THE INVESTIGATION OF TRAJECTORY GENERATION OF A CURVED WALL BRICK LAYING USING 5 DOF MOBILE ROBOT

MOHAMMED AHMED SALEM

BACHELOR OF ELECTRICAL ENGINEERING
(MECHATRONICS ENGINEERING WITH HONOURS)
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

MOHAMMED AHMED SALEM

2017

SUPERVISOR'S ENDORSEMENT

I hereby declare that I have read through this report "THE INVESTIGATION OF TRAJECTORY GENERATION OF A CURVED WALL BRICK LAYING USING 5 DOF MOBILE ROBOT" and found that it has comply the partial fulfillment for awarding the degree of Bachelor of Electrical Engineering (Mechatronics)

Signature: ………………………………………….

Supervisor's Name: PROFESOR MADYA DR MUHAMMAD FAHMI BIN MISKON

Date: 14 JUNE 2017

THE INVESTIGATION OF TRAJECTORY GENERATION OF A CURVED WALL BRICK LAYING USING 5 DOF MOBILE ROBOT

MOHAMMED AHMED SALEM

A report submitted in partial fulfilment of the requirements for the degree of Mechatronics Engineering with Honor

Faculty of Electrical Engineering
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2016 - 2017

STUDENT'S DECLERATION

I (Mohammed Ahmed Salem) declare that this report entitles "THE INVESTIGATION OF TRAJECTORY GENERATION OF A CURVED WALL BRICK LAYING USING 5 DOF MOBILE ROBOT" is the result of my research except as cited in the references. The report has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature:

…………………………………………

Name: MOHAMMED AHMED SALEM

Date: 14 JUNE 2017

To my beloved mother" AMAL" and father "AHMED"

ACKNOWLEDGEMENT

Firstly, I am grateful that **PROFESOR MADYA DR MUHAMMAD FAHMI BIN MISKON** is my supervisor for my Final Year Project. His guidance throughout the period of the final year project is very appreciated. His patience, motivation, and knowledge guide me to the right track of conducting this project. His guidance and corrections on writing the report, has helped me a lot in term of writing a good and qualified technical report. Also, I would like to thank my panels **DR. RAHIFA BINTI RANOM** and **CIK NUR MAISARAH BINTI MOHD SOBRAN**.

Secondly, I would like to thank my friends for their enormous assistance and discussion with me throughout this Final Year Project. Through discussion we were able to exchange and share ideas among us, so I gained a lot of meaningful knowledges about my project.

Lastly, I am very thankful to my family who give me the courage and confidence throughout the whole research of my Final Year Project. I keep receive all the support from them to do my best in achieving the success whole university life.

ABSTRACT

The masonry brick layer is related strongly with human health risks such as dermatitis, struck by falling object or musculoskeletal disorders. According to Map of the Construction Industry Amsterdam, Netherlands, 38% of bricklayers reported musculoskeletal disorders. In this project the main problem is on how to enable the user to describe the initial conditions of the desired curved wall brick layering (The coordinate of the first, middle and the last brick). To generate trajectory to obtain the profile of the curved wall brick positioning using 5 degree of freedom manipulator (KUKA youBot). The objectives of this project are, firstly to analyze the problem of curved wall brick laying using a five degree of freedom manipulator (KUKA YouBot), secondly is to design a trajectory profile relative to the curved wall brick laying and lastly to evaluate the accuracy of the brick positioning using KUKA YouBot utilizing parabolic polynomial equation. The parabolic polynomial equation is used, due to its simplicity, smooth motion and high accuracy. To achieve the objectives, trajectory is generated by using the parabolic polynomial equation. In order to validate the curved brick laying process, a simulation using VREP simulator is used to get the Cartesian axis (x, y, z) of the brick positioning. The parabolic equation is written in term of a code in MATLAB software to prove it is ability to produce any type of desired curve. The curves obtained from the parabolic equation are compared in term of the equation used, the arc length of the curve and the estimated bricks number required to complete the brick laying process. The result of the project is the curved wall brick laying obtained from the V-REP simulator with minimum accuracy of 77.53571% and Root Mean Square error (RMSe) of 0.02190791. In conclusion, the trajectory profile relative to the curved wall brick laying is designed with the use of the parabolic polynomial equation. The future work of this project is to increase the curve layers (starting from the second layer) as this project focuses on how to build any desired curve for the first layer. Moreover, this project is done in the V-REP simulator, so testing the experiment in the real (lab or physical) environment is highly recommended, especially to analyze the accuracy of every constructed curved wall.

ABSTRAK

Lapisan batu bata berkait kuat dengan risiko kesihatan manusia seperti dermatitis, dipanah objek atau gangguan otot jatuh. Menurut Peta Pembinaan Industri Amsterdam, Netherlands, 38% daripada penurap bata dilaporkan mengalami gangguan muskuloskeletal. Dalam projek ini masalah utama ialah bagaimana untuk membolehkan pengguna untuk menggambarkan keadaan awal melengkung bata lapisan dinding yang dikehendaki (Koordinat pertama, tengah dan bata yang lalu) dan untuk menjana trajektori untuk mendapatkan profil kedudukan bata dinding melengkung menggunakan 5 darjah kebebasan pemboleh ubaha (KUKA youBot). Objektif projek ini adalah, untuk menganalisis masalah dinding bata melengkung dengan meletakkan dan menggunakan lima darjah kebebasan pemboleh ubah (KUKA YouBot), kedua adalah untuk mereka membina profil trajektori berbanding dengan melengkung bata dinding memasang dan seterusnya, untuk menilai ketepatan kedudukan bata menggunakan KUKA YouBot menggunakan persamaan polinomial parabola. Persamaan polinomial parabolic digunakan, kerana kesederhanaan, gerakan yang lancar dan ketepatan yang tinggi. Bagi mencapai objektif, trajektori yang dihasilkan dengan menggunakan persamaan polinomial parabolic. Dalam usaha untuk mengesahkan proses bata meletakkan melengkung, simulasi menggunakan VREP simulator digunakan untuk mendapatkan paksi Cartesian (x, y, z) bagi kedudukan bata. Persamaan parabola ditulis dalam perisian MATLAB untuk membuktikan ia adalah keupayaan untuk menghasilkan apa-apa jenis lengkungan yang dikehendaki. Hasil daripada projek ini adalah bata dinding yang melengkung diletakkan diperolehi daripada simulator V-REP dengan ketepatan minimum 77,53571% dan kesilapan Root Mean Square (RMSe) daripada ,02190791. Kesimpulannya, profil trajektori berbanding dengan lengkungan bata dindingyang diletakkan ini direka dengan menggunakan persamaan polinomial parabola. Pada masa hadapan projek ini adalah untuk meningkatkan lapisan lengkungan (bermula dari lapisan kedua) sebagaimana projek ini memberi tumpuan kepada bagaimana untuk membina apa-apa keluk dikehendaki untuk lapisan pertama. Selain itu, projek ini dilakukan dalam simulator V-REP, ujian eksperimen dalam keadaan nyata (makmal atau fizikal) Persekitaran amat dititik beratkan, terutamanya untuk menganalisis ketepatan setiap dinding melengkung yang dibina.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Motivation

Scientific literature suggests that masonry work is often related to human health and diseases. There are several health risks that a bricklayer may face during the brick layering which may include: dermatitis, struck by falling objects, eye injuries or musculoskeletal disorders [1].

In 1999, a study observed that 38% of the bricklayers in the Netherlands reported musculoskeletal disorders [2]. In addition to the health risks, the bricklayers are not expert, fast and precise enough in comparison with a bricklayer robot such as bricklaying robot manufactured by The Australian company FASTBRICK Robotics [3].

The studies show that the average bricklaying rate of one human is 140 bricks/hour [2], while the robot can lay 255 bricks per hour [2]. Moreover, several studies observed a high rate of inefficiency in the masonry industry of different countries. For example, studies reported inefficiencies at approximately 68%, 50%, 40%, 7%, and 2% for firms in Spain, Canada, Portugal, Greece, and China respectively [2].

Automation increases productivity and reduces the risk of work disorders in highly repetitive and physically strenuous tasks. Therefore, this project (the investigation of trajectory generation of a curved wall brick layer robot) is to provide an assistance to the curved wall bricklayers to overcome the risks and increase the efficiency rates. This research will provide as well the use of trajectory kinematics to achieve the time history of position, velocity and acceleration of the curved wall brick layer robot.

1.2 Problem Statement

In this project, the problem of trajectory generation in curved brick laying is on how to enable the user to describe the initial conditions of the desired curved wall brick layering (The coordinate of the first, middle and the last brick). Path planning is the points in which the curved wall brick layer manipulator should follow to reach the desired location.

In order to make the motion of the curved wall brick layer manipulator flexible for the user of the robot system, the user is required to describe the desired motion by non-complicated functions of space and time to specify the trajectory motion.

By creating a trajectory generation that allow the curved wall brick robot to use similar algorithms for curve bricklaying having different degree of curvature and different arc length, by changing the desired parameters such as position of the first and the end bricks, then let curved wall brick laying robot decide the details of the complicated trajectory generation of space and time.

The user might want to specify the desired destination position and orientation of the end-effector for the brick layering process and leave it to the KUKA robot to decide on the exact shape of the path to reach to the desired position, the duration, the velocity profile. We also are concerned with how we can simulate these trajectories and represent them in the computer after they have been planned using V-rep simulation software.

By using KUKA manipulator as a brick layer, the user will have an advantage of not doing the complicated trajectory calculations but the accuracy of the brick positioning may be decrease because of the automatically control of the inverse kinematics by the KUKA robot arm, and the singularities that may happen. This problem may happen due to the performances of KUKA arm's angular velocities of the joints with differences distance between robot and the brick at the desired placing coordinate. Moreover, some desired positions are not reachable by the 5 degree of freedom manipulator due to some geometric problems of the path in Cartesian space or the limitation of the workspace of the manipulator.

Based on this problem described, the project need to address a method on how to achieve any type of desired curve. The proposed method is to use the parabolic polynomial to generate the desired trajectory and observe how it responds.

1.3 Objectives

The objectives of this research focused on the points that are related to the trajectory generation of the curved brick layer process, these points are as follows:

1. To analyze the problem of curved wall brick laying using a five degree of freedom manipulator (KUKA YouBot).

2. To design a trajectory profile relative to the curved wall brick laying using parabolic polynomial equation.

3. To evaluate the accuracy of the brick positioning using KUKA YouBot.

1.4 Scope of the project

The scope of this research focused on several points that are related to the trajectory generation of the curved brick layer process, these points are summarized as follows:

1. Study on the trajectory generation methods in order to choose the smooth and accurate method in generating the position, velocity and acceleration profiles.
2. Use V-REP simulator to generate trajectory planning profiles by using parabolic polynomial equations.
3. The study of the curved brick layering process is developed in a controlled environment with an assumption of no external factors need to be consider such as obstacles.
4. The achievement of any curved wall brick laying of the first layer.
5. Use a five degree of freedom (5 DOF) manipulator (KUKA) to apply the curve brick layering process.
6. To find error and accuracy between desired coordinate (x_d, y_d, z_d) and actual coordinate (x_a, y_a, z_a) by using methods of analysis.
7. The position and orientation of the robot while pick and place the brick using Cartesian space trajectory generation.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

The standing chapter is intended to summarize and present the prior research works from open literature review which are relevant to the curved wall brick laying. However, the path description and trajectory generation of the curved wall brick laying robot are the most main points taken under considerations in order to identify the most important difference between the path description and trajectory generation. In line with making potential effort to designate the most significant points as general review, the following subtitles such as Analyzing the information related to trajectory generation of the bricklaying robot (path planning and trajectory generation of the curved wall brick layer robot, basic of trajectory generation (Cartesian-Space Scheme, joint-space scheme), Geometric problems with Cartesian paths and the Process of Bricklaying), synthesizing the information gathered (the performance matrix, existing brick layer robot's compression and conclusion on the limitation of the existing design) and evaluating the information by selecting the suitable method in term of curved wall brick laying process.

2.2 Path planning and trajectory generation of the curved wall brick layer robot

Generally, pure geometric description of motion is provided by the path. In conjunction of the obstacle avoidance or brick layering in the curved wall with regards to path is usually planned globally (path planning). Furthermore, trajectory is a path with considering of both velocities and accelerations in each point in the space in which brick laying robot should follow. Table 2.1 summarizes the comparison between the path planning and trajectory generating, while Figure 2.1 shows the General block diagram. as shown below. In Figure 2.1 the relation of the general block diagram with the brick layer robot is explained in more details as follows:

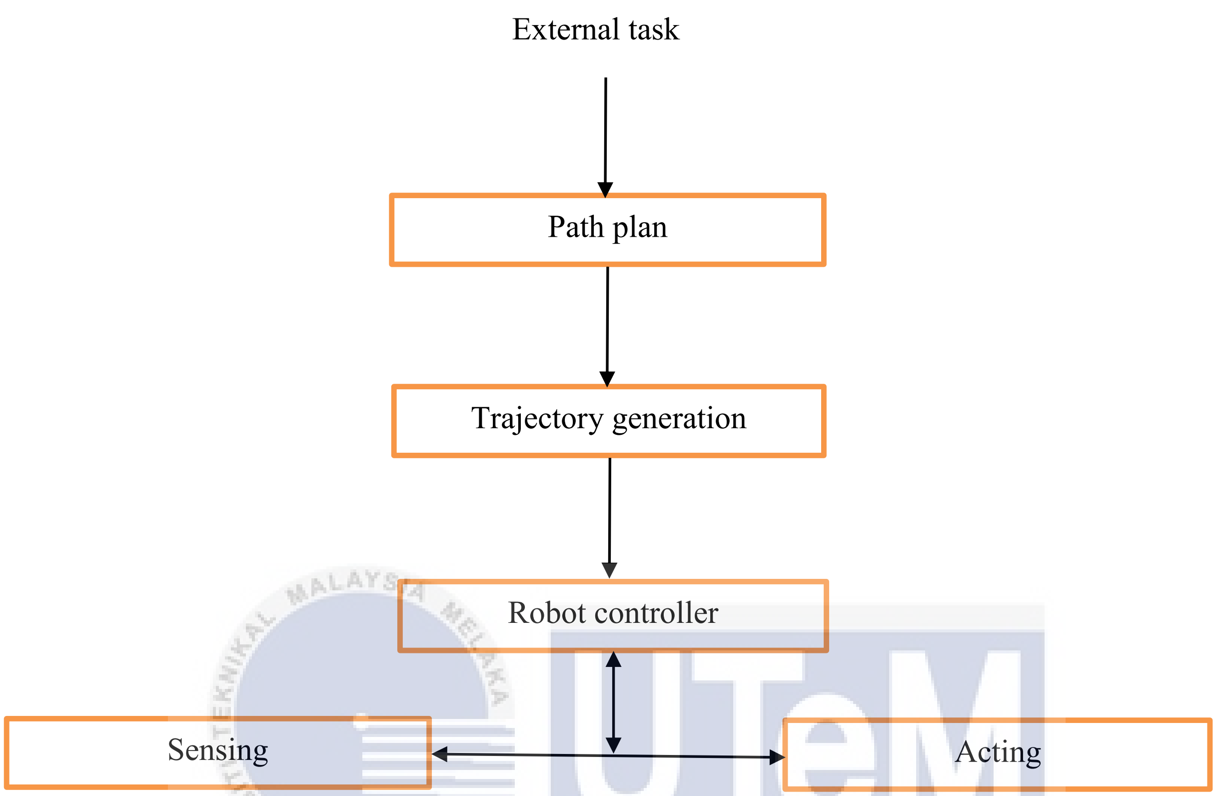

Figure 2.1: General block diagram

Where:

External task stands to lay bricks in the curve wall.

Path plan stands for desired curved path of brick laying process.

Trajectory generation considers of velocities and accelerations in each point in the space in which brick layering robot should follow.

Robot controller stands for using the KUKA robot controller.

Sensing stands for vision sensor.

Acting stands for putting the brick at the desired position by generating a time-based control sequence.

In this project, the consideration is focused on the trajectory generation only.

Table 2.1: Path planning (global) vs trajectory generating (local)[4]

Path planning (global)	Trajectory generating (local)
The (geometric) path is a sequence of Waypoints defining the trajectory coarsely. Issues solved at this level: obstacle avoidance, brick layering in the curved wall.	The path provided by path planning constitutes the input to the trajectory generator. Trajectory generator approximates the desired waypoints by the use of the polynomial functions and generates a time-based controlled sequence moving the end effector of the manipulator from the based point to the desired point.

In order to complete the curved brick laying process, the path planning and trajectory generations are integrated together. The connection between path planning and trajectory generation is shown through the block diagram in. Figure 2.2.

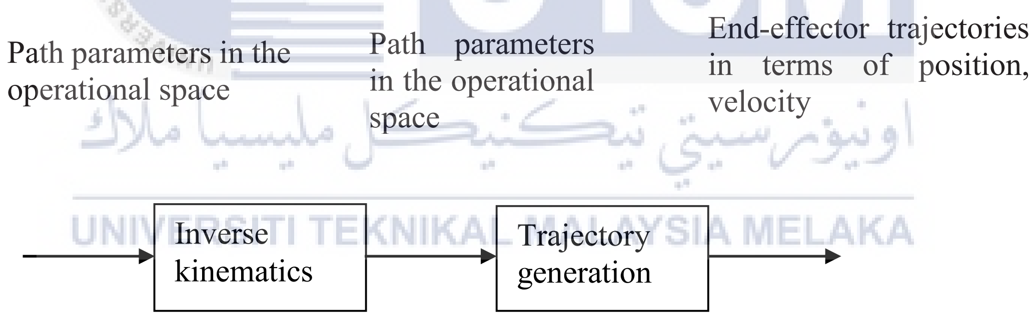

Figure 2.2: Functional block diagram of the brick layer robot[4]

Figure 2.2 illustrates the block diagram that the path parameters in the operational space are the initial and final end-effector location, the inverse kinematics is to calculate the required angles need to be rotated which is the inputs to the trajectory generation to specify the position, velocity and the acceleration of the end effector of the manipulator. In addition, Figure 2.3 shows the process of moving the end effector of the manipulator from an initial position to a desired final position.

