame. The information is then calculated using a transformation matrix to change the location to MIR frame and relate the same information to the global frame. The position is finally determined, and the parameters are updated to the mission action under the MIR move action. Similarly, the control is sent to UR, and the UR program stored in the system runs. The UR program is built to request data in bit format; therefore, the python sends the coordinates in the form of bits to UR via MIR API. The UR receives the coordinates, and after moving the EE to the coordinate position, the UR conforms and requests another coordinate. This continues until all the categorized coordinates are computed and sent to the UR. When there are no more coordinates remaining, the system jumps back to the MIR, and the UR remains dormant, and MIR activates. It depends entirely on the python program to perform the task sequentially, and any errors during a sequence cannot be corrected. The program is an open loops system due to access limitations from the MIR. **Figure 17** Flowchart of the program between different platforms below shows the flowchart of the programs running in different units. In the future, it is possible to have an open-source MIR system with full access, which can be used to produce a more integrated feedback loop to control. A prospective method can be to record and log each task; however, the system would be far complex to read, analyze, save and run the data on every cycle before running a new task. It would still not be able to calibrate unforeseen errors such as power outages.

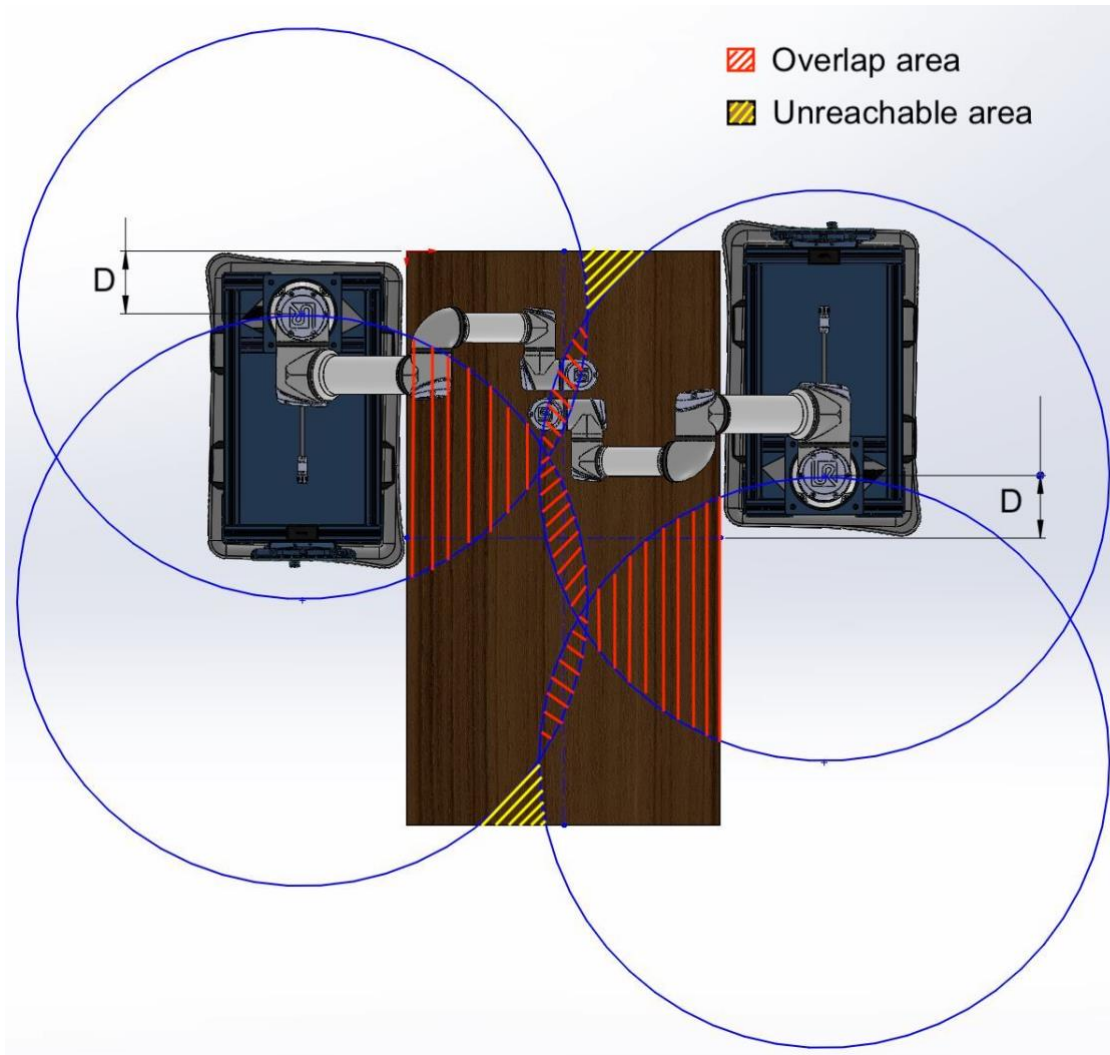


**Figure 17** Flowchart of the program between different platforms

### 3 RESULTS AND ANALYSIS

#### 3.1 First test

During the first test to handle the task, a controlled step method was adopted following the first approach. Depending upon the robot arm reach, the task space was sectioned into four quadrants. Inside a rectangular task space and all the drill position was then subjected to classification among the four quadrants. The robot was prepositioned accordingly around the edges of the task space and perform operations inside the quadrant. This approach demonstrated few limitations as the positioning of the robot was governed by the position of the drill coordinates. There was less coverage of the task space creating more unreachable areas between the quadrants. **Figure 18** Prepositioning of MIR and selective coverage workspace of the UR16 reach below shows the position and coverage of the workspace. The other disadvantage of this method was the unintuitive positioning of the robot for covering the task space. The program was specific to one particular dimension of the table and could not attribute to the table with different shapes. Although the program performed flawlessly in its final run but few issues were observed during the experiment. The Mobile robot was unable to reach the location, and the program crashed. The power shortage from the inverter and the emergency stop functionality made the program obsolete as there was no recovery mode, and all the information was gone. The only solution was to run the program all from the beginning. This was a significant drawback of the open-loop system.

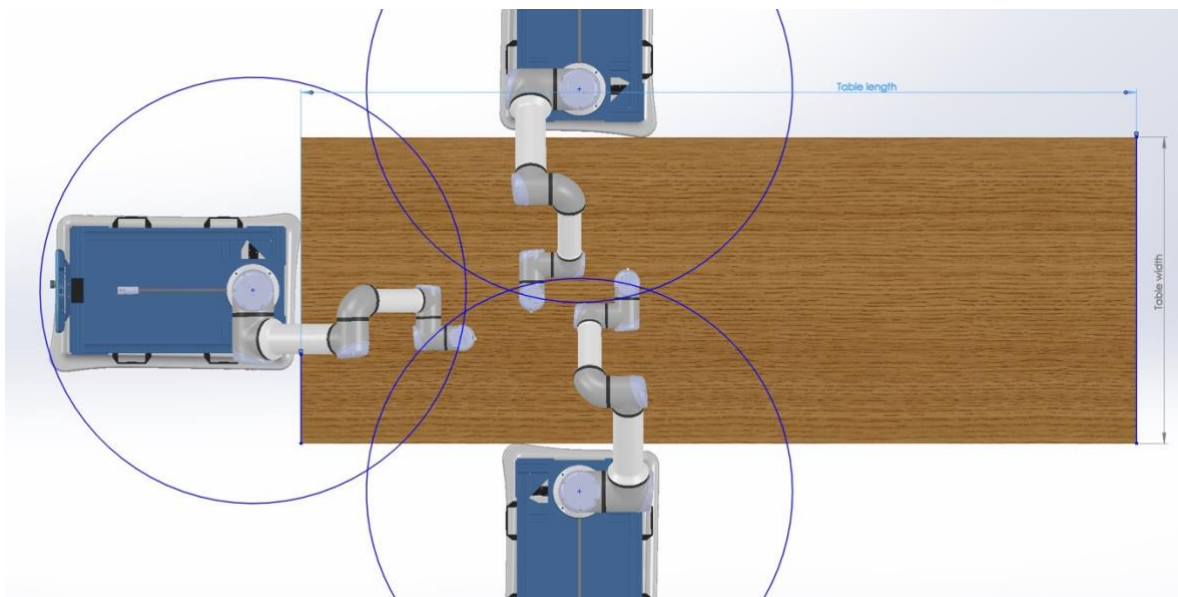


**Figure 18** Prepositioning of MIR and selective coverage workspace of the UR16 reach

### 3.2 Second test

The second test was done using an optimization algorithm. The MIR positioning would be dependent upon two factors, the overall overlapping coverage of the workspace and the minimization of the unreachable area. Depending upon these two premises, the geometrical expression presented via subtraction of overlap to the unreachable area when minimized can yield an expression for an object function. The expression would contain variables for the area of sectors of the circles in terms of the dimension of the table, and the unreachable area can be considered as an approximate triangle. The effective method was to reach the drill positions and minimize the unreachable task space. The positioning of the robot guaranteed coverage of a higher percentage of task space.