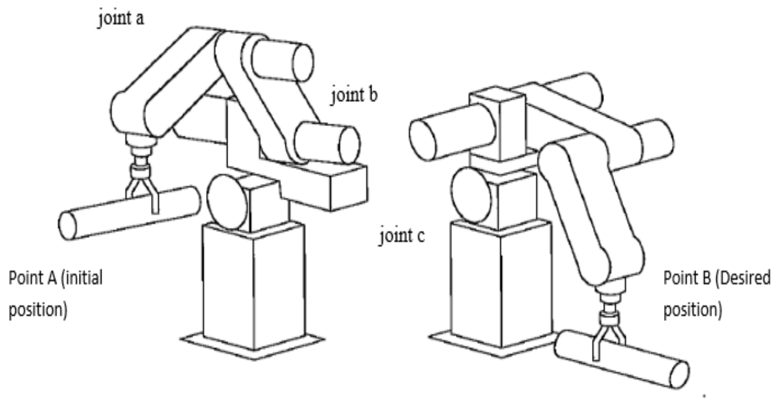

Figure 2.3: Moving the manipulator from an initial position to final position [4]

The most functional processes in Figure 2.3 are explained and elaborated as follows:

The path parameters in the operational space process is the initial and desired locations (Point A and B), while Inverse kinematics process is to calculate the angles rotation required at joint a, b and c, however trajectory parameters in the joint space process is to calculate the angles and trajectory generation process is to obtain the desired end-effector's position (Point B axis (X, Y and Z)), velocity and its acceleration.

2.3 Basic of Trajectory Generation

Trajectory is basically identified with velocity, acceleration and position for each joint (degree of freedom). The trajectory motion incorporates the human-interface in term of determination through space [4].

The most straightforward approach to portray the robot movement for client of a robot system is that the human client ought to be required to compose uncomplicated function (inputs) with a specific end goal to decide the task of the robot.

At the same time the ability of deciding Trajectories with straightforward portrayals must be in thought keeping in mind the end goal to get the desired movement. For example, in some cases the client (user) has to decide the wanted position and orientation of the end effector. In the meantime, the client abandons it to the manipulator to pick the wanted path of the trajectory generated to get alternate points of interest, for example, velocity and the acceleration profile [4].

Furthermore, identification of the trajectories by utilizing PC after the trajectories that have been arranged is the enormous concern, toward the end, obtaining the desired path from the generated trajectory is an issue that is accessible.

Widely recognized case, three parameters must be processed which are velocity, position and acceleration. Therefore, these trajectories are obtained at different sampling time [4]. The application of generating the desired trajectory path will be discussed in chapter four.

2.3.1 Joint-Space Schemes

Techniques for way generation in which the path are depicted as elements of joint angles. Each path point is normally shown as a desired position and orientation of the tool frame, {T}, with respect to the station frame, {S} [4]. By utilizing application in respect to inverse kinematics, every one of these through focuses is "exchanged" into an arrangement of coveted joint angles.

Along these lines, every portion is required a similar time for each joint so all joints can have the capacity to reach through the via point in the meantime, so the alluring Cartesian position of the end effector (tool frame) will accomplish at each via point. In addition, every joint has its own particular sought join point which is independent of alternate joints. Subsequently, join space plans accomplish the desired position and orientation through via points.

Moreover, the trajectory generation is considered to be ease to compute in joint space scheme compared with the Cartesian space because of the simple calculations required in the join space schemes. Furthermore, In the join space schemes there is fundamentally no issue with singularities of the system [4].

2.3.2 Cartesian-Space Schemes

Cartesian space is defined as the movement between the two points that is specified at all times and controllable. Cartesian space is easy method that can help to imagine the trajectory, although Cartesian space is not easy to ensure the singularity. Thus, in the joint space, the paths in order to ensure that via and desired points are achieved, even if these path points were determined by means of Cartesian frames [4].

In this part, methods of trajectory generation in the way that the path shapes are illustrated relative to functions that calculate Cartesian position and orientation such as functions of time. In this way, locative shape of the trajectory within path points can be specified.

The straight line is the most popular path shape and also circular and sinusoidal can be used. Every path point is commonly determined relative to a goal position and orientation of the end effector (tool frame) respected to the station frame. the time based functions splined to represent Cartesian variables.

There are various schemes that are used to generate Cartesian paths which have been suggested from the research and industrial robotics community. In this scheme, the same linear/parabolic spline will be used as it is used in the joint-space case such as Cartesian straight-line motion. Cartesian straight-line motion sometimes, we want to have ability to determine smoothly a spatial path that effects on the tip of the tool in order to move within space in a straight line. clearly, when we identify many carefully separated via points that is lying on a straight line, so the tool tip Wi-Fi seem to go after a straight line, without considering the option of smooth function that links the via points. In contrast, it is more comfortable when the tool goes after straight line paths between even greatly separated via points. So this mode is called Cartesian straight line motion [4].

Relative to straight lines, motions are defined as a subset of the common capability of Cartesian motion, in the way that arbitrary functions of Cartesian variables can be used to identify a path. In a system which allowed common Cartesian motion, such as path shapes as ellipses or sinusoids could be achieved.

2.4 Geometric problems with Cartesian paths

This subtitle presents two main geometric issues in Cartesian path which are, the intermediate points are Unreachable and base point and destination reachable in different solutions. These issues are discussed repressively as follows.

2.4.1 The intermediate points are Unreachable.

The final goal point and the based position of the end effector is in the workspace of the manipulator (A), even though it is quite possible that not all points on the straight line connecting the two points are in the workspace of the manipulator. Figure 2.4 illustrate the problem.

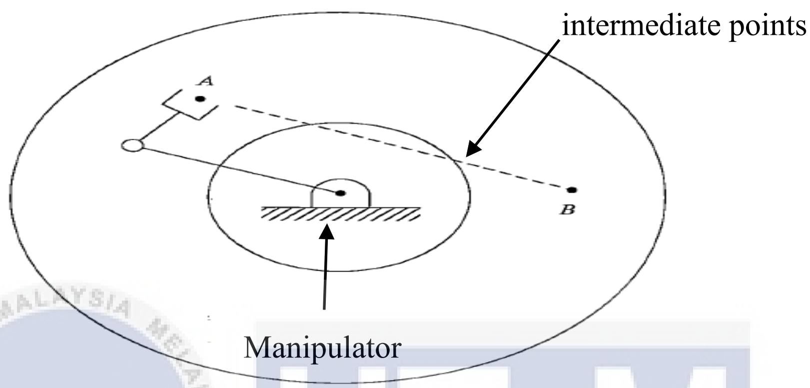

Figure 2.4: Planar two-link robot [4].

In this case the joint-space path would be performed easily, then the Cartesian straight-line path which would fail.

2.4.2 Base point and destination reachable in different solutions

A two-link with equivalent lengths has joint limits that keep the number of solutions with which it can achieve an accepted point in space. Particularly, an issue will appear if the desired point cannot be achieved in similar physical solution as the manipulator is at the based point. Figure 2.5 illustrate the problem.

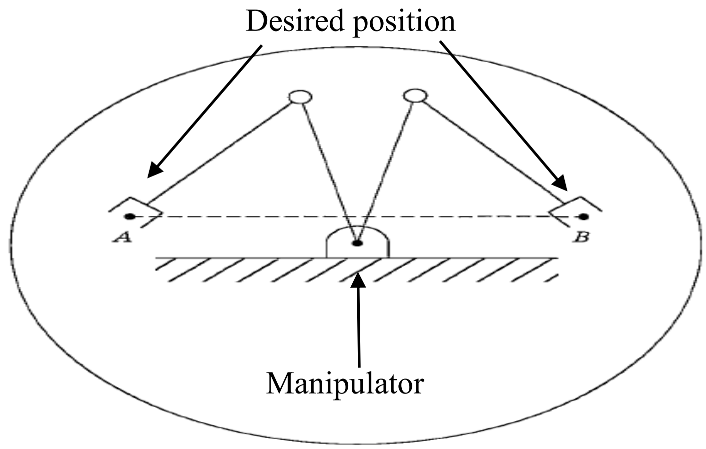

Figure 2.5: Planar two-link robot [4].

Most industrial manipulator supports both joint-space scheme and Cartesian-space scheme that, due to the troubles with Cartesian ways, joint-space scheme need to be utilized as the default, and Cartesian-space paths is utilized just when it is required by the application.

2.5 The Process of Bricklaying

In overall bricklaying process, the human and his tools are replaced by the robotic arm and end-effector, while the interface defining the outcome of the process is transferred from a drawing describing the design to information in the form of digital data. The human brick laying process is the human's interpretation of the drawing and the quality of his work that defines the outcome. In the robotic process, the critical interface switches to the data input. The input data defines the design and additionally controls the robot. A further change is that the design input of the robotic process can basically have an infinite information depth, while the design input of the human process is limited to the comprehension of the drawing. The performances of mason and robotic is differences, the inherent strength of the robot to precisely execute any number of commands, thus being able to follow different instructions and transform them into a physical output [5].

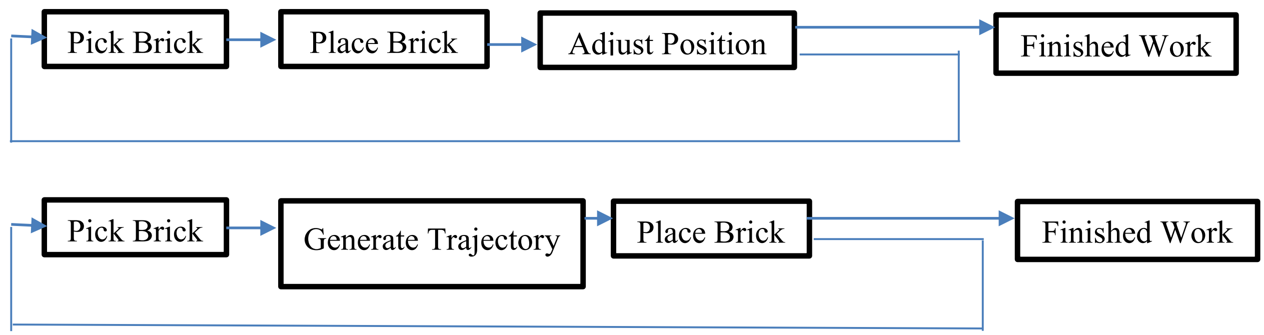

Figure 2.6: Manual process VS robotic process.

The human needs to make the brick in a correct positioning that achieved through visual alignment. Whereas, the robot can work without any additional guide system. In addition, the above mentioned aspects of the robotic set-up, the materials, and the tools used and how these influence the resulting robotically assembled brickwork are essential. On the other hand, the resulting architectural brickwork artefacts and their tectonic and structural qualities. The goal is thereby to identify the potential impact of an integrated design and robotic assembly process on brickwork [5].

2.6 State of Art in generation of curved wall brick laying robot

Many researches have carried out on how to generate trajectory motion of the curved brick layer process. According to the researches, the methods used to generate trajectories are focused on specific parameter performance such as generating motion with stability.

However, less methods which generate trajectory motion with considering a mechanical effect such as the structure of the model and the energy consumption during motion that can lead to improve some performance.

2.6.1 Performance matrix

In this section, a comparative study is carried on the possible and available trajectory generation methods to find the deference between the methods and the link with the bricklaying process. Table 2.2 demonstrates a comparison between three methods that are commonly used in generating curve trajectory profiles. The categories of this comparison are in terms of the complexity of the calculation and the accuracy of its stability. The ranking is from number 1 to number 5, where number 1 implies extremely low, number 2 implies low, number 3 implies medium, number 4 implies high and number 5 implies extremely high. The ranking of the accuracy is based on the friction force. Whereas complexity ranking is in term of difficulty of calculation and the time taken to finish the problem.

Table 2.2: Comparison between existed methods based on complexity and accuracy.

Criterion / Methods	Complexity in term of calculation	Accuracy in term of stability
Quintic polynomial[6].	4	5
Cubic spline [7].	3	4
parabolic trajectory [8]	2	4

Quintic polynomial is a high order polynomial (fifth order) used to generate trajectories smoothly [6]. The general form of the quintic equation is shown in equation 2.1.

$$y = a_{10} + a_{01}x + a_{11}x^2 + a_{21}x^3 + a_{31}x^4 + a_{41}x^5 \tag{2.1}$$

Where $a_{10}, a_{01}, a_{11}, a_{21}, a_{31}$ and a_{41} are constants. The general graph of the quintic equation is shown in figure 2.7.

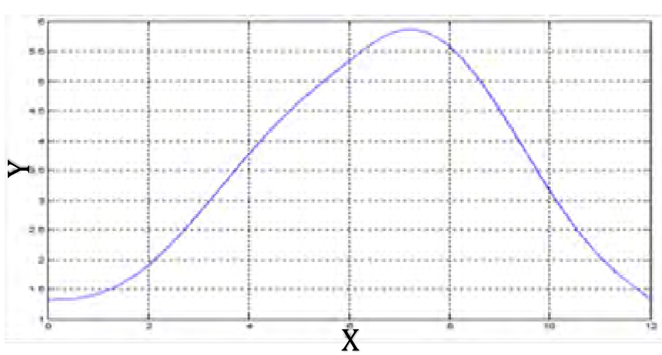

Figure 2.7: Quintic equation graph.

The quintic polynomial generates more than one curve. Moreover, the complexity on doing the quintic calculations is high compared to the other trajectory equations. Although the accuracy in term of stability is extremely high, this equation will not be used to do the brick laying process due to the generation of unwanted curves (generate more than one curve).

Cubic spline is a third order polynomial that Generate trajectory between two points [7]. The general form of the Cubic spline equation is shown in equation 2.2.

$$y = a_{20} + a_{02}x + a_{12}x^2 + a_{22}x^3 \quad (2.2)$$

Where a_{20}, a_{11}, a_{21}, a_{31} and a_{41} are constants. The general graph of the quintic equation is shown in figure 2.8.

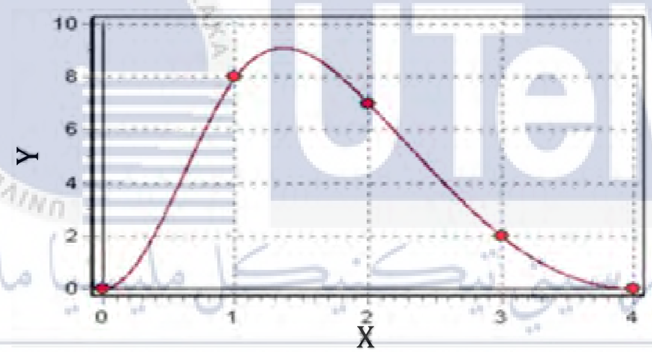

Figure 2.8: Cubic equation graph.

The cubic polynomial requires the user to identify the via points (the points between the initial and the final points), this may cause a problem to the user who wants to build curved wall without the knowledge of the via points values. Moreover, the complexity on doing the cubic calculations is high compared to the parabolic equation so this equation will not be used to do the brick laying process.

Parabolic polynomial is a Second order polynomial used to generate trajectories in curve structure [8]. The general form of the Parabolic polynomial equation is shown in equation 2.3.

$$y = y_0 + a(x - x_{initial})(x - x_{final}) \tag{2.3}$$

Where a is a constant that determine the maximum / minimum height. $x_{initial}$ and x_{final} are the x coordinates of the initial and the final points, y_0 is the initial y coordinate. The general graph of the quintic equation is shown in figure 2.9.

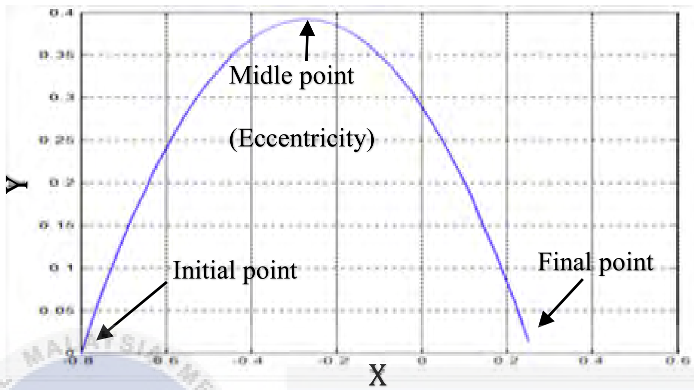

Figure 2.9: Parabolic equation graph.

The parabolic polynomial requires the user to identify three main points namely the initial, middle (the eccentricity of the parabola) and the final points coordinates. The parabolic polynomial generates only one curve. Moreover, the complexity on doing the parabolic calculations is low compared to the other trajectory equations. Although the accuracy in term of stability is lower than the quintic polynomial, this equation will be used to do the brick laying process.

2.6.2 Existing brick layer robot's comparison.

In this section, a comparative study is carried on the existed bricklaying robots. Table 2.3 shows the comparison of the existed bricklaying robots in term of the manufacturer, bricklaying speed, control algorithm and the cost for each robot.

Table 2.3: Comparison between existed bricklaying robots

Criterion \ Product	THE HADRIAN 105	SEMI-AUTOMATED MASON (SAM)
Manufacturer	Australian company namely (FASTBRICK ROBOTICS)	American company namely (CONSTRUCTION ROBOTICS)
Brick-Layering Speed	Top layering speed is 225 bricks per hour.	800 to 1200 bricks per day
Control Algorithm	instructions from 3D CAD software	'Construction Robotics' proprietary software Tablet app.
COST	Claimed to be cost efficient (Not out yet) [3].	$500,000 (US Dollar)

Table 2.3 demonstrate the comparison between the different existed designs namely (THE HADRIAN 105 and SEMI-AUTOMATED MASON (SAM)) against four criterions considered which consists of the manufacturer, the brick-layering speed, control algorithms, and the cost.

Both products have different manufacturer, but both of them are doing the same function which is brick-layering. From the aspect of the brick-laying speed.

The Handrian 105 provide high speed in compare with the Semi-automated mason. This feature illustrates one of the limitation of the semi-automated mason.

In term of achieving a desired wall brick-laying, both robots need a human mason to input the instruction and specific measurements to the robots through specified software, then the robot will build the specified walls, with the respective speed.

The working principles of the robots are somehow straight forward, the Handrian 105 eliminates the use of cement, but it replaces it with spreading adhesive to each brick, while the semi-automated mason is using cement as the mortar. the robots are not completely automated, a human mason needs to follow it and tidy up after it. The robots discussed are able to do the wall bricklaying process in high speed compared to human mason, but when it comes to intricate tasks such as corner or curved wall brick laying process, the robot faces challenges and it need a human mason to follow it and take out the intricate tasks.

2.6.3 Conclusion of the literature review

In this project, the focus is on how to solve the curved wall brick laying problem by creating a trajectory generation that allow the curved wall brick robot to use similar algorithms for curve bricklaying having different degree of curvature and different arc length, by changing the parameters desired such as the coordinates of the first and final bricks.

In conclusion, there are many methods of trajectory generation motion. The parabolic trajectory is the most suitable method in order to generate trajectory motion. Although it is not accurately excellent but it is the simplest and the most known method. Moreover, the parabolic polynomial is capable of generating a smooth and stable motion. Furthermore, the parabolic equation gives the user an advantage of laying the bricks in any curved wall structure. This property will be proved and discussed in chapter 4. In this project, parabolic polynomial will be used, which means that the trajectory motion will be generated based on the parameters required.