Furthermore, it revealed that the model of the robot's positioning would be irrelevant to the length of the table. Upon further examination, it was found that the model could be adapted to use with circular shapes of the table. **Figure 19** Positioning of MIR at circle center for maximum coverage below shows the adapted new positioning of MIR at the circle center to give maximum coverage of the workspace. Although this method made a more intuitive positioning of the robot, it was not a feedback loop program to handle all the errors mentioned above in the first results.



**Figure 19** Positioning of MIR at circle center for maximum coverage

The following results were obtained with changes in parameters of length and the robot arm reach. The optimization was done with bounds  $[0, 5]$ , mutation  $0.8$ , crossover  $0.7$ , population size  $10$ . The  $1^{st}$  and the last placement for the robot are always in the order of around the table length with a distance of  $0.179$  m away from edge of the table. The  $2^{nd}$  and the  $2^{nd}$  last placement are as a result of optimization, with the equal value in the sector area and the unreachable area between the three workspace. Similarly, the positions in between, are the optimization values as a deduction of balance between reachable and unreachable areas between the corresponding sectors and rhombus.

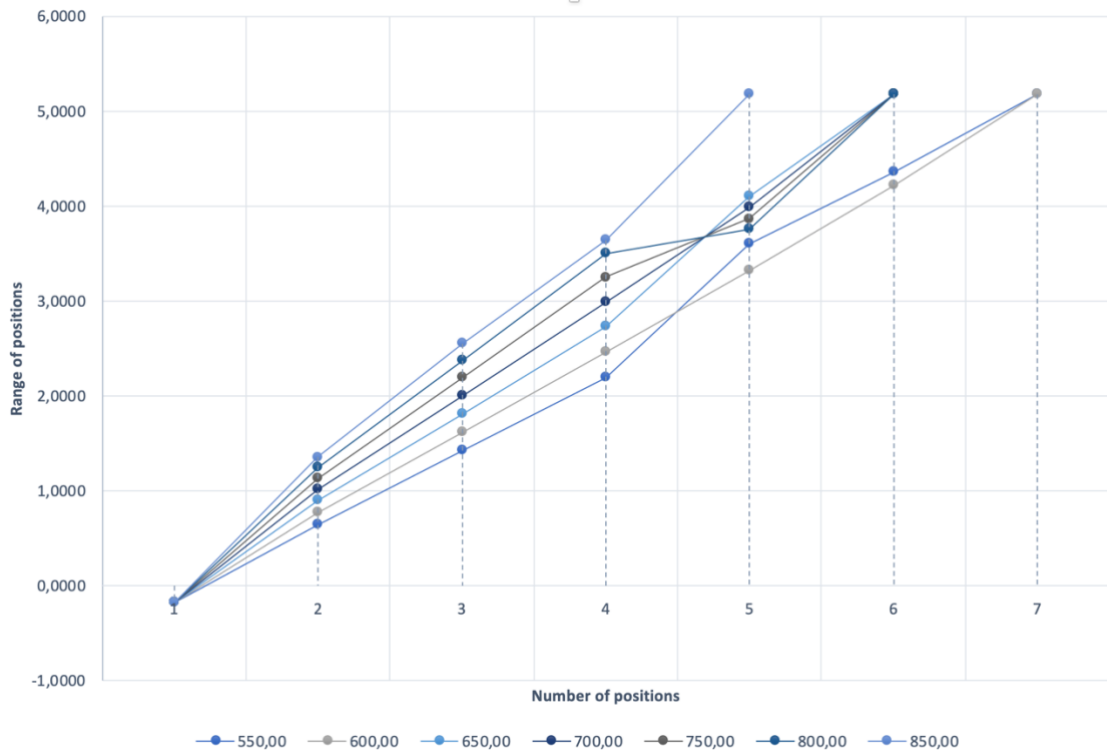
**Table 2** Differential evolution Optimization

length (m)	width (m)	reach (mm)	placement 1 (m)	placement 2 (m)	placement 3 (m)	placement 4 (m)	placement 5 (m)	placement 6 (m)	placement 7 (m)
5,00	1,10	550,00	-0,1792	0,6444	1,4222	2,2000	3,6092	4,3556	5,1792
5,00	1,10	600,00	-0,1792	0,7738	1,6224	2,4709	3,3194	4,2262	5,1792
5,00	1,10	650,00	-0,1792	0,8970	1,8163	2,7355	4,1030	5,1792	-
5,00	1,10	700,00	-0,1792	1,0159	2,0059	2,9958	3,9841	5,1792	-
5,00	1,10	750,00	-0,1792	1,1316	2,1923	3,2529	3,8684	5,1792	-
5,00	1,10	800,00	-0,1792	1,2450	2,3763	3,5077	3,7550	5,1792	-
5,00	1,10	850,00	-0,1792	1,3565	2,5505	3,6435	5,1792	-	-

**Table 3** Teaching learning based optimization

length (m)	width (m)	reach (mm)	placement 1 (m)	placement 2 (m)	placement 3 (m)	placement 4 (m)	placement 5 (m)	placement 6 (m)	placement 7 (m)
5,00	1,10	550,00	-0,1792	0,6444	1,4222	2,2000	3,6092	4,3556	5,1792
5,00	1,10	600,00	-0,1792	0,7738	1,6224	2,4709	3,3194	4,2262	5,1792
5,00	1,10	650,00	-0,1792	0,8970	1,8163	2,7355	4,1030	5,1792	-
5,00	1,10	700,00	-0,1792	1,0159	2,0059	2,9958	3,9841	5,1792	-
5,00	1,10	750,00	-0,1792	1,1316	2,1923	3,2529	3,8684	5,1792	-
5,00	1,10	800,00	-0,1792	1,2450	2,3763	3,5077	3,7550	5,1792	-
5,00	1,10	850,00	-0,1792	1,3565	2,5505	3,6435	5,1792	-	-

When the equations were simulated with both optimization algorithms, they yielded the same results. As the objective function was a simple geometrical function is inherently made similar solution candidate and population size. **Figure 20** Range and number of positions below gives an overview of the robot's position in a table length of 5 m. The task space is well distributed with almost equidistance spacing between robot positioning when the robot reachability is taken to be 850 mm; on the other hand, when the reach is 600 mm, the task space is accounted with more excellent robot positioning.



**Figure 20** Range and number of positions

#### 4 CONCLUSION AND FUTURE WORK

The object function generated evaluates an effective method to produce a positioning value for the autonomous robot MIR200 regardless of the length of the table. Although prepositioning of the robot through intuition was found to have a lower number of positions to the optimization approach, it limited its program only to a specific dimension of the table. The optimization approach yielded more coverage of the task space and was found to be reliable to work with variable length dimensions of the table while the width constraints remained to less than twice the arms reach. In theory, the method can be adapted to divide the workspace of either circular or rectangular task space using the same algorithm. The model can also position multiple robots in a different position, except to define collision-free workspace for individual robots working in the overlap workspace.

The predictive trajectory planning based on algorithms has been proven to be effective and efficient in the range of small workspace. However, the limitation exists as numerous sensors need to be installed around the workspace along with high-speed networking throughout to send responses to the system wirelessly. The system can be built up to a fully integrated mobile articulated system that can navigate and utilize more dynamic trajectory planning

algorithms depending upon the terrain while simultaneously adapting to the workspace's relative motion.

In principle, the concept implemented in this system has been bounded by its limitation. MIR architecture's access limitation and its functionalities have centered the python program to build around it in a relatively open loop system. In the future, it could be possible to have an open-source platform from MIR to utilize its full potential and produce a simultaneous control flow program to guide MIR using a UR sensor as well as relative trajectory planning.