CHAPTER 3

METHODOLOGY

3.1 Introduction

This section is a discussion on the methodology of the project. The project will be achieved by using MATLAB software, Microsoft Excel and V-REP simulator. The simulation will be modeled by writing the equations of the parabolic polynomial as a code. Trajectory motion will be generated in term of Cartesian position.

Microsoft Excel is used to compare between the desired brick positions (x_d, y_d, z_d) out of the parabolic equation and the actual brick positioning (x_a, y_a, z_a) from the V-REP simulator. This compression is important to calculate the root mean square error (RMSE) and the accuracy of brick positioning in V-REP simulator.

MATLAB software is used for the curved brick analysis such as the arc length and the curvature of the curved wall theoretically.

This chapter will contain the following subtitles; Theoretical description of the proposed idea, Objectives of the simulation, Materials and equipment, Setup of the simulation, The type of data gathered, the type of data gathered, the data analyzing method and the procedures of conducting the simulation and getting the data.

3.2 Theoretical description of the proposed idea

In order to generate trajectory for curved brick layering motion profile starts from an initial position as shown in Figure 3.1 and stop at a desired position as shown in Figure 3.1 this project propose the using of the parabolic polynomial equation to generate the desired curve motion of the end effector.

The user is required to enter the desired initial conditions such as The coordinates of the first brick at the picking area, the gap between the picking bricks in X, Y and Z directions, the coordinates of the first desired brick position at the placing area, the coordinates of the maximum/minimum desired brick position at the placing area, the coordinates of the final desired brick position at the placing area, the gap between the bricks (also could be automatically calculated based on the dimensions of the brick) and the initial rotation of the brick.

KUKA manipulator is programmed using LUA (programming language) language to apply the parabolic trajectory in the curved brick laying process. The joint angles of the manipulator (θ1, θ2, θ3, θ4 and θ5) are calculated using the inverse kinematics module in the V-REP simulator. The position, velocity and acceleration profiles of every degree of freedom are obtained from the V-REP simulator. V-REP simulator is used because V-REP provides a suitable environment for testing, simulating and evaluating the curved brick laying process. Figure 3.20 shows the general flowchart of the project.

Figure 3.1: Curved bricklaying conceptual diagram

In this project, from the based point to the desired point there are no obstacles assumed to be found. Based on Figure 3.1 assume that the robot arm will pick and place the bricks from initial position to the desired position. In this project, parabolic polynomial trajectory is used.

In overall bricklaying process, the human arm is replaced by the robotic arm and the end-effector, in the robotic brick layering process, the path planning in Cartesian space of the desired motion profiles (position, velocity and acceleration) are the inputs to the robot. The robot is designed to obtain the suitable angles of the joint (five angles) by using the inverse kinematics algorithms.

In order to find the required data for the curved brick layering process the following points need to be conceder

3.2.1 Define the desired position of the bricks (x_p, y_p, z_p) At the picking area

$$p_p = (x_p, y_p, z_p)$$

$$x_p = (x_0 + x_{last1}) + x_{pickgap1}$$

$$y_p = (y_0 + y_{last1}) + y_{pickgap1}$$

$$z_p = z_0 + z_{pickgap1} \quad (3.1)$$

Where:

p_p: The position of the bricks at the piking area.

x_0, y_0 and z_0: The coordinates of the first brick (Get from the user).

x_{last1} and y_{last1}: The coordinates of the previous brick.

$x_{pickgap1}$: The distance between the centers of two bricks in X-axis.

$y_{pickgap1}$: The distance between the centers of two bricks in Y-axis.

$z_{pickgap1}$: The distance between the centers of two bricks in Z-axis.

Figure 3.2 shows the positioning of the bricks in respect with brick sizes at the picking area.

Figure 3.2: positioning of the bricks in respect with brick sizes

3.2.2 Define the desired position of the bricks (x, y, z) At the placing area

$$p = (x, y, z)$$

$$x = (x_{initial} + x_{last2}) + \mathrm{D}$$

$$y = y_{initial} + y_{from\ the\ parabolic\ equation}$$

$$z = z_{initial} + z_{pickgap2}$$

$$D = \sqrt{G^2 - H^2}$$

$$g = G - (\frac{1}{2}(B1) + \frac{1}{2}(B2)) \tag{3.2}$$

Where:

p: The position of the bricks at the placing area.

$x_{initial}$, $y_{initial}$ and $z_{initial}$: The coordinates of the first brick (Get from the user).

$x_{initial}$: The x-coordinate of previous brick.

D: The distance between the centers of two bricks in X-axis.

H: The distance between the centers of two bricks in Y-axis.

G: The desired distance between the centers of two bricks.

g: The desired gap between two bricks (Get from the user).

Figure 3.3 shows the positioning of the bricks in respect with brick sizes at the placing area.

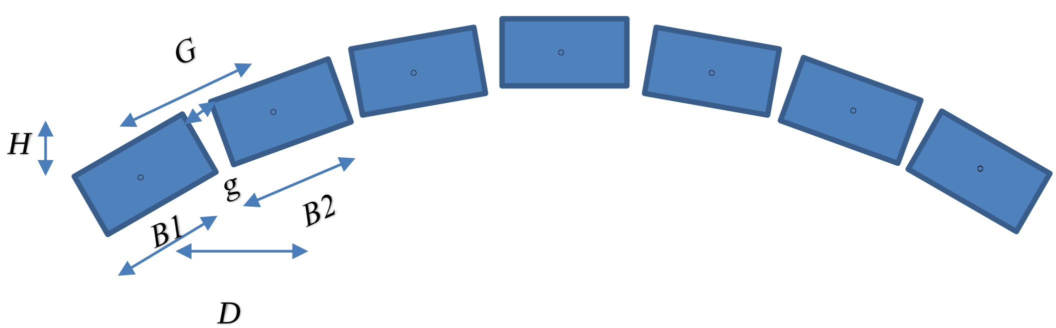

Figure 3.3: Bricks Positioning

3.2.3 Parabolic Polynomial

Parabolic polynomial equation is a second degree polynomial equation which is formulated as shown in Equation (3.3).

$$y = y_{initial} + a(x - x_{initial})(x - x_{final})$$

or

$$y = y_{initial} + ax^2 - ax_{initial} - ax_{final} + a(x_{initial})(x_{final})$$

$$x = (x_{initial} + x_{last}) + \mathrm{D}$$

$$a = \frac{y_{(max\backslash min)} - y_{initial}}{((x_{(max\backslash min)} - x_{initial})(x_{(max\backslash min)} - x_{final}))} \qquad (3.3)$$

Figure 3.4 shows the parabolic equation graph used to guide the manipulator in order to achieve the curved wall brick laying.

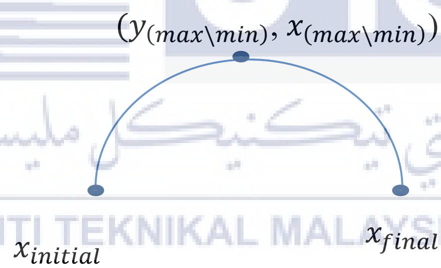

Figure 3.4: Parabolic equation graph

In this project MATLB is used to plot the desired parabolic equation based on the variables giving by the user. At the same time these equations are also written in form of a code in V-REP using LUA programming language to generate the trajectory of a curved brick laying process. This trajectory generation is the guidance of the KUKA manipulator to achieve the curve brick laying process.

3.2.4 Find the joint angles using inverse kinematics module.

The mathematical calculations of inverse kinematics are not obtained in this research, because this research is focusing on the trajectory generation only. However, we can use the simulation (V-REP) to obtain the joint angles by using the inverse kinematics module (IK module).

3.2.5 Find the joint velocities and acceleration.

The velocities and acceleration could be obtained by using either **Taylor's** differentiation theorem's shown in equation 3.4[4].

$$\dot{\theta} = \frac{\theta_f - \theta_0}{t_f - t_0}$$

$$\ddot{\theta} = \frac{\dot{\theta}_f - \dot{\theta}_0}{t_f - t_0} \quad (3.4)$$

Or using the inverse Jacobian method as shown in equation 3.5 [4].

$$\dot{\theta} = J(\theta)^{-1}.\dot{X}$$

$$\ddot{\theta} = J(\theta)^{-1}.(\ddot{X} - \left(\frac{d}{dt}J(\theta)\right).\dot{\theta}) \quad (3.5)$$

In this project the velocities and acceleration of each degree of freedom (joint) are obtained from the V-REP simulator.

3.3 Objectives of the V-REP simulation

The simulation is carried on to achieve the following objectives.

1 Proof that parabolic polynomial equation is suitable to generate trajectory for the curved brick layer motion.
2 To analyze the arc length and the estimated brick numbers of the curved wall.
3 To analyze the error and the accuracy of the bricks positioning.

3.4 Materials and equipment

In this project, the programs that are utilized to get the proposed aide are MATLAB, Microsoft excel and V-REP simulator. MATLAB is a high language and intelligent environment for numerical calculation, representation, and programming. Besides, analyzing data, creating calculations, and making models should be possible by utilizing MATLAB.

In addition, V-REP simulator is one of the software that provide data recording and representation so a substantial assortment of recordable information streams can show time based graphs, or can be joined to form x, y diagrams, furthermore, in order to generate the curved brick layering trajectory process, a five-degree of freedom manipulator is required.

3.4.1 KUKA YouBot workspace in physical world

KUKA Laboratories GmbH manufactured a five-degree of freedom manipulator called KUKA YouBot arm[9]. KUKA YouBot is an expandable, modular and open robot system that is developed for various research purposes. At the same time KKA YouBot is a mobile manipulator that consists of two main parts which are the KUKA arm and the KUKA platform.

The KUKA arm is a serial of chain consisting of five revolute joints called degree of freedom. The end effector of the KUKA arm is a two finger gripper with a 70 mm maximum opening.

Figure 3.5 shows the kinematics structure of the KUKA arm and the frames of KUKA arm. While Table 3.1 illustrate the maximum workspace of the manipulator. The KUKA youBot arm the dimensions and workspace are important parameters in order to specify the values of the joint angles by using the inverse kinematics technique.

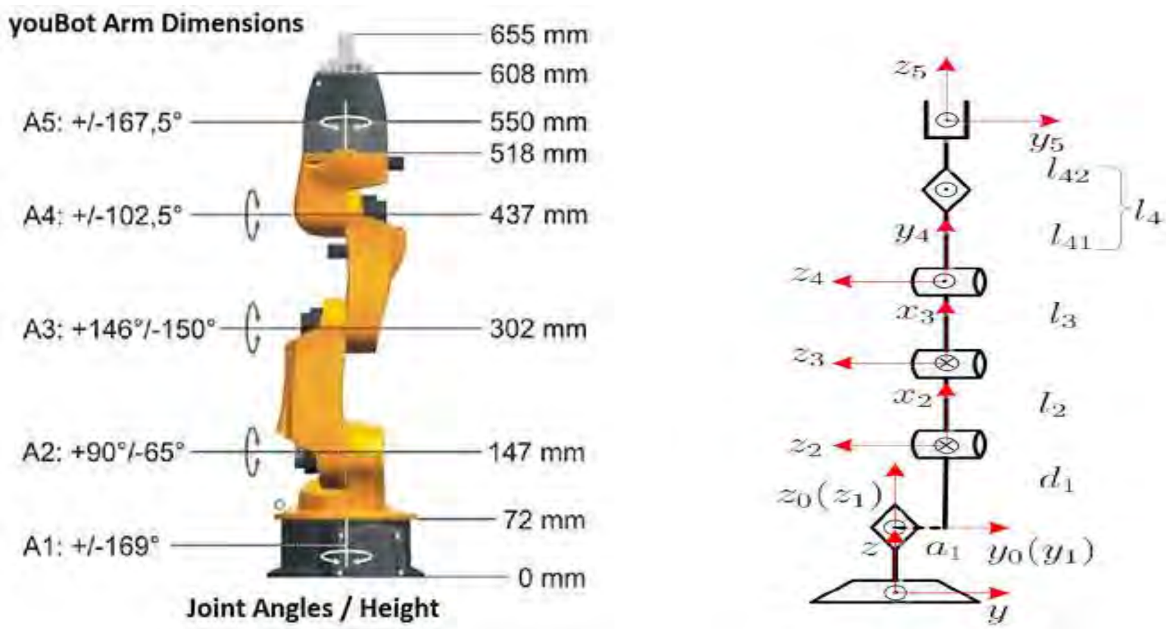

Figure 3.5: a) kinematics structure of the KUKA arm [14], b) The KUKA arm frames[11].

Figure 3.6 and Table 3.1 illustrate the maximum workspace of the KUKA arm manipulator

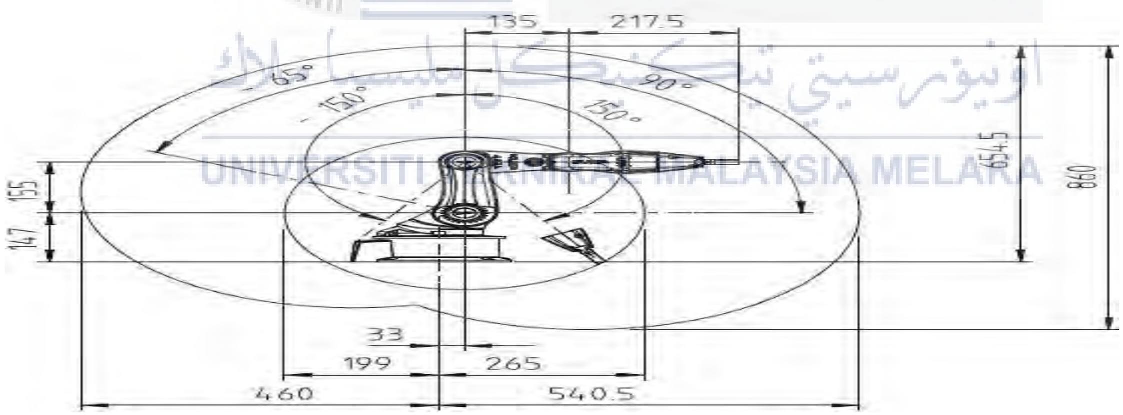

Figure 3.6: Maximum workspace of the KUKA youBot arm[14].

Table 3.1 is the derived from Figure 3.4. Table 3.4.1 illustrate the maximum and minimum workspace of the KUKA youBot.

Table 3.1: Maximum workspace of the KUKA youBot arm

Axis data	Range
Axis 1 (A1)	±169°
Axis 2 (A2)	+ 90°/– 65°
Axis 3 (A3)	+ 146°/– 151°
Axis 4 (A4)	±102°
Axis 5 (A5)	±167°
Gripper	Detachable, 2 fingers with 70 mm Gripper range.

The KUKA YouBot is an Omni directional mobile platform with four wheels. Table 3.2 illustrate the KUKA platform basic data.

Table 3.2: KUKA platform basic data.

DATA	SPECIFICATION
Material	Steel
Length	531 mm
Weight	20 kg
Wheel diameter	100 mm
Ground clearance	20 mm
Width	380 mm
Payload	20 kg
Height	140 mm

3.4.2 KUKA YouBot workspace in V-REP environment

In V-REP environment the workspace of the KUKA arm is not exactly the same as in the physical environment, there are some deference. Table 3.3 illustrate the maximum and minimum workspace of the KUKA youBot in V-REP environment.

Table 3.3: Maximum workspace of the KUKA youBot arm in V-REP environment.

Axis data	Range
Axis 1 (A1)	±169°
Axis 2 (A2)	+ 75°/– 90°
Axis 3 (A3)	+ 131°/– 131°
Axis 4 (A4)	±102°
Axis 5 (A5)	±90°
Gripper	Detachable, 2 fingers with 70 mm Gripper range.

Figure 3.7 shows the minimum workspace of the joints 1, 2, 4 and 5.

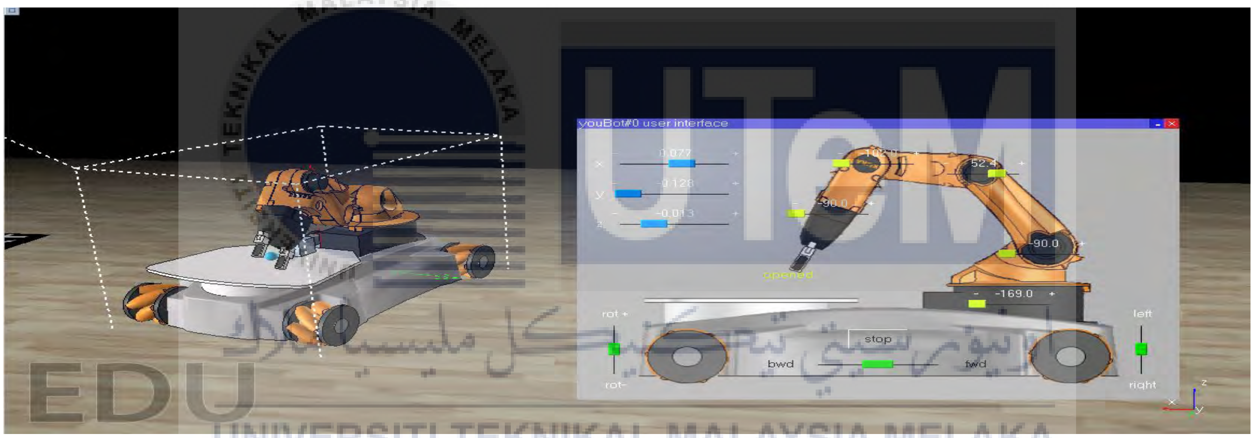

Figure 3.7: Minimum workspace of the joints 1, 2, 4 and 5.

Figure 3.8 shows the maximum workspace of the joints 1, 2, 4 and 5.

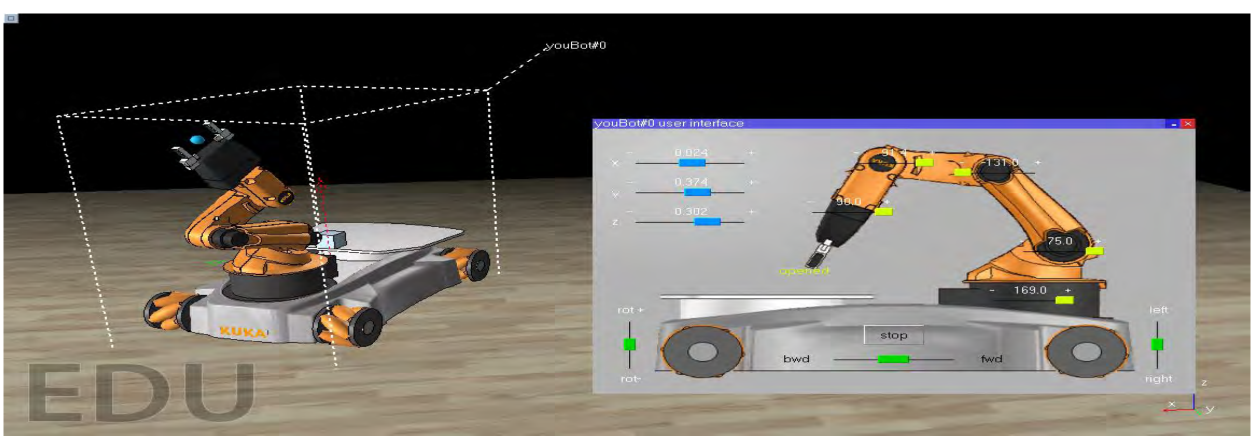

Figure 3.8: Maximum workspace of the joints 1, 2, 4 and 5.