## LIST OF REFERENCES

- Biswas, J. and Veloso, M. (2012) 'Depth camera based indoor mobile robot localization and navigation', *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 1697–1702. doi: 10.1109/ICRA.2012.6224766.
- Bogue, R. (2016) 'Europe continues to lead the way in the collaborative robot business', *Industrial Robot*, 43(1), pp. 6–11. doi: 10.1108/IR-10-2015-0195.
- Craig, J. J. (2004) *Introduction to Robotics: Mechanics and Control*. Pearson.
- Diankov, R. (2010) *Automated Construction of Robotic Manipulation Programs*. Carnegie Mellon University. Available at: <https://www.proquest.com/docview/859875469>.
- Forstenhausler, M., Wetner, T. and Dietmayer, K. (2020) 'Optimized mobile robot positioning for better utilization of the workspace of an attached manipulator', *IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM*, 2020-July, pp. 2074–2079. doi: 10.1109/AIM43001.2020.9158922.
- Fragapane, G. *et al.* (2020) 'Increasing flexibility and productivity in Industry 4.0 production networks with autonomous mobile robots and smart intralogistics', *Annals of Operations Research*. doi: 10.1007/s10479-020-03526-7.
- Grand View Research (2021) *Autonomous Mobile Robots Market Size, Share & Trends Analysis Report*, viewed 3 January 2021, Available at: <https://www.grandviewresearch.com/industry-analysis/autonomous-mobile-robots-market>.
- GreyOrange (2020) *GrayOrange Mobile Robots*, viewed 3 January 2021, Available at: <https://www.greyorange.com/news/greyorange-launches-next-era-fulfillment-operating-system/>.
- Hillier, M. S. and Hillier, F. S. (2003) 'Conventional Optimization Techniques', in *Evolutionary Optimization*. Boston: Kluwer Academic Publishers, pp. 3–25. doi: 10.1007/0-306-48041-7\_1.
- Huang, Z. and Chen, Y. (2013) 'An Improved Differential Evolution Algorithm Based on Adaptive Parameter', *Journal of Control Science and Engineering*, 2013, pp. 1–5. doi: 10.1155/2013/462706.
- International Federation of Robotics, I. (2017) 'Executive Summary World Robotics 2017 Industrial Robots', *World Robotic Report*, pp. 15–24.
- Kalawoun, R., Lengagne, S. and Mezouar, Y. (2018) 'Optimal robot base placements for coverage tasks', *IEEE International Conference on Automation Science and Engineering*,

- 2018-Augus, pp. 235–240. doi: 10.1109/COASE.2018.8560402.
- Keating, R. (no date) *UR5 Inverse Kinematics*, Scribd, viewed 3 January 2021, Available at: <https://www.scribd.com/document/456876867/UR5-Inverse-Kinematics>.
- Kok, K. Y. and Rajendran, P. (2016) 'Differential-evolution control parameter optimization for unmanned aerial vehicle path planning', *PLoS ONE*, 11(3), pp. 1–12. doi: 10.1371/journal.pone.0150558.
- Kuka (2021) *KUKA Mobile Robots*, viewed 3 January 2021, Available at: <https://www.kuka.com/en-us/products/mobility/mobile-robot-systems>.
- Lynch, K. M. and Park, F. C. (2017) *Modern Robotics Mechanics, Planning, and Control*. Cambridge University Press. doi: 10.1017/9781316661239.
- Makhal, A. and Goins, A. K. (2018) 'Reuleaux: Robot base placement by reachability analysis', *Proceedings - 2nd IEEE International Conference on Robotic Computing, IRC 2018*, 2018-Janua, pp. 137–142. doi: 10.1109/IRC.2018.00028.
- MiR (2018) *MiR100/MiR200 Ur Interface Operating Guide, 2018*, viewed 3 January 2021, Available at: <https://www.mobile-industrial-robots.com/media/4623/mir100-200-ur-interface-operating-guide-20.pdf>.
- MiR (2019a) *Artificial Intelligence Drives Advances in Collaborative Mobile Robots*, viewed 3 January 2021, Available at: <https://www.mobile-industrial-robots.com/pl/insights/human-amr-collaboration/artificial-intelligence-drives-advances-in-collaborative-mobile-robots/>.
- MiR (2019b) *MiR Safety*. Available at: <https://www.mobile-industrial-robots.com/en/insights/amr-safety/safety/>.
- MiR (2019c) *Mobile Industrial Robots (MIR200)*.
- Pech, M. and Vrchota, J. (2020) 'Classification of small-and medium-sized enterprises based on the level of industry 4.0 implementation', *Applied Sciences (Switzerland)*, 10(15). doi: 10.3390/app1015150.
- Rao, R. V., Savsani, V. J. and Vakharia, D. P. (2011) 'Teaching-learning-based optimization: A novel method for constrained mechanical design optimization problems', *CAD Computer Aided Design*, 43(3), pp. 303–315. doi: 10.1016/j.cad.2010.12.015.
- Rao, R. V., Savsani, V. J. and Vakharia, D. P. (2012) 'Teaching-Learning-Based Optimization: An optimization method for continuous non-linear large scale problems', *Information Sciences*, 183(1), pp. 1–15. doi: 10.1016/j.ins.2011.08.006.
- Research and Markets Ltd (2020) *Global Collaborative Robot (Cobot) Market: Focus on*

- Payload, Application Sales Channel, Component, and Industry - Analysis & Forecast, 2020–2025*, viewed 3 January 2021, Available at: [https://www.researchandmarkets.com/reports/5165436/global-collaborative-robot-cobot-market-focus?utm\\_source=BW&utm\\_medium=PressRelease&utm\\_code=4qsw86&utm\\_campaign=1455374+-+Global+Collaborative+Robot+Market+\(2020+to+2025\)+-+Focus+on+Payload%2C+Applicatio](https://www.researchandmarkets.com/reports/5165436/global-collaborative-robot-cobot-market-focus?utm_source=BW&utm_medium=PressRelease&utm_code=4qsw86&utm_campaign=1455374+-+Global+Collaborative+Robot+Market+(2020+to+2025)+-+Focus+on+Payload%2C+Applicatio).
- Rodríguez, N. *et al.* (2018) 'Optimization algorithms combining (meta)heuristics and mathematical programming and its application in engineering', *Mathematical Problems in Engineering*, 2018. doi: 10.1155/2018/3967457.
- Rubio, F., Valero, F. and Llopis-Albert, C. (2019) 'A review of mobile robots: Concepts, methods, theoretical framework, and applications', *International Journal of Advanced Robotic Systems*, 16(2), pp. 1–22. doi: 10.1177/1729881419839596.
- Schrimpf, J. (2013) *Sensor-based Real-time Control of Industrial Robots*. Available at: <http://hdl.handle.net/11250/261111>.
- Siderska, J. (2020) 'Robotic Process Automation-a driver of digital transformation?', *Engineering Management in Production and Services*, 12(2), pp. 21–31. doi: 10.2478/emj-2020-0009.
- Storn, R. and Price, K. (1997) 'Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces', *Journal of Global Optimization*, 11, pp. 341–359. doi: 10.1023/A:1008202821328.
- Unger, H., Markert, T. and Müller, E. (2018) 'Evaluation of use cases of autonomous mobile robots in factory environments', *Procedia Manufacturing*, 17, pp. 254–261. doi: 10.1016/j.promfg.2018.10.044.
- Universal Robots (2021a) *Cobot UR16e*. viewed 3 January 2021, Available at: <https://www.universal-robots.com/products/ur16-robot/>.
- Universal Robots (2021b) *UR16e D-H Parameter table*. viewed 3 January 2021, Available at: <https://www.universal-robots.com/articles/ur/application-installation/dh-parameters-for-calculations-of-kinematics-and-dynamics/>.
- Vahrenkamp, N. and Asfour, T. (2015) 'Representing the robot's workspace through constrained manipulability analysis', *Autonomous Robots*, 38(1), pp. 17–30. doi: 10.1007/s10514-014-9394-z.
- Vahrenkamp, N., Asfour, T. and Dillmann, R. (2013) 'Robot placement based on reachability

- inversion', *Proceedings - IEEE International Conference on Robotics and Automation*, (2), pp. 1970–1975. doi: 10.1109/ICRA.2013.6630839.
- Weyrer, M., Brandstötter, M. and Husty, M. (2019) 'Singularity Avoidance Control of a Non-Holonomic Mobile Manipulator for Intuitive Hand Guidance', *Robotics*, 8(1), p. 14. doi: 10.3390/robotics8010014.
- Wisskirchen, G. *et al.* (2017) 'Artificial Intelligence and Robotics and Their Impact on the Workplace', *IBA Global Employment Institute*, (April), p. 120.
- Xiao, N. (2009) 'Evolutionary Algorithms', *International Encyclopedia of Human Geography*, pp. 660–665. doi: 10.1016/B978-008044910-4.00525-3.
- Yang, X.-S. (2011) 'Metaheuristic Optimization: Algorithm Analysis and Open Problems', in, pp. 21–32. doi: 10.1007/978-3-642-20662-7\_2.
- El Zaatari, S. *et al.* (2019) 'Cobot programming for collaborative industrial tasks: An overview', *Robotics and Autonomous Systems*, 116, pp. 162–180. doi: 10.1016/j.robot.2019.03.003.
- Zacharias, F., Borst, C. and Hirzinger, G. (2007) 'Capturing robot workspace structure: Representing robot capabilities', *IEEE International Conference on Intelligent Robots and Systems*, pp. 3229–3236. doi: 10.1109/IROS.2007.4399105.