Figure 3.9 shows the minimum workspace of the joints 1, 3, and 5.

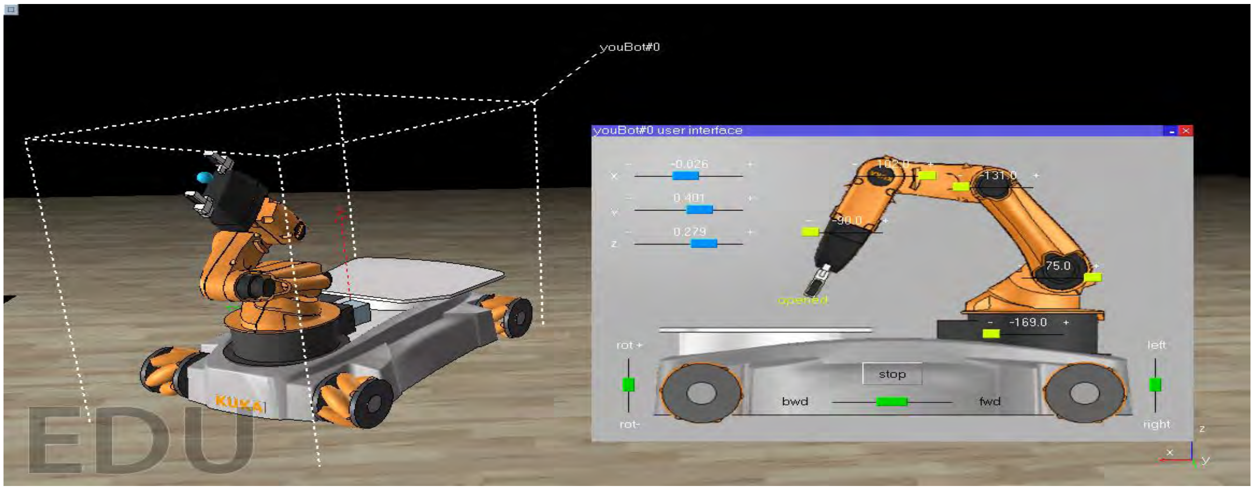

Figure 3.9: Minimum workspace of the joints 1, 3, and 5

Figure 3.10 shows the maximum workspace of the joints 3, and 5.

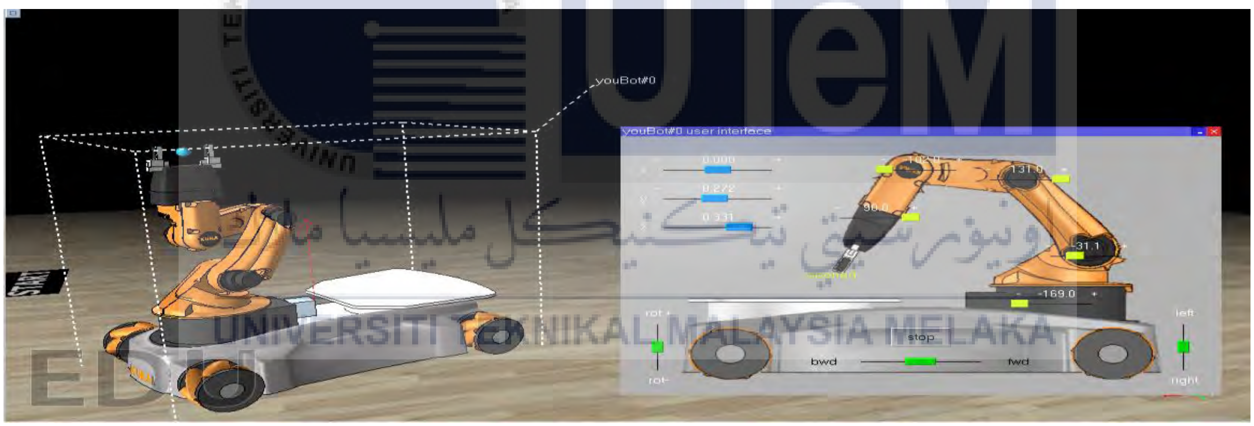

Figure 3.10: Maximum workspace of the joints 3, and 5

3.5 Setup of the simulation

In order to fulfill the setup of the simulation several steps need to done. The first step is starting the V-REP simulator, then Create the V-REP environment by Adding the KUKA YouBot.

The second step is to construct the bricks depends on the standard dimensions (112.5 mm, 225 mm and 75 mm) or depending on the desired dimensions.

The third step is to put the bricks to the initial picking area, then use the picking's gap equations (equation 3.1) to put the bricks in an understandable pattern for the KUKA manipulator.

Appendix C illustrates the steps in detail with respective figures.

Figure 3.11 shows the V-REP environment for the curved wall brick laying using 5 DOF mobile robot.

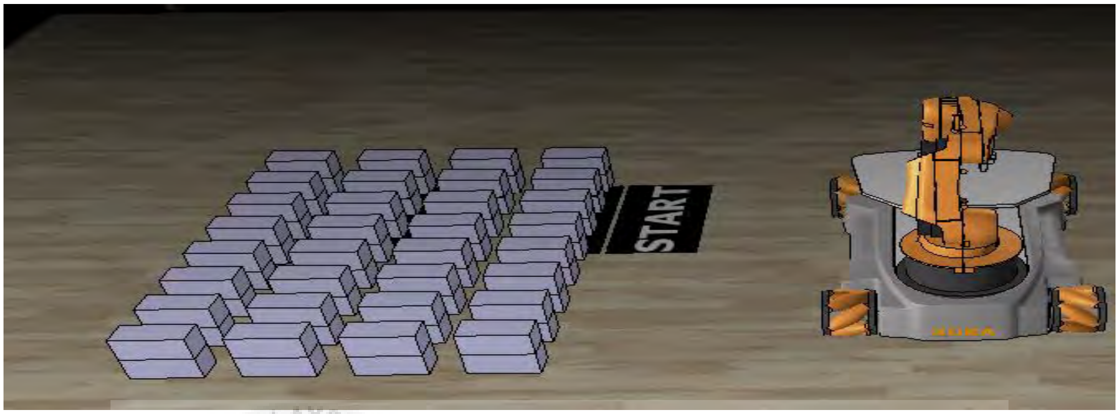

Figure 3.11: Setup of the V-REP environment.

At this stage the V-REP environment is constructed, the next step is to start writing the code in V-REP simulator using LUA programming language. In order to write the code, it is necessary to have a detailed flowchart.

3.6 Detailed flowchart of the programming code

The flowchart starts with the main function of the curved wall brick laying process. This function calls the other functions in order to complete the brick laying process. The first function to be called is the initial conditions that need to be an input to the manipulator. The loop technique is used to do the routine pick and place the bricks based on the initial conditions.

In the loop, three functions are called. The first function is the picking function, in this function the KUKA manipulator is directed to the picking area. However, the piking gap calculations (equation 3.1) are applied to guide the KUKA mobile robot to move to the desired brick and pick it. The second function is the placing function, in this function the placing gap calculations (equation 3.2) and the parabolic equation (equation 3.3) are applied to place the bricks in the desired curve structure.

Moreover, the rotation angle of the brick is also calculated in this function. The third function is the output function, this function is to display the actions and status of the manipulator such as the coordinate and the rotation angle of the gripper to the user. This process is repeated until reaching the end point or if the maximum number of bricks is reached. Figures 3.12 to 3.19 shows the detailed flowchart of the programming code.

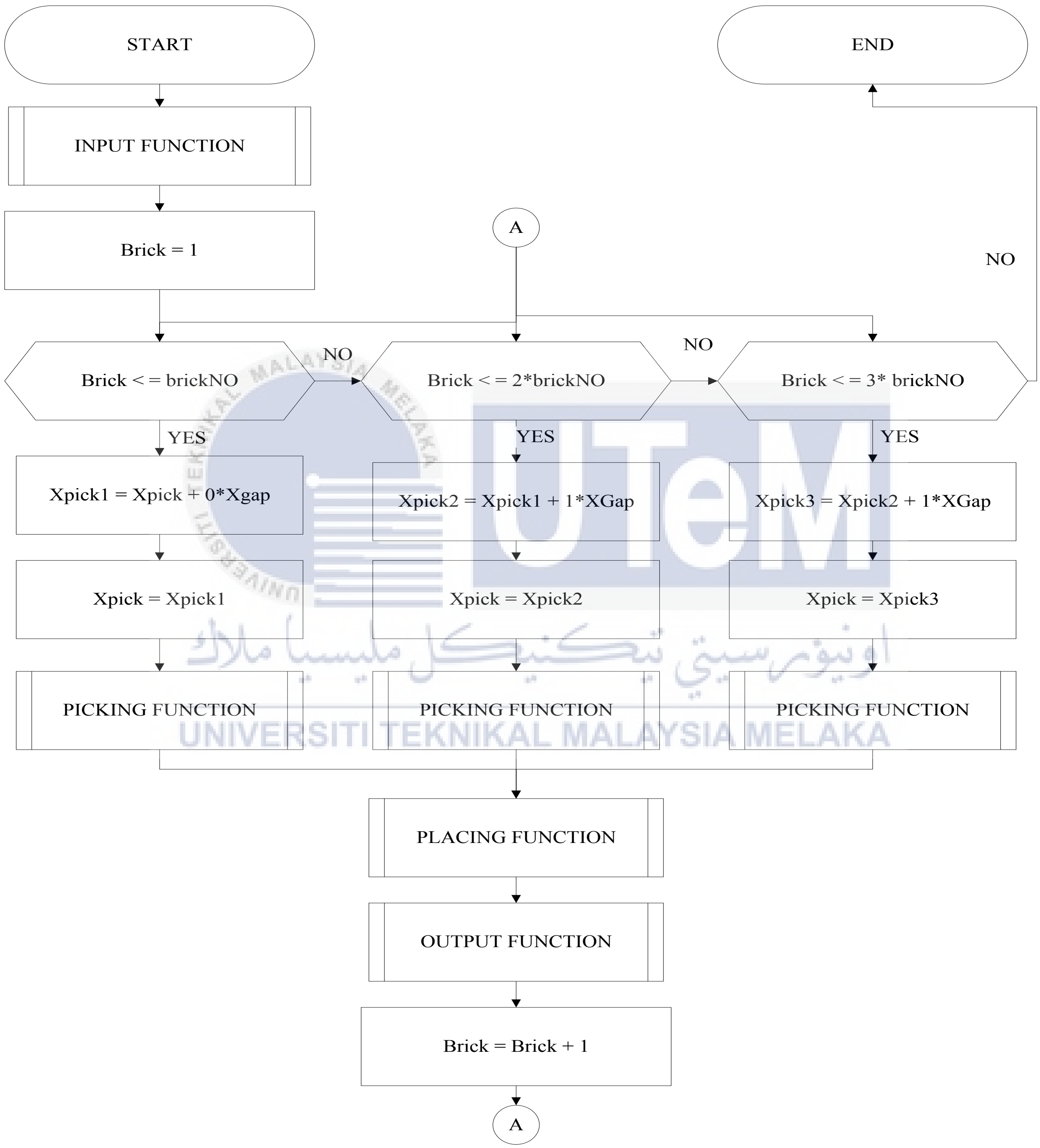

Figure 3.12: Brick laying main Function.

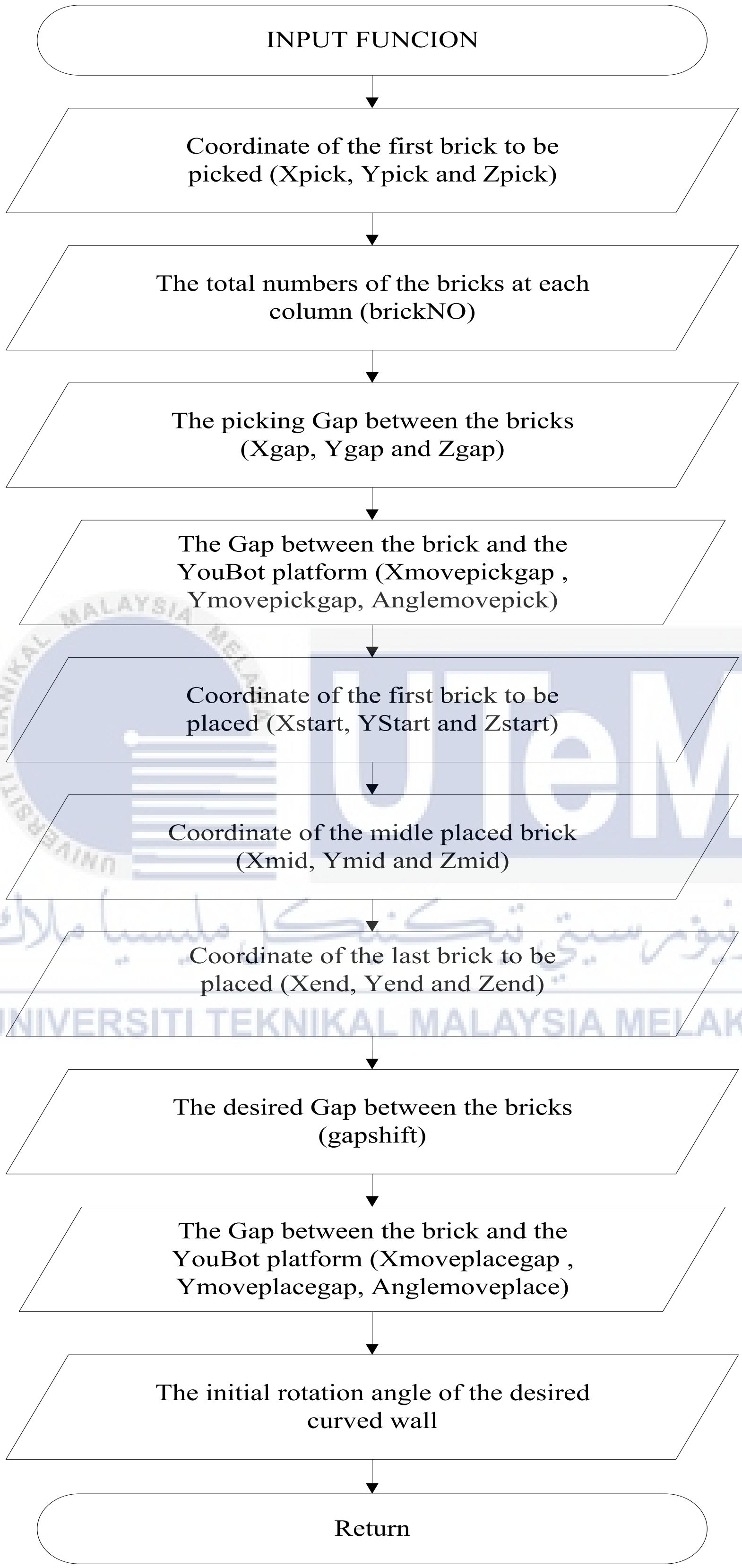

Figure 3.13: The initial conditions that need to be an input to the manipulator

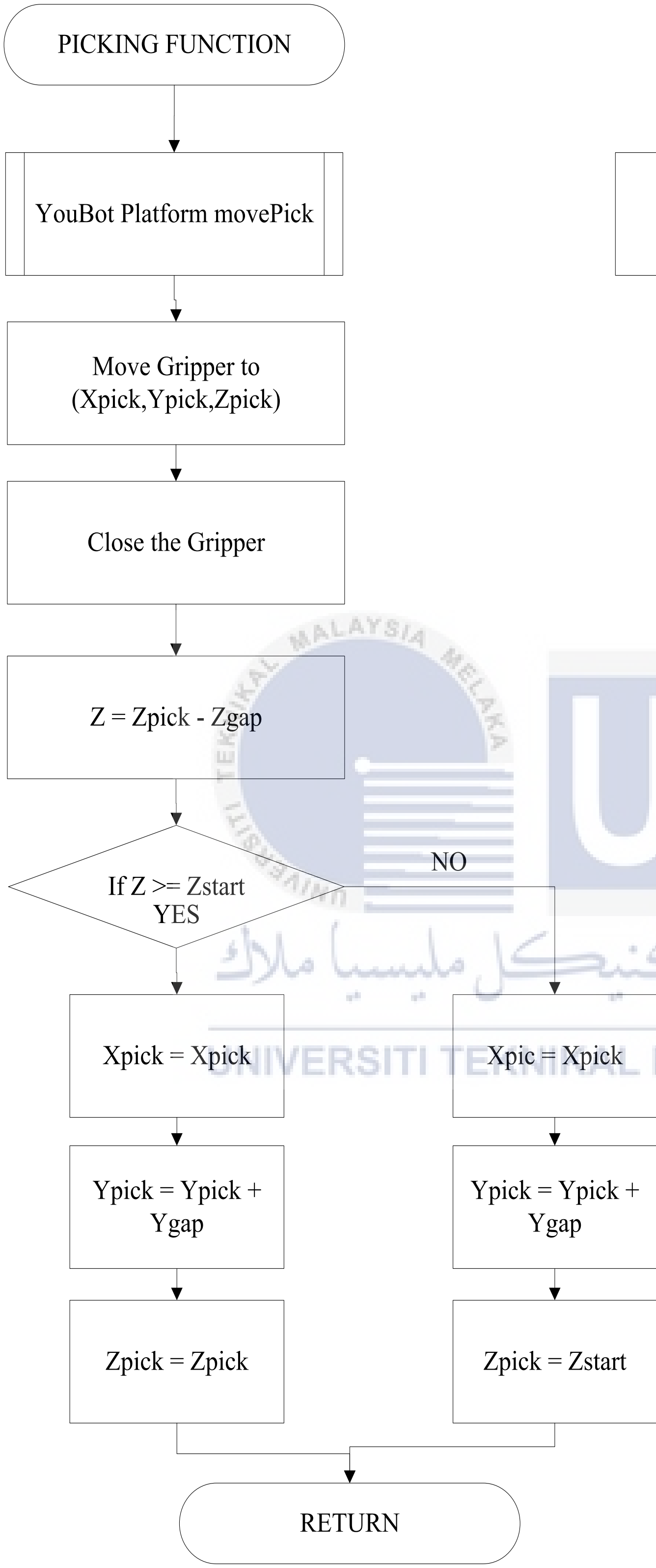

Figure 3.14: Picking the bricks from the picking area.

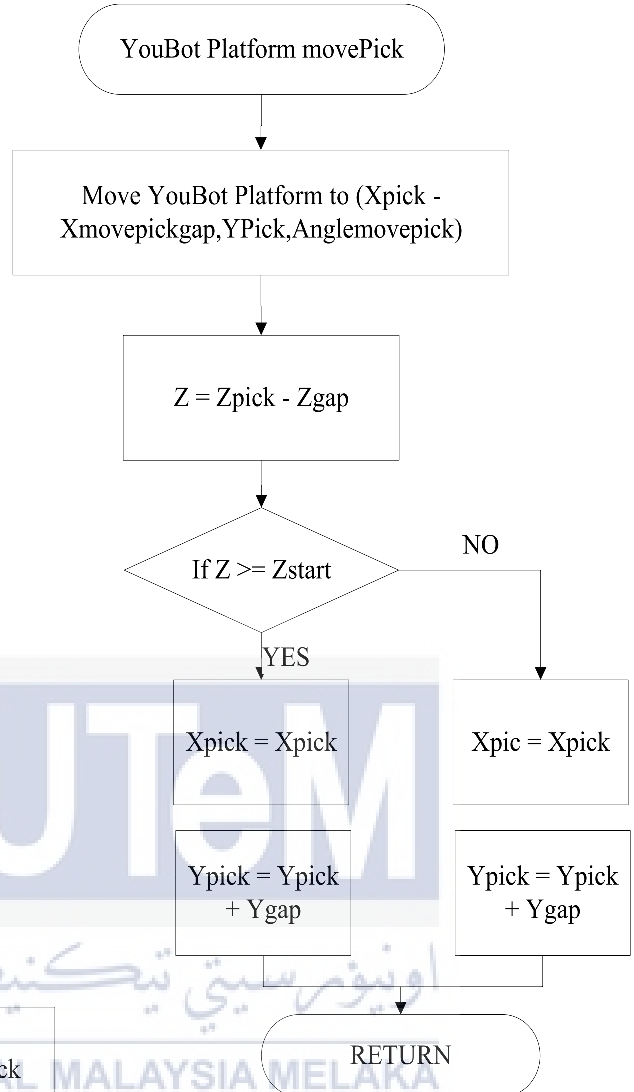

Figure 3.15: Moving the YouBot to the picking area.

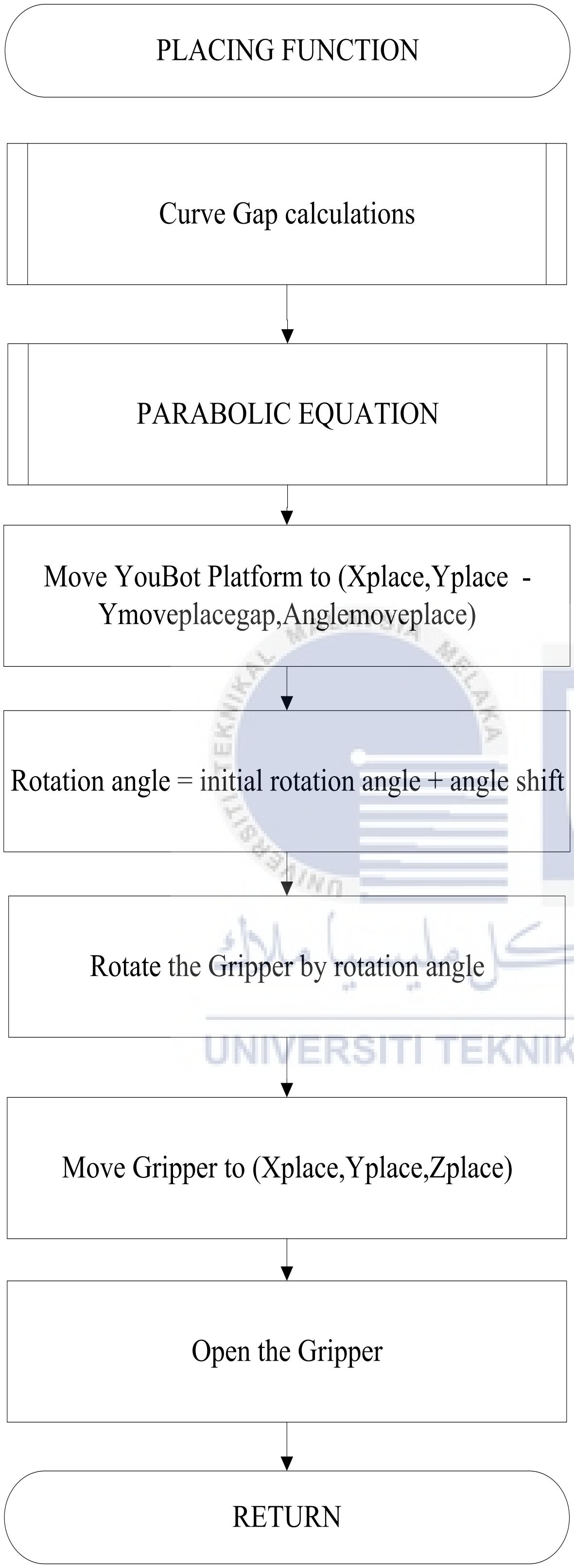

Figure 3.16: Placing the bricks at the placing area.

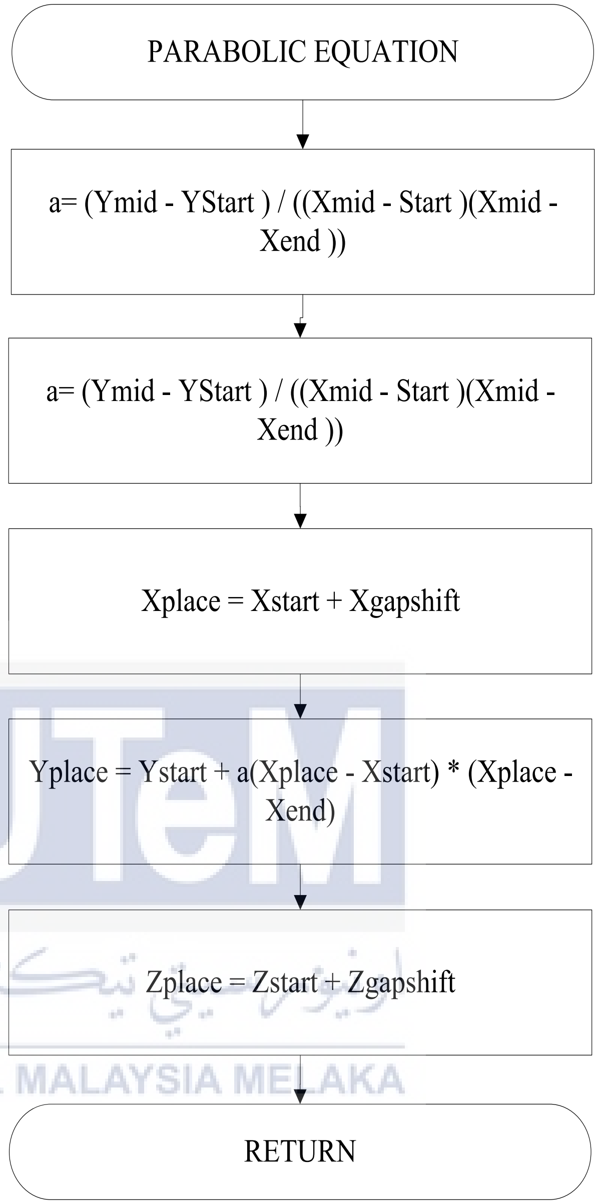

Figure 3.17: Parabolic equation. Formulating and applying.

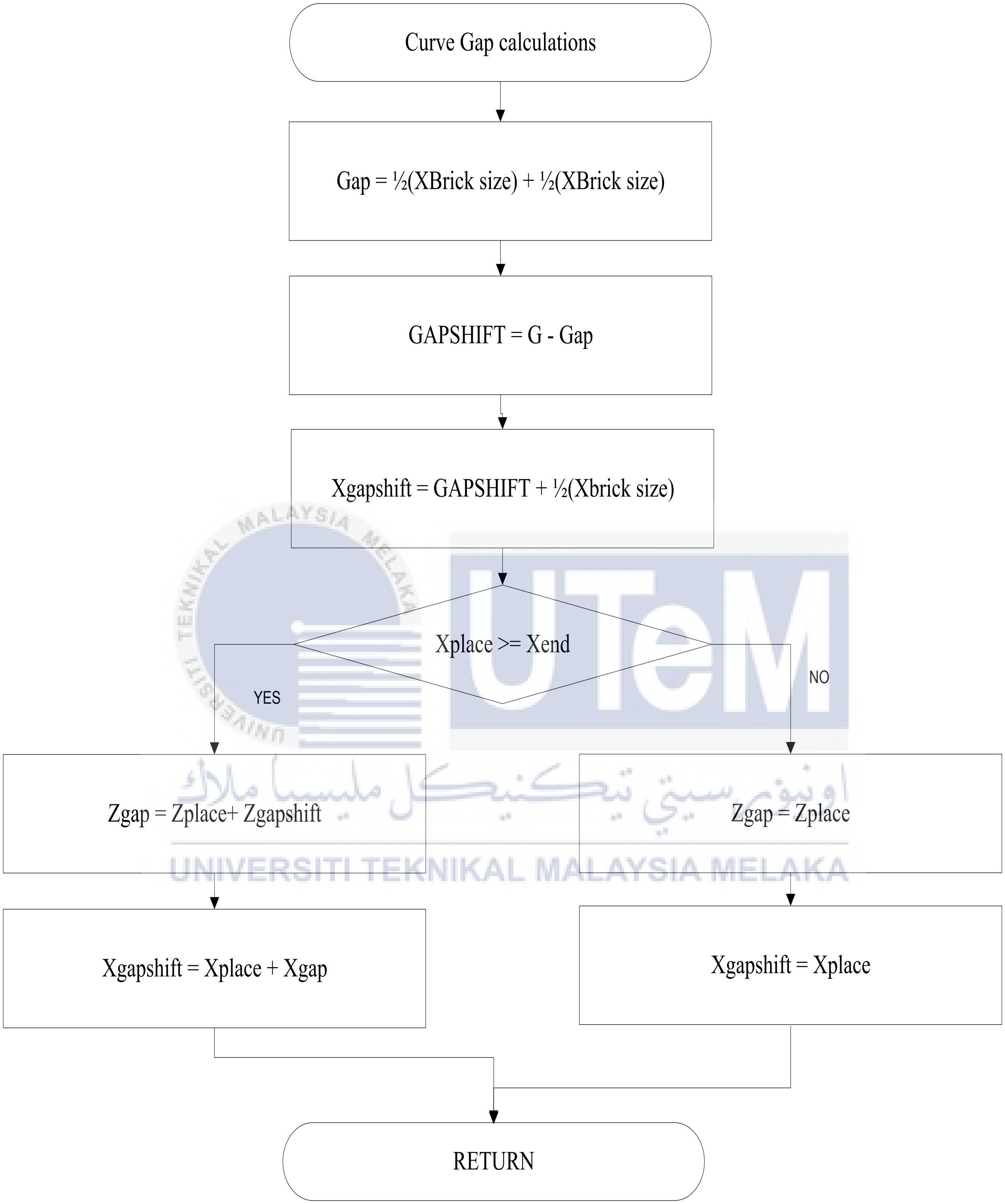

Figure 3.18: Calculation of the curved wall gap between the bricks.

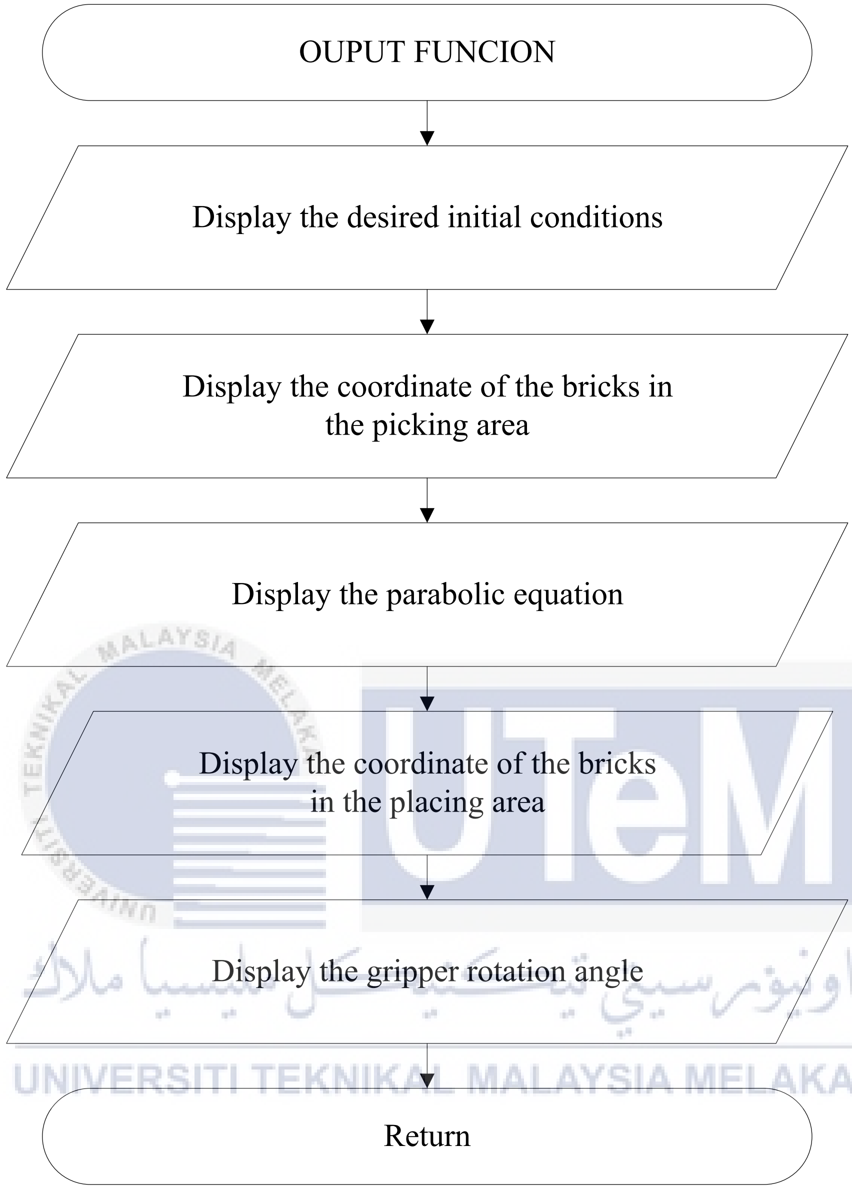

Figure 3.19: Display the actions and status of the manipulator.

3.7 The type of data gathered and the method of analyzing it

In this project the data gathered are the position of the bricks (X, Y and Z), the positions and velocities of the KUKA arm joints.

The analysis of this project will be carried on curve analysis, error analysis and the accuracy percentage of using KUKA manipulator as a curved wall brick laying mobile robot.

3.7.1 Curve analysis

The parabolic equations provide the advantage of finding the arc length and the curvature of the curved wall theoretically.

The arc length give the advantage of finding the Estimated Brick Numbers (EBN) reqired in the bricklaying process.

The arc length of the curved wall and the EBN are calculated mathematecally using equation 3.6 [13].

$$L = \int_{x_{initial}}^{x_{final}} \sqrt{1 + (\frac{dy}{dx})^2}\, dx$$

$$EBN = \frac{L}{g + Brick\ size} + SF \qquad (3.6)$$

Where :

L: The arc length of the curve.

$x_{initial}$: The initial brick position.

x_{final}: The final brick position.

g : The Gap between two bricks.

SF : Safety factor (SF assumed to be 5 bricks)

3.7.1 Error analysis

In order to approach the objective of this project which is to evaluate the accuracy of the brick positioning using KUKA YouBot utilizing parabolic polynomial equation, the error analysis is carried on by comparing the desired curve obtained from the parabolic polynomial equation with the actual bricks coordinates placed by the KUKA manipulator obtained from the V-REP. The mean error (m_e), percentage error (p_e), the accuracy percentage (A_p) and the root mean square error (RMSe) equations 3.7, 3.8, 3.9 and 3.10.

$$m_e = |Desired\ brick\ coordinates - Actual\ brick\ coordinates| \quad (3.7)$$

$$p_e = \left|\frac{Desired\ brick\ coordinates - Actual\ brick\ coordinates}{Desired\ brick\ coordinates}\right| \times 100\% \quad (3.8)$$

$$A_p = \left(1 - \left(\left|\frac{Desired\ brick\ coordinates - Actual\ brick\ coordinates}{Desired\ brick\ coordinates}\right|\right)\right) \times 100\% \quad (3.9)$$

$$\mathrm{RMSe} = \sqrt{\frac{\sum_{i=1}^{n}(X_{Theoretical} - X_{Actual})^2}{n}} \quad)3.10($$

3.8 The procedures of conducting the simulation and getting the data

In order to complete this project, the procedures and methodology of computing the resulted data is shown in form of a flowchart in figure 3.19. the flowchart starts by asking the user to put the desired initial positions, then Matlab is used proof the ability of the parabolic equation to produce the desired curve. Then the V-REP environment is set. Then process of the routine pick the bricks from the picking area and placed it at the placing area based on the parabolic equation.

The flowchart shown in Figure 3.20 illustrate the procedures and the methodology of conducting the simulation and compute the resulted data.

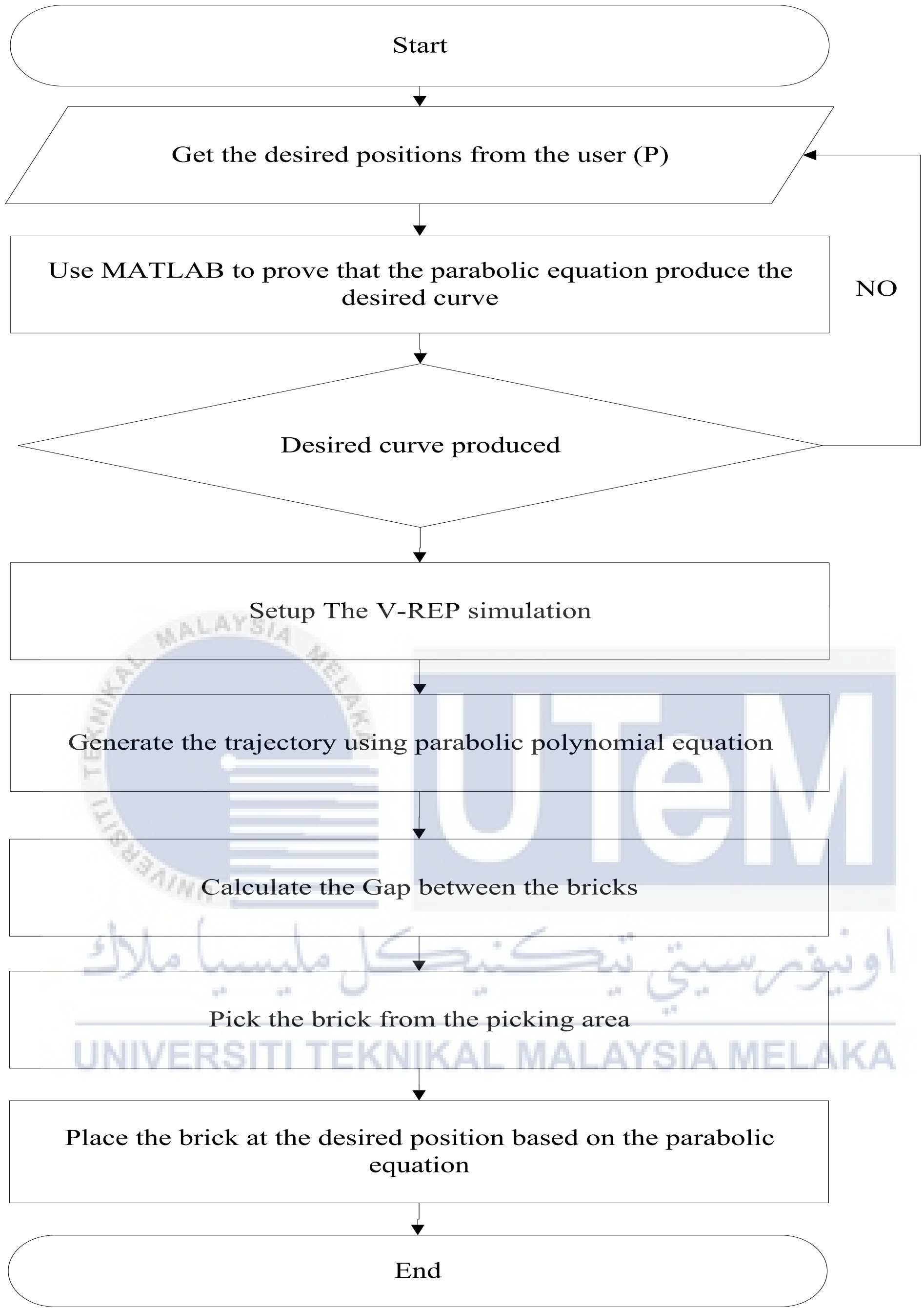

Figure 3.20: procedures and the methodology of conducting the simulation

3.9 Consideration on the validity of the simulation

In this section, the discussion of the factors that may invalidate the results of the simulation and the reliability of the data in terms of effects of error and uncertainty in making the inverse kinematics in order to calculate the joints angles rotation values.

3.9.1 Factors that may invalidate the simulation result

The accuracy of positioning the curved wall bricks in VREP simulation may be not as same as the positioning of curved wall bricks in the real world. One of the factors that may invalid the simulation result is the friction between the mobile wheels of the YouBot and the ground floor. In V-REP environment the floor is smooth so the slip of the wheel is negligible and not considered. However, in the real world environment the friction between the ground floor and the wheels of the mobile YouBot must be taken into consideration. Moreover, the gravity is a factor that effects the weight of the bricks as the weight is directly proportional to the gravity. In VREP simulation the gravity is assumed to be the standard value (9.81 m/s^2) but in real world the gravity is not exactly the same. The gravity in real world dependents upon the location. There are slight variations in the value of the gravity about the earth's surface. These variations result from the varying density of the geologic structures below each specific surface location. For these reasons, in future it is necessary to do the curved wall brick laying experiment in real world before using the KUKA YouBot as a curve brick layer.

3.9.2 Discussion on the reliability of data

Reliability of using the KUKA YouBot as curved wall brick laying robot shows that the resulted brick positioning may not be fully trusted. This is due to the uncertainties of the joints rotation angles. The uncertainties happen due to the auto calculations of the inverse kinematics by the invers kinematics module (IK module) in V-REP simulator. As the values of the joints angles are controlled by the V-REP simulator, the uncertainty in terms of geometric problem in Cartesian space and the singularities take place.

CHAPTER 4

RESULT

In this chapter, the findings and results of the project will be discussed and analyzed. The flow of this chapter is based on the detailed flowchart discussed in chapter three. This chapter will contain the following subtitles; Initial conditions (Input function), results of the parabolic equation and curve analysis, picking area results, results of the V-REP simulation (KUKA YouBot), analyzing the error and the distance between the KUKA mobile platform and the bricks.

4.1 Initial conditions (Input function)

This is the first step after completing the environmental setup of the V-REP simulation. In this step the initial conditions that are required by the KUKA YouBot in order to know the desired process are asked to be insert by the user. Table 4.1, 4.2, 4.3, 4.4 and 4.5 illustrate the required initial conditions with some assumed values.

Table 4.1: Coordinate of the first brick

COORDINATE OF THE FIRST BRICK	X	Y	Z
VALUE	0.875	-0.8	0.075

Table 4.2: Gap shift at picking area.

GAPSHIFT	Xgap	Ygap	Zgap
VALUE	0.15	0.1	0.05

Table 4.3: KUKA YouBot platform Gap shift at picking area.

YouBot platform Movegap	Xmovegap	Ymovegap	Anglemovegap
VALUE	0.1	0.1	0

Table 4.4: Maximum bricks at each column.

COLUMN NUMBER	1	2	3	4
MAXIMUM BRICKS	16	16	16	16

Table 4.5: Initial conditions required by the parabolic equation.

INITIAL CONDITIONS	Start point		Max point (Eccentricity)		End point		Initial Rotation angle
SYMBOL	X	Y	X	Y	X	Y	θ
VALUE	-0.8	0	-0.014	-0.3	0.259	0	60

4.2 Results of the parabolic equation and curve analysis.

In this section it is necessary to prove that the parabolic equation is able to produce the desired curve based on the initial conditions. Equation 3.3 is used in order to get the graph of the parabolic equation. Appendix A illustrates the desired position of three different curves. However, the arc length and the estimated brick numbers of the desired curved wall are found by using equation 3.6 and equation 3.7. Equations 3.6 and 3.7 are written in term of a code in MATLAB software to obtain the arc length, and the estimated brick numbers of the desired curved wall. Appendix B shows the MATLAB coding.

Figure 4.1 shows serval different curves (12 curves), which are generated by using the parabolic polynomial equation.

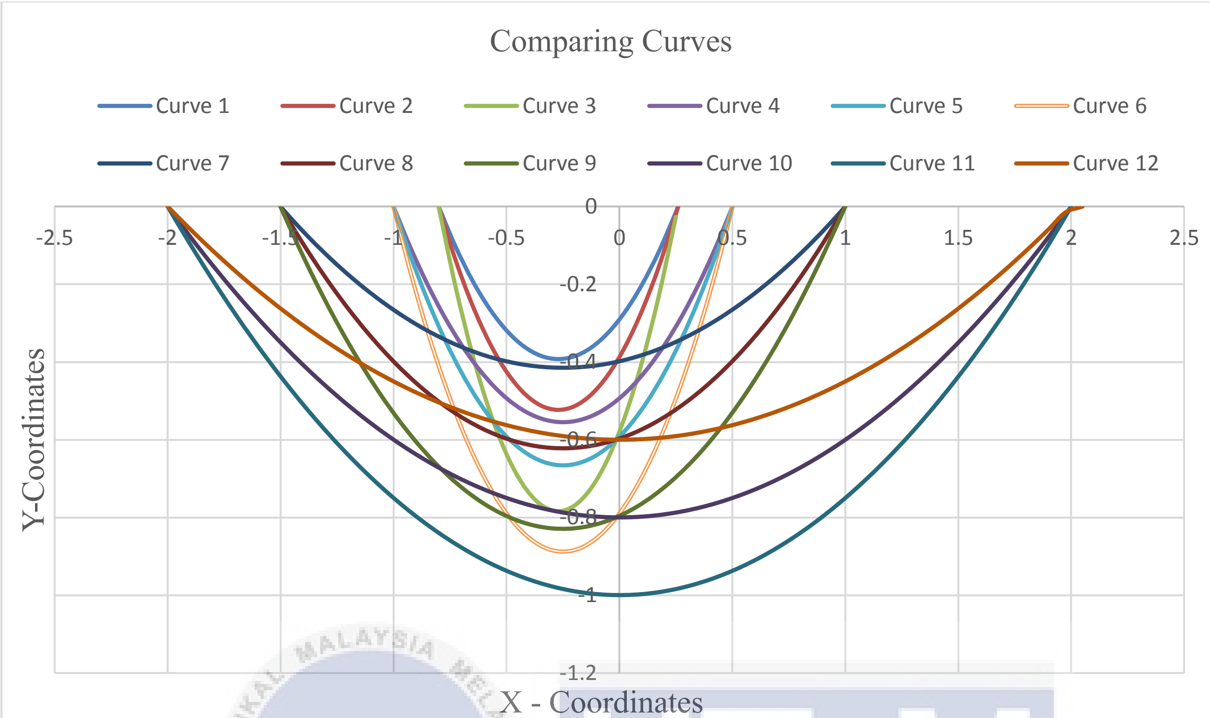

Figure 4.1: Several types of curves

Table 4.6 illustrate the comparison of the desired curves in terms of the parabolic equation, arc length and the estimated brick numbers.

Table 4.6: Comparing the curves in figure (4.1)

Comparison / Curves	Start point X	Start point Y	Max point X	Max point Y	End point X	End point Y	Equation	Arc length (m)	EBN bricks
Curve 1	-0.8	0	-0.014	-0.3	0	0.259	y= 1.3981 (x+0.8) (x−0.259)	1.3694	13
Curve 2	-0.8	0	-0.014	-0.4	0	0.259	y= 1.8641 (x+0.8) (x−0.259)	1.5559	14
Curve 3	-0.8	0	-0.014	-0.6	0	0.259	y= 2.7962 (x+0.8) (x−0.259)	1.9779	16

Curve 4	-1	0	-0.014	-0.5	0	0.5	y= 0.9866 (x+1) (x−0.5)	1.9394	16
Curve 5	-1	0	-0.014	-0.6	0	0.5	y= 1.1839 (x+1) (x−0.5)	2.09388	17
Curve 6	-1	0	-0.014	-0.8	0	0.5	y= 1.5785 (x+1) (x−0.5)	2.4336	19
Curve 7	-1.5	0	-0.014	-0.4	0	1	y= 0.2655 (x+1.5) (x−1)	2.6730	20
Curve 8	-1.5	0	-0.014	-0.6	0	1	y= 0.398 (x+1.5) (x−1)	2.86648	21
Curve 9	-1.5	0	-0.014	-0.8	0	1	y= 0.530 (x+1.5) (x−1)	3.10853	23
Curve 10	-2	0	-0.014	-0.6	0	2	y= 0.150 (x+2) (x−2)	4.2285	29
Curve 11	-2	0	-0.014	-0.8	0	2	y= 0.200 (x+2) (x−2)	4.3930	30
Curve 12	-2	0	-0.014	-0.4	0	2	y= 0.100 (x+2) (x−2)	4.1043	29

From figure 4.1 and table 4.6 it is clear that using the parabolic equation gives the user an advantage of producing any desired curve with different arc length. The arc length gives the advantage of estimating the number of bricks needed to complete the curved wall brick laying. Figures 4.3, 4.4 and 4.5 shows that the number of bricks are 11, 13 and 15 respectively. By comparing the bricks number in the V-REP simulator with the estimated bricks number (in curve 1, curve 4 and curve 5), it is clear that the EBN give an enough number of bricks required in order to complete the first layer of the curved wall brick laying process.

The examples on how to calculate the arc length, estimated brick numbers (EBN) and the curvature (if necessary) are shown in Appendix A, while the MATLAB codes are shown in Appendix B.

4.3 Picking area results

This results show the coordinates of the bricks at the picking area. The results show that there are 4 columns, in which there are 16 bricks in each columns, so there are 64 total bricks. These number could be increase or decrease base on the desire of the user. The KUKA YouBot starts to move to the Brick's coordinates with respect to the gap between bricks and the YouBot platform, so the youBot moves to the coordinates as shown in table 4.8.

Table 4.7: KUKA YouBot platform coordinate at picking area.

COORDINATE OF THE youBot platform	X	Y	Angle
VALUE	X - Xmovegap	Y- Ymovegap	0

The gripper of the KUKA arm YouBot moves to the center of the first brick, then the gripper will be closed to pick the brick. After delivering the brick at the placing area, the KUKA platform will move with respect to the specified gap.

The coordinates of the bricks at the picking area are illustrated in Appendix A.

4.4 Results of the V-REP simulation.

Figure 4.2 shows picking area in the V-REP simulator. Figures 4.3, 4.4 and 4.5 shows the placing area of three different desired curved wall bricks. The three curves are having the same parabolic equation with the first, forth and the fifth curves in table 4.6.

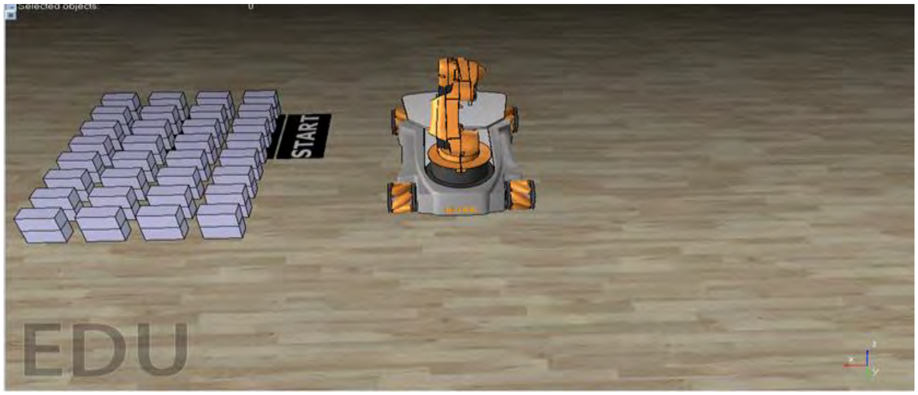

Figure 4.2: Picking area in the V-REP simulator.

Figure 4.3: Actual first curved wall Brick in V-REP

Figure 4.4: Actual second curved wall Brick in V-REP

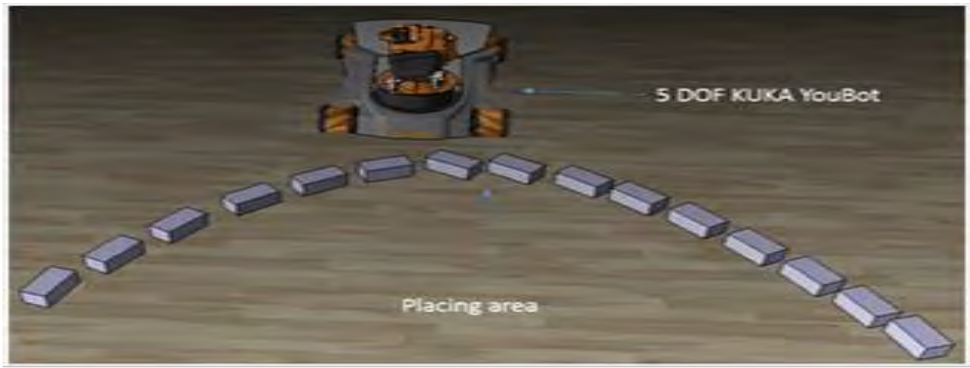

Figure 4.5: Actual third curved wall Brick in V-REP

4.5 Analyzing the error and accuracy percentage.

In this section, the error and accuracy analysis is done for the first curve In figure 4.3. V-REP is a simulation tool, so for that reason the accuracy of the other generated curves based on the parabolic equation is assumed to be the same or very close to the accuracy of the first generated curve. But in the physical world experiment the accuracy analysis is necessary to be done for every generated curve trajectory. However, Equations 3.8 and 3.9 are used to calculate the mean error and the accuracy percentage of using the KUKA manipulator as a curved wall brick layer in V-REP environment. Figure 4.6 shows the accuracy percentages of using KUKA manipulator as a curved wall brick layer for the first curve.

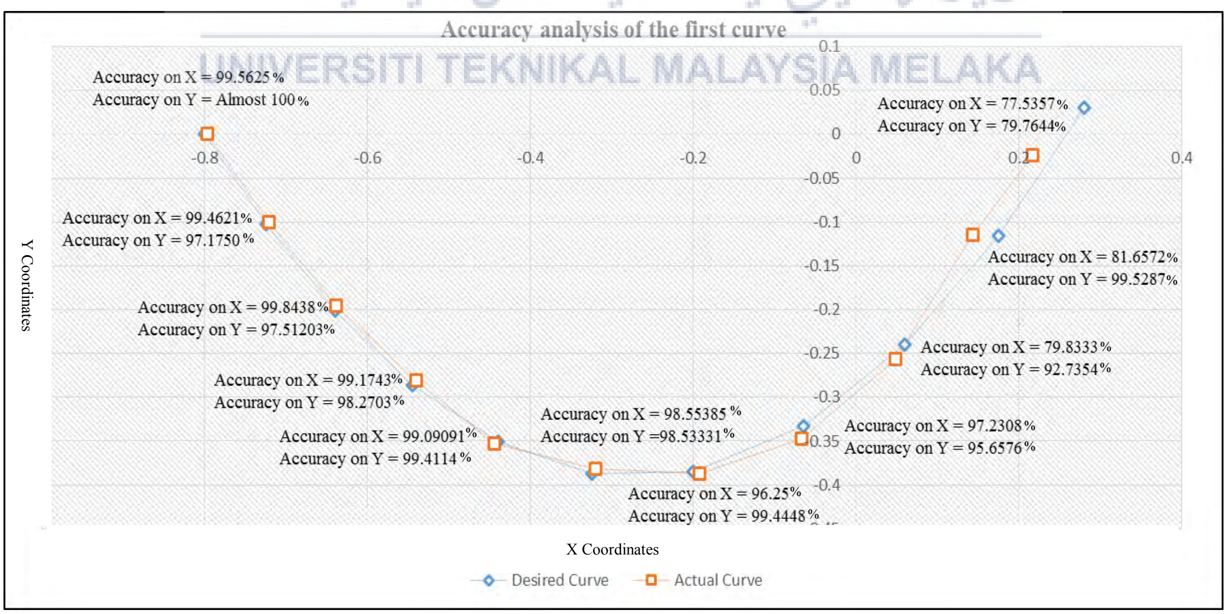

Figure 4.6: Accuracy percentages of the first curve

The maximum accuracy is almost 100%, while the minimum accuracy is 77.53571%. The error is an inversely proportional to the accuracy percentage, in other words the error increases as the accuracy decrease. The Accuracy analysis show that the error starts to increase as curve get near to the end point. This mean the error is a systematic. The Root Mean Square error (RMSe) of using the KUKA YouBot to construct the first curved wall is 0.02190791. this means that the accuracy is acceptable in curved wall brick layering process. The calculations of the error, percentage accuracy and the RMSe are illustrated in Appendix A.

4.6 The distance between the KUKA mobile platform and the bricks

In this section the consideration of the distance between the YouBot platform and the bricks is discussed. In order to avoid any clashes between the manipulator and the bricks, some distance should be in between the manipulator and the bricks. Figure 4.7 shows a method of observing the actions and performance of the KUKA YouBot by applying several distances between the YouBot and the bricks. Figures 4.8, figure 4.9 and figure 4.10 show the distances between the KUKA YouBot and the brick in V-REP environment.

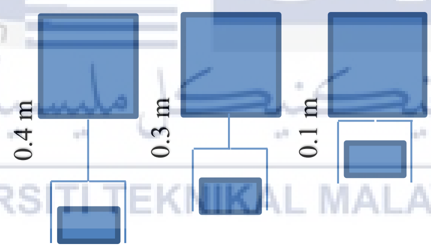

Figure 4.7: observing the actions and performance of the KUKA YouBot when different distances are applied

Figure 4.8: Distances between bricks and robot is 0.1 m

Figure 4.9: Distances between bricks and robot is 0.3 m

Figure 4.10: Distances between bricks and robot is 0.4 m

Figure 4.8 show that the distance between the YouBot and the brick (0.1 m) cause a clash between the manipulator and the brick. So this distance should not be used. The 0.3 m distance between the YouBot and the bricks in figure 4.9 is used to avoid the clash between the manipulator and the brick. While the 0.4 m distance between the YouBot and the bricks in figure 4.10 is used to avoid the clash between the manipulator and the previous arranged bricks.

CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1 Conclusion

In conclusion, the trajectory profile relative to the curved wall brick laying is designed with the use of the parabolic polynomial equation. the parabolic equation is used to generate trajectories that enables the manipulator (KUKA) to place the bricks in any desired curve structure.

In this project, several different types of curved are compared in terms of the parabolic equations, arc lengths and the estimated brick number. From this comparison, the prove of the ability of the parabolic equation to produce a desired curve is achieved.

In general, curved wall brick laying is a process of laying bricks in a desired curved structure. In this project, the brick laying process is done by using a five degree of freedom (5 DOF) manipulator namely KUKA YouBot. The manipulator (KUKA) picks the bricks from the picking area and place (lay) it in the desired placing area with a structure of a desired curve.

This project is done in the Virtual Robot Experimentation Platform (VREP) simulator. However, the analysis and drawings are done using the Matrix Laboratory (MATLAB) and the Microsoft Excel.

At the end, the accuracy analysis show that the error starts to increase as the curve get near to the end point, this show that the error is a systematic error. However, the minimum accuracy of using the KUKA manipulator as a curved wall brick laying robot is 77.53571%. Moreover, the Root Mean Square error (RMSe) is 0.02190791. this indicate that the accuracy is acceptable.

The main points learned from this research are the fundamental knowledge of the curved wall brick laying trajectory generation, the using of the V-REP as a simulator for curved wall brick laying process and the use of the parabolic equation in order to achieve any desired curve.

5.2 Future work

The future work of this project is to increase the accuracy of the brick positioning by investigating more on the V-REP Properties and the Kinematics of the KUKA manipulator. At the same time increasing the number of the layers (starting from the second layer) is highly recommended as this research achieve the construction and design of the first layer of any desired curve wall. This project is done in a simulation environment (V-REP), so some external factors, such as obstacle avoidance and the floor friction is negligible, so it is recommended to take consideration of all the factors that may affect the process of brick laying.

At the end, testing the experiment in the real (Lab or Physical) environment is highly recommended, especially to analyze the accuracy of every constructed curved wall.

REFERENCES

[1] J. Seo, M. Yim and V. Kumar, “Assembly planning for planar structures of a brick wall pattern with rectangular modular robots,” *2013 IEEE International Conference on Automation Science and Engineering (CASE)*, Madison, WI, 2013, pp. 1016-1021.

[2] M. L. Aguair and K. Behdinan, “Design, Prototyping, and Programming of a Bricklaying Robot,” *J. Student Sci. Technol.*, vol. 8, no. 3, pp. 27–34, 2015.

[3] H. C. Choi, G. p. Jung and K. J. Cho, “Design of a milli-scale, biomimetic platform for climbing on a rough surface,” *2015 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Zhuhai, 2015, pp. 2205-2210.

[4] J. J. Craig, “Introduction to Robotics: Mechanics and control 3rd,” *Prentice Hall*, vol. 1, no. 3, p. 408, 2004.

[5] P. Gallo, A. Krasilov, A. Lyakhov, I. Tinnirello, and G. Bianchi, “Breaking layer 2: A new architecture for programmable wireless interfaces,” *Int. Conf. ICT Converg.*, pp. 342–347, 2012.

[6] J. K. Rai and R. Tewari, “Quintic polynomial trajectory of biped robot for human-like walking,” *2014 6th International Symposium on Communications, Control and Signal Processing (ISCCSP)*, Athens, 2014, pp. 360-363.

[7] H. R. E. Haghighi and M. A. Nekoui, “Inverse kinematic for an 8 degrees of freedom biped robot based on cubic polynomial trajectory generation,” *The 2nd International Conference on Control, Instrumentation and Automation*, Shiraz, 2011, pp. 935-940.

[8] X. Li, S. Tan, X. Feng and H. Rong, “LSPB Trajectory Planning: Design for the Modular Robot Arm Applications,” *2009 International Conference on Information Engineering and Computer Science*, Wuhan, 2009, pp. 1-4.

[9] E. Cuevas, D. Zaldivar, M. Pérez-Cisneros, and M. Ramírez-Ortegón, ”Polynomial trajectory algorithm for a biped robot,” *Int. J. Robot. Autom*., vol. 25, no. 4, pp. 294–303, 2010.

[10] W. Li, W. Cui, Y. Zhang and C. Yu, “Trajectory analysis of painting robot on building wall,” 2016, *11th International Conference on Computer Science & Education (ICCSE)*, Nagoya, 2016, pp. 563-567.

[11] Y. Zhang, Y. Li and X. Xiao, “A novel kinematics analysis for a 5-DOF manipulator based on KUKA youBot,” *2015 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Zhuhai, 2015, pp. 1477-1482.

[12] R. V. Cano, “Modelling in Robotics Derivation of Kinematic Equations for the youBot and implementation of those in Simulink,” *Umea University*, 2015.

[13] J. Stewart, “Calculus Early Transcendentals 6th edition,” *Tomson Brooks/Cole*, 2008.

[14] R. Bischoff, U. Huggenberger and E. Prassler, “KUKA youBot - a mobile manipulator for research and education,” *2011 IEEE International Conference on Robotics and Automation*, Shanghai, 2011, pp. 1-4.

[15] L. Zhang and C. Zhou, "Kuka youBot arm shortest path planning based on geodesics,” *2013 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Shenzhen, 2013, pp. 2317-2321.

APPENDIX A

Bricks coordination's and examples on the used equations and analysis.

Table A.1 Bricks desired coordinate at placing area based on the parabolic equation.

Curves	Brick NO.	Gap shift	Place X	Place Y	Place Z
Curve 1	1	0.075	-0.8	0	0.025
	2	0.085	-0.725	0.102907171	0.025
	3	0.095	-0.64	0.200590628	0.025
	4	0.105	-0.545	0.285946119	0.025
	5	0.115	-0.44	0.351033599	0.025
	6	0.125	-0.325	0.387077227	0.025
	7	0.135	-0.2	0.38446537	0.025
	8	0.125	-0.065	0.332750599	0.025
	9	0.115	0.06	0.239594361	0.025
	10	0.105	0.175	0.115444085	0.025
	11	0.095	0.28	-0.030088594	0.025
Curve 2	1	0.075	-1	0	0.025
	2	0.085	-0.925	-0.126528204	0.025
	3	0.095	-0.84	-0.253825937	0.025
	4	0.105	-0.745	-0.37585536	0.025
	5	0.115	-0.64	-0.485868304	0.025
	6	0.125	-0.525	-0.576406264	0.025
	7	0.135	-0.4	-0.6393004	0.025
	8	0.125	-0.265	-0.665671542	0.025
	9	0.115	-0.14	-0.651612852	0.025
	10	0.105	-0.025	-0.606003504	0.025
	11	0.095	0.08	-0.537012336	0.025
	12	0.085	0.175	-0.452097852	0.025

	13	0.075	0.26	-0.358008224	0.025
	14	0.065	0.335	-0.260781288	0.025
	15	0.055	0.4	-0.165744548	0.025
	16	0.045	0.455	-0.077515174	0.025
Curve 3	1	0.075	-1	0	0.025
	2	0.085	-0.925	-0.126528204	0.025
	3	0.095	-0.84	-0.253825937	0.025
	4	0.105	-0.745	-0.37585536	0.025
	5	0.115	-0.64	-0.485868304	0.025
	6	0.125	-0.525	-0.576406264	0.025
	7	0.135	-0.4	-0.6393004	0.025
	8	0.125	-0.265	-0.665671542	0.025
	9	0.115	-0.14	-0.651612852	0.025
	10	0.105	-0.025	-0.606003504	0.025
	11	0.095	0.08	-0.537012336	0.025
	12	0.085	0.175	-0.452097852	0.025
	13	0.075	0.26	-0.358008224	0.025
	14	0.065	0.335	-0.260781288	0.025
	15	0.055	0.4	-0.165744548	0.025
	16	0.045	0.455	-0.077515174	0.025

Table A.2: The coordinates of the bricks at picking area.

COLUMNS NUMBER	**BRICK NUMBER**	**X**	**Y**	**Z**
COLUMNS 1	1	0.875	-0.8	0.075
	2	0.875	-0.8	0.025
	3	0.875	-0.7	0.075
	4	0.875	-0.7	0.025
	5	0.875	-0.6	0.075
	6	0.875	-0.6	0.025

	7	0.875	-0.5	0.075
	8	0.875	-0.5	0.025
	9	0.875	-0.4	0.075
	10	0.875	-0.4	0.025
	11	0.875	-0.3	0.075
	12	0.875	-0.3	0.025
	13	0.875	-0.2	0.075
	14	0.875	-0.2	0.025
	15	0.875	-0.1	0.075
	16	0.875	-0.1	0.025
	17	1.025	-0.8	0.075
	18	1.025	-0.8	0.025
	19	1.025	-0.7	0.075
	20	1.025	-0.7	0.025
	21	1.025	-0.6	0.075
	22	1.025	-0.6	0.025
	23	1.025	-0.5	0.075
	24	1.025	-0.5	0.025
COLUMNS 2	25	1.025	-0.4	0.075
	26	1.025	-0.4	0.025
	27	1.025	-0.3	0.075
	28	1.025	-0.3	0.025
	29	1.025	-0.2	0.075
	30	1.025	-0.2	0.025
	31	1.025	-0.1	0.075
	32	1.175	-0.1	0.025
	33	1.175	-0.8	0.075
	34	1.175	-0.8	0.025
	35	1.175	-0.7	0.075
	36	1.175	-0.7	0.025

COLUMNS 3	37	1.175	-0.6	0.075
	38	1.175	-0.6	0.025
	39	1.175	-0.5	0.075
	40	1.175	-0.5	0.025
	41	1.175	-0.4	0.075
	42	1.175	-0.4	0.025
	43	1.175	-0.3	0.075
	44	1.175	-0.3	0.025
	45	1.175	-0.2	0.075
	46	1.175	-0.2	0.025
	47	1.175	-0.1	0.075
	48	1.175	-0.1	0.025
COLUMNS 4	49	1.325	-0.8	0.075
	50	1.425	-0.8	0.025
	51	1.425	-0.7	0.075
	52	1.425	-0.7	0.025
	53	1.425	-0.6	0.075
	54	1.425	-0.6	0.025
	55	1.425	-0.5	0.075
	56	1.425	-0.5	0.025
	57	1.425	-0.4	0.075
	58	1.425	-0.4	0.025
	59	1.425	-0.3	0.075
	60	1.425	-0.3	0.025
	61	1.425	-0.2	0.075
	62	1.425	-0.2	0.025
	63	1.425	-0.1	0.075
	64	1.425	-0.1	0.025

Table A.3: Example to calculate the values in table A.1 and table A.2.

<table>
<tr><td rowspan="3">Example 4.1</td><td colspan="2">Formulate the parabolic equation</td></tr>
<tr><td>By Formula</td><td>By values</td></tr>
<tr><td>$a = \frac{Y(max\backslash min) - Y0}{((X(max\backslash min) - Xstart)(X(max\backslash min) - Xend))}$

$Y = Y0 + a(X - Xstart)(X - Xend)$</td><td>$a = \frac{0.3 - 0}{(-0.0140 + 0.8)(-0.0140 - 0.26)}$

= -1.398093001

$Y = -1.398093\ (X\ + 0.8)(X - 0.26)$</td></tr>
<tr><td rowspan="3">Example 4.2</td><td colspan="2">From table A.1, Brick number 2 (gap shift and Xplace)</td></tr>
<tr><td>By Formula</td><td>By values</td></tr>
<tr><td>Gap shift = gap shift previous + 0.01

Xplace = Xplace previous + gap shift</td><td>Gap shift = 0.075 + 0.01 = 0.085

Xplace = -0.8 + 0.085 = -0.725</td></tr>
<tr><td rowspan="3">Example 4.3</td><td colspan="2">From table A.1, Brick number 2 (Yplace)</td></tr>
<tr><td>By Formula</td><td>By values</td></tr>
<tr><td>$Y = Y0 + a(X - Xstart)(X - Xend)$</td><td>$Y = -1.398093\ (-0.725 + 0.8)(-0.725 - 0.26)$

= 0.102907171</td></tr>
</table>

Example 4.4	From table A.2, column1, Brick number 2	
	By Formula	By values
	Xpick = Xpickprevious + 0 = Xpick Ypick = Ypickprevious + 0 = Ypick Zpick = Zpickprevious - Zgap = Zpick	Xpick = 0.875 + 0 = 0.875 Ypick= -0.8 + 0 = -0.8 Zpick= 0.075 - 0.05 = 0.025
Example 4.5	From table A.2, column1, Brick number 3	
	By Formula	By values
	Xpick = Xpickprevious + 0 = Xpick Ypick = Ypickprevious + Ygap = Ypick Zpick = Zpickprevious + Zga= Zpick	Xpick = 0.875 + 0 = 0.875 Ypick= -0.8 + 0.1 = -0.7 Zpick= 0.025 + 0.05 = 0.075
Example 4.6	From table A.2, column 2, Brick number 17	
	By Formula	By values
	Xpick = Xpickprevious + Xgap = Xpick Ypick = Yfirstbrick= Ypick Zpick = Zpickprevious + Zgap = Zpick	Xpick = 0.875 + 0.15 = 1.025 Ypick= -0.8 = -0.8 Zpick= 0.075 + 0.05 = 0.075

Table A.4 Error and Accuracy analysis

Example 4.7	Error and Accuracy percentage (Brick Number 1 table 4.12)	
	By Formula	By values
	Error $= \left\lvert \frac{\text{Desired X coordinate} - \text{Actual X coordinate}}{\text{Desired X coordinate}} \right\rvert$	Error $= \left\lvert \frac{-0.8 - (-0.7965)}{-0.8} \right\rvert$ Error = 0.004375
	%Error= Error * 100%	%Error = 0.004375 * 100 %Error = 0.4375 %
	%Accuracy = 100 * (1 - Error)	%Accuracy = (1 – 0.004375) * 100% %Accuracy = 99.5625
Example 4.8	Error and Accuracy percentage (Brick Number 8 table 4.13)	
	By Formula	By values
	$Error = \left\lvert \frac{Desired\ X\ coordinate - Actual\ X\ coordinate}{Desired\ X\ coordinate} \right\rvert$	$Error$ $= \left\lvert \frac{-0.332750599 - (-0.3472)}{-0.332750599} \right\rvert$ = 0.04342411722
	%Error= Error * 100%	%Error = 0.004375 * 100 %Error = 4.342411722 %

	%Accuracy = 100 * (1 - Error)	%Accuracy = (1 – 0.043424117) * 100% %Accuracy = 95.6575883
Example 4.9	RMSE calculation for X coordinate	
	By Formula	By values
	$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(X_{Theoretical} - X_{Actual})^2}{n}}$	$\sum_{i=1}^{n}(X_{Theoretical} - X_{Actual})^2 =$ $(0.4375/100)^2 + (0.537931/100)^2 + (0.15625/100)^2 + (0.825688073/100)^2 + (0.909090909/100)^2 + (1.446153846/100)^2 + (3.75/100)^2 + (2.769230769/100)^2 + (20.16666667/100)^2 + (18.34285714/100)^2 + (22.46428571/100)^2 = 0.00527952$ $RMSE = \sqrt{\frac{0.00527952}{11}}$ RMSE = 0.02190791

Table A.5: Arc length and the curvature of the curved wall.

Example 4.10	The arc length of the desired curved wall	
	By Formula	By values
	$Y = Y0 + a(X - Xstart)(X - Xend)$	$Y = -1.398(X + 0.8)(X - 0.259)$
	$(\frac{\mathrm{dy}}{\mathrm{dx}})^2 = (\frac{\mathrm{d}(-1.398(X+0.8)(X-0.259))}{\mathrm{dx}})^2$	$(\frac{dy}{dx})^2 = (20796186\, X + 0.7563683)^2$
	$L = \int_{X0}^{Xf} \sqrt{1 + (\frac{dy}{dx})^2}\, dx$	$L = \int_{-0.8}^{0.259} \sqrt{1 + (2.796186\, X + 0.7563683)^2}\, dx$ L = 1.36944466014268698233045754893 m
Example 4.11	The curvature of the desired curved wall (if desired)	
	By Formula	By values
	$r(t) = (x, y, z)$	$\mathrm{r(t)} = (\mathrm{x}, -1.398093(\mathrm{t} + 0.8)\ (\mathrm{t} - 0.259), 0.025)$
	$\dot{r}(t) = (\dot{x}, \dot{y}, \dot{z})$	$\dot{r}(t) = (1\ , -2.796186\, t\ -\ 0.756368, 0)$
	$\ddot{r}(t) = (\ddot{x}, \ddot{y}, \ddot{z})$	$\ddot{r}(t) = (0, -2.796186002292872, 0)$

	$\dot{r}(t)\ X\ \ddot{r}(t) = (\dot{x} + \dot{y} + \dot{z})\ X\ (\ddot{x} + \ddot{y} + \ddot{z})$	$\dot{r}(t)\ X\ \ddot{r}(t) = (1\ , -2.796186\,t - 0.756368, 0)\ X\ (0, -2.796186, 0)$ = (0,0, -2.7961860023)
	$\lvert\dot{r}(t)\ X\ \ddot{r}(t)\rvert = \sqrt{(\dot{r}(t)\ X\ \ddot{r}(t)^2)}$	$\lvert\dot{r}(t)\ X\ \ddot{r}(t)\rvert = \sqrt{(0 + 0 + (-2.796186002293)^2)}$ = 7.8186561594185960961892331961573
	$\lvert\dot{r}(t)\rvert^3 = \left\lvert\sqrt{(\dot{x}^2 + \dot{y}^2 + \dot{z}^2}\right\rvert^3$	$\lvert\dot{r}(t)\rvert^3 = \sqrt{(1 + ((-2.796186t - 0.75636831362)^2) + 0)}$
	$k = \frac{\lvert\dot{r}(t)\ X\ \ddot{r}(t)\rvert}{\lvert\dot{r}(t)\rvert^3}$	$k = \frac{7.8186561594185960961892331961573}{\sqrt{(1 + ((-2.796186t - 0.75636831362)^2) + 0)}}$

APPENDIX B

MATLAB codes

The MATLAB code to find the arc length of the desired curved wall.

```
% This code is used to calculate the arc length of the desired curved wall
% brick layer (Done by Mohammed Ahmed Salem B011310375)
%________________________________________________________________________

% 1) initial conditions
fprintf('Initial conditions');
syms x;
startpointx=-0.8
maxpointy=0.3
maxpointx=-0.0140
endpointx=0.2590

% 2) calculate the factor a
a = (maxpointy/((maxpointx-startpointx)*(maxpointx-endpointx)))

% 3) writh the parabolic equation
fprintf('The parabolic equation is \n');
y = a*((x-startpointx)*(x-endpointx));
pretty(y)

% 4) Derive the parabolic equation;
fprintf('The derevitive of The parabolic equation is \n');
derivitive = diff(y,x);
pretty(derivitive);

% 5) Find the square root
fprintf('The square root is \n');
power = derivitive^2;
pretty(power);
addone = 1 + power;
```

```
pretty(addone);
squareroot = sqrt(addone);
pretty(squareroot);

% 6) Integrate the square root
fprintf('The arc length is \n');
length = int(squareroot,x,startpointx,endpointx);
pretty(length);
vpa(length)
%______________________________________________________________________
% To find the estimated brick numbers (EBN)
EBN=(length/(0.075+0.1))+ 5
```

The MATLAB code to find the curvature of the desired curved wall.

```
% This code is used to calculate the curvature of the desired curved wall
% brick layer (Done by Mohammed Ahmed Salem B011310375)
%______________________________________________________________________

% 1) initial conditions
format long g
fprintf('Initial conditions');
syms x;
startpointx=-0.8;
maxpointy=0.3;
maxpointx=-0.0140;
endpointx=0.2590;
z=0.025;

% 2) calculate the factor a
a = (maxpointy/((maxpointx-startpointx)*(maxpointx-endpointx)))

% 3) writh the parabolic equation
fprintf('The parabolic equation is \n');
yd = a*((x-startpointx)*(x-endpointx));
y=vpa(yd)
```

```
% 4) write the parabolick eqation as a vector equation
r = [x;y;z];
%r = [x;-x^2;-(x^3)/4];
pretty(r)

% 5) find the first derivative of the vector
d1=diff(r);
pretty(d1)

% 6) find the second derivative of the vector
d2=diff(d1);
pretty(d2)

% 7) Find ||r'||^3
d12 = d1.^2;
d1plus=d12(1)+d12(2)+d12(3);
pretty(d1plus)
d1square=sqrt(d1plus);
pretty(d1square)
d1power3=(d1square)^3;
pretty(d1power3)

% 8) Find ||r' x r''||
C = cross(d1,d2);
pretty(C)
C=C.^2;
pretty(C)
Cplus= (((C(1)))+((C(2)))+((C(3))));
Csquareroot=sqrt(Cplus);
pretty(Csquareroot)

% 9) Find The curvature equation (k = ||r' x r''|| / ||r'|| ^3)
curvature=(Csquareroot)/(d1power3);
fprintf('\nThe curvature equation \n');
pretty(curvature)
fprintf('\nOr\n');

pretty(vpa(curvature))

% 10) Find The curvature at specific place
```

```
x = -0.08;
fprintf('\nThe curvature is \n');
l=subs(curvature);
(vpa(l))
%____________________________________________________________________________
```

APPENDIX C

Detailed simulation Setup

1. Write the parabolic equation as a code in MATLAB, then plot the position (X and Y) graph, to prove that the parabolic equation generates the desired curve trajectory.

2. Start the V-REP PRO EDU, version 3.3.0 (rev.1) or any updated version, as shown in figure 3.10.

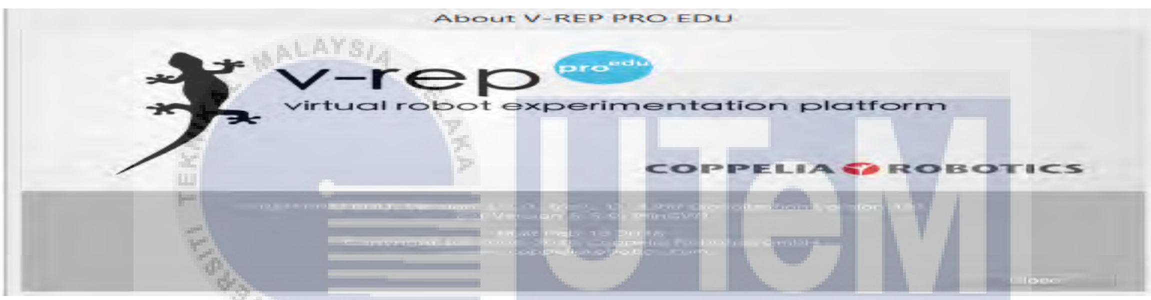

Figure C.1: V-REP PRO EDU, version 3.3.0 (rev.1)

3. Create the V-REP environment by select the suitable mobile manipulator (KUKA YouBot) by cling on Model browser button, then click on robots, then click on mobile, then select KUKA YouBot as shown in figure 3.11.

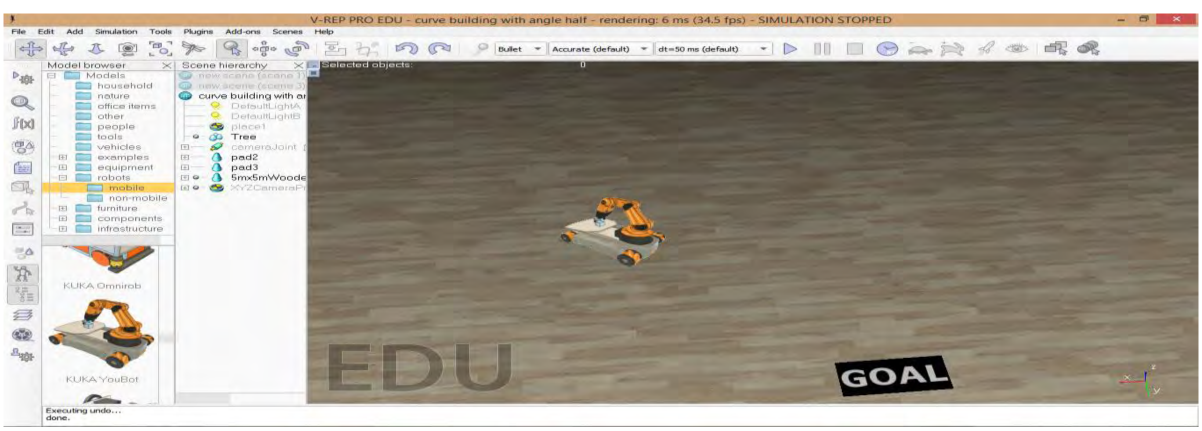

Figure C.2: KUKA YouBot.

4. Start to construct and add the desired bricks number based on the standard or desired dimensions and density (standard dimensions are recommended which are (112.5 mm, 225 mm and 75 mm)). First at all click on the add button, then chose primitive shape, then chose cuboid finally choose the dimensions and the density. Figure 3.12 and 3.13 illustrates the process respectively.

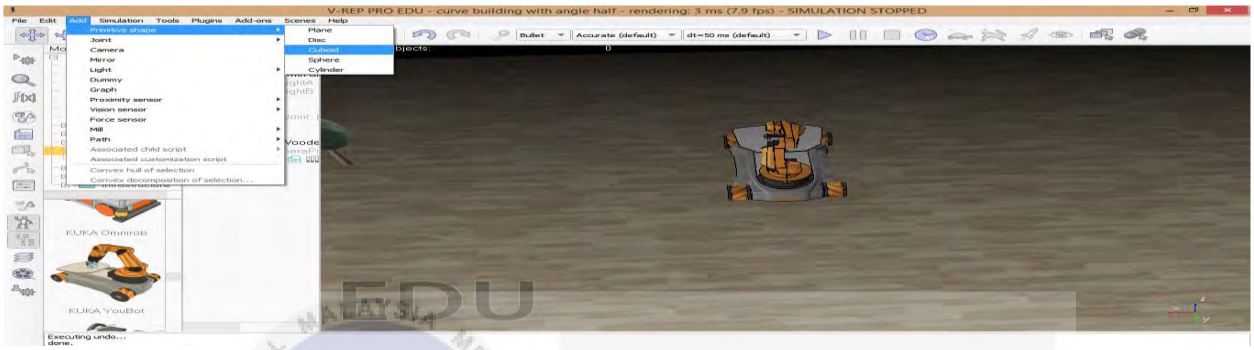

Figure C.3: Construct the Bricks.

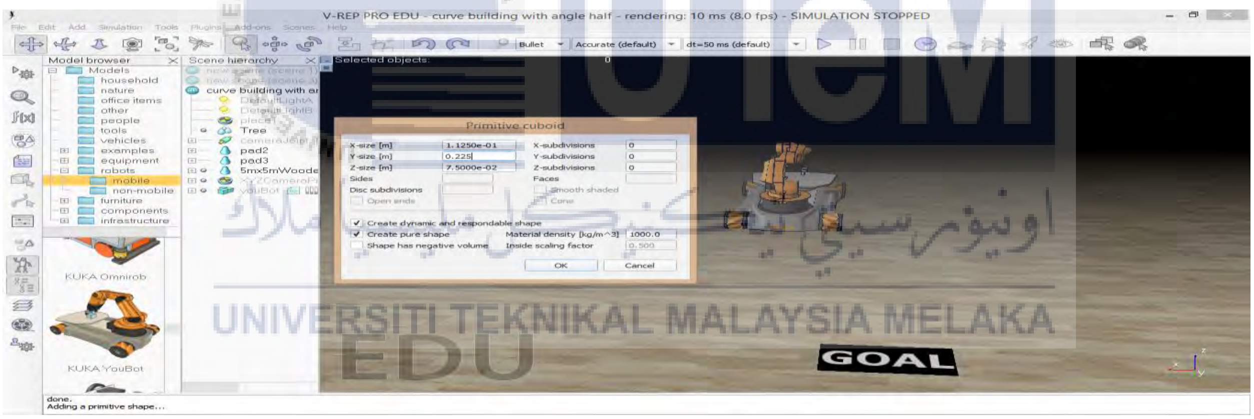

Figure C.4: Construct the Bricks.

After adding the. brick, move it to the initial picking coordinates (X0, Y0 and Z0), then use the gap equations for the picking area to put the bricks in an understandable pattern foo the KUKA manipulator as shown in figure 3.14.

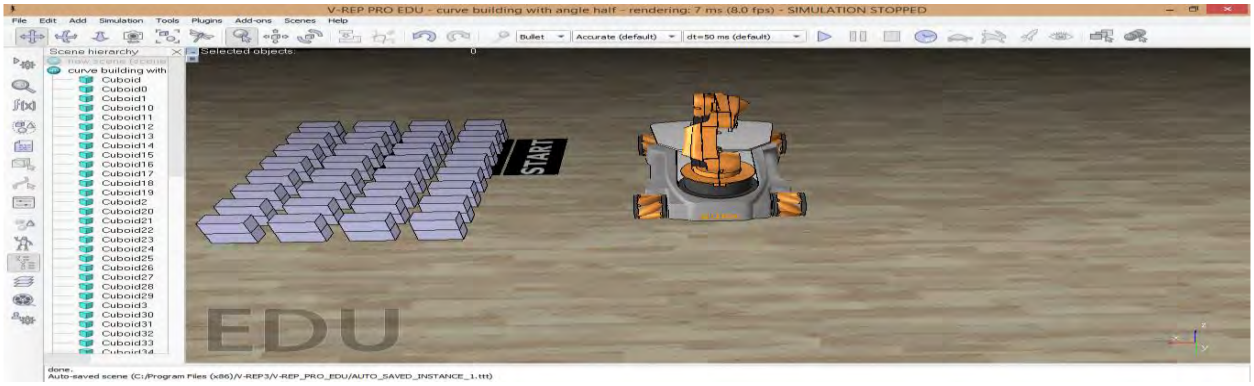

Figure C.5: Move bricks into picking area.

5. To get the motion profiles of the five degree of freedom (joints), the following steps is done.

5.1 click add, then click graph, then click add again and then click floating view. Figures 3.15 and 3.16 illustrate the processes respectively.

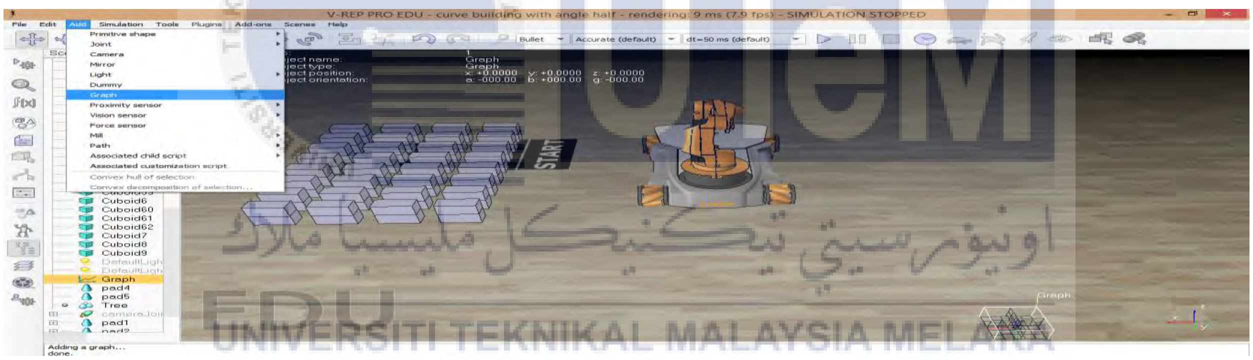

Figure C.6: Adding floating point.

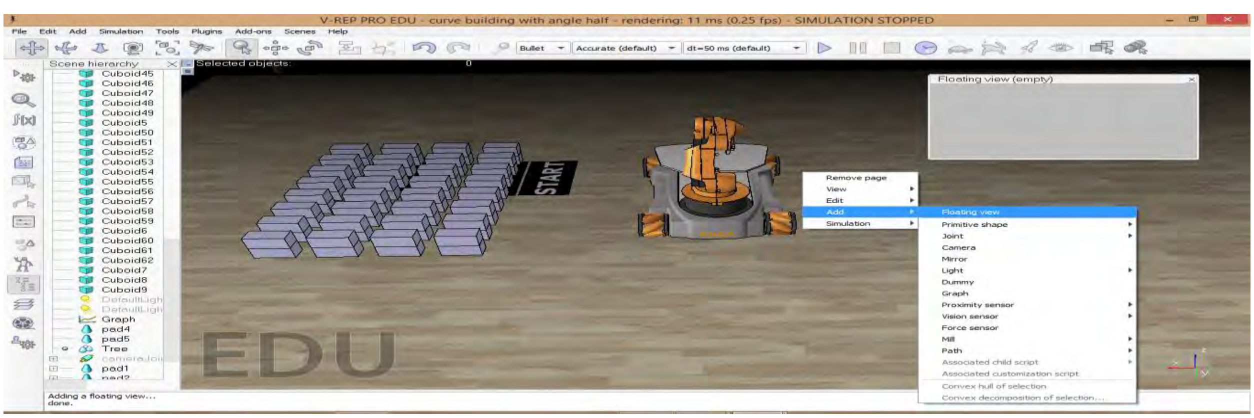

Figure C.7: Adding floating point.

5.2 Double click on the graph, then click add new data stream to record, then at data stream type it is optional to choose, here the joint position is selected. At the object / item to record the interested joint is selected. In this process we can add more than one joint by clicking on add new data stream to record again. Figure 3.17 illustrate the process.

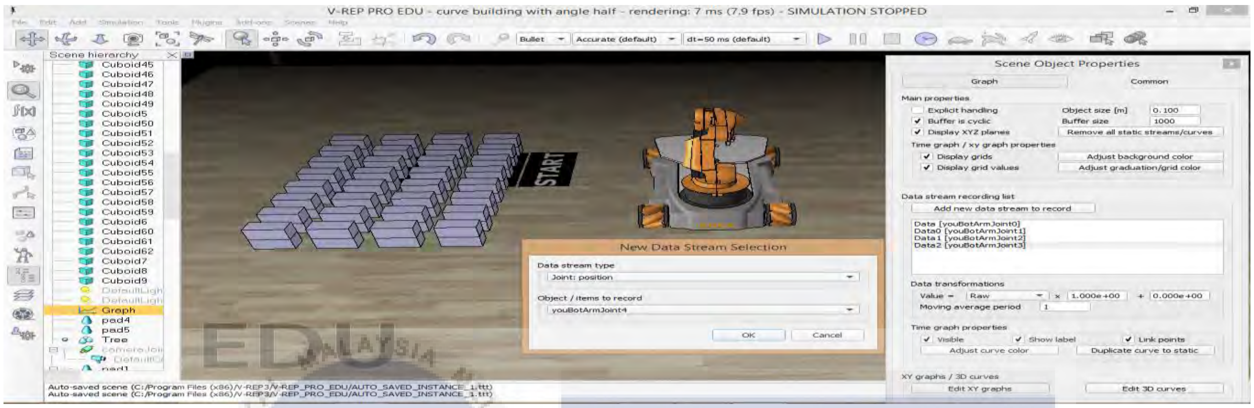

Figure C.8: Add desired data stream to record.

5.3 Now right click on the floating view, then click view, then click associate view with selected graph. Now the graph is ready to record the selected data. Figure 3.18 illustrate the process.

Figure C.9: Associate the view with selected graph.