“I hereby declare that I have read through this report entitled “HIGH SPEED 3D PRINTING STRATEGY USING 6 DOF NON-MOBILE ROBOT MANIPULATOR BASED ON SHORTEST DISTANCE ALGORITHM” and found that it complies the partial fulfillment for awarding the degree of Bachelor of Electrical Engineering”

Signature : ……………………………………………………………………

Supervisor’s Name : PM DR MUHAMMAD FAHMI BIN MISKON

Date : ……………………………………………………………………

HIGH SPEED 3D PRINTING STRATEGY USING 6 DOF NON-MOBILE ROBOT MANIPULATOR BASED ON SHORTEST DISTANCE ALGORITHM

BEE SHENG CHONG

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

Faculty of Electrical Engineering
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2018

I declare that this report entitled "HIGH SPEED 3D PRINTING STRATEGY USING 6 DOF NON-MOBILE ROBOT MANIPULATOR BASED ON SHORTEST DISTANCE ALGORITHM" is the result of my own research except as cited in the references. The report has not been accepted for any degree and is not concurrently in candidature of any other degree.

Signature : ....

Name : BEE SHENG CHONG

Date : ....

To my beloved mother and father

ACKNOWLEDGEMENT

Firstly, I would like to express my sincere appreciation to my FYP supervisor PROFESOR MADYA DR. MUHAMMAD FAHMI BIN MISKON for his patience and kindness in guiding me on this project. His passion and sense of responsibility had help me to keep the progress of the project at a satisfactory level. Besides that, I would also very grateful to his professional comment and critics which greatly promote my understanding on the project.

Meanwhile, I sincerely thank you to both of my FYP panels, DR. MOHD SHAHRIEEL BIN MOHD ARAS and Pn NURDIANA BINTI NORDIN for the generous enlightenment they gave to me on the shortcomings and weakness of my works after the presentation. The suggestions they gave strengthen my report and make it more complete.

Lastly, I am very thankful for all the supports and helps which came from my course mates and friends. Their attitudes of always willing to help and fast respond resolved most of my confused moment during the work for this project. I would also like to thank my family for their caring and tolerant to me during this tough time.

ABSTRACT

3D printing is an increasing demanded technology nowadays and the 3D printers sold on the mass market today are limited on its workspace. In order to solve the problem, a strategy that combining non-mobile articulated robot and 3D printing is introduced in this project. The main focus in this project is to study, design and evaluate the shortest distance algorithm in order to achieve high speed 3D printing that using 6 DOF non-mobile robot manipulator. The Dijkstra's algorithm is referred in this project for determine the shortest distance travelled from coordinate to coordinate. The performance of the shortest distance algorithm is examined and analysed by using Scilab. Meanwhile, the performance of the manipulator is simulated in V-REP and corresponding analysis is carried out to measure the speed of the 3D printing. The result shows that the designed algorithm able to shorten the distance travelled by percentage ratio of 72.57%. On the other hand, time complexity of the designed algorithm is $O(n^2)$ while 3D printing analysis result indicates that the algorithm can increase the efficiency of the 3D printing but the starting coordinate may affect the efficiency of the 3D printing caused by different trajectory path.

ABSTRAK

Pencetakan 3D adalah teknologi yang semakin menuntut pada masa kini dan pencetak 3D yang dijual di pasaran massa hari ini terhad pada ruang kerjanya. Untuk menyelesaikan masalah ini, satu strategi yang menggabungkan artikulasi robot bukan mudah alih dengan percetakan 3D diperkenalkan dalam projek ini. Tumpuan utama dalam projek ini adalah untuk mengkaji, merekabentuk dan menilai algoritma jarak terpendek untuk mencapai pencetakan 3D berkelajuan tinggi yang menggunakan 6 DOF robot bukan mudah alih. Algoritma Dijkstra dirujuk dalam projek ini untuk menentukan jarak terpendek dari koordinat ke koordinat. Prestasi algoritma jarak terpendek diperiksa dan dianalisis dengan menggunakan Scilab. Sementara itu, prestasi manipulator disimulasikan dalam V-REP dan analisis yang sama dijalankan untuk mengukur kelajuan percetakan 3D. Hasilnya menunjukkan bahawa algoritma tersebut dapat memendekkan jarak perjalanan dengan nisbah peratusan sebanyak 72.57. Di samping itu, kerumitan masa algoritma yang direka adalah $O(n^2)$ manakala hasil analisis percetakan 3D menunjukkan bahawa algoritma dapat meningkatkan kecekapan pencetakan 3D tetapi koordinat permulaan mungkin mempengaruhi kecekapan pencetakan 3D disebabkan oleh laluan trajektori yang berbeza.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATION

SP – Shortest Path

RP – Rapid Prototyping

DOF – Degree Of Freedom

CAD – Computer-Aided Drawing

3D – Three-Dimensional

dist – Distance

STL – STereoLithography

AM – Additive Manufacturing

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Motivation

Nowadays, 3D printing becomes a trending technology due to the impact it had brought to rapid prototyping [1]. Many people believe that it will be widespread and become more common in future because of those benefits it brought to us. For example, a prototype can be created rapidly in just a few hours by 3D printing technique compared with traditional manufacturing methods which taken days or even weeks to receive a prototype[2]. Due to the advantage of 3D printing on saving cost and time, mitigate risks, able to create complex products and many more, this technology is increasing demanded on various field, such as, architecture and construction, healthcare and medical, mechanic, aeronautics and space, and so on [1].

However, 3D printer on the market nowadays cannot create a large enough prototype because of its limited workspace and size. For example, the 3D printer produced by CreatBot named as ‘D600’ can only print the largest build volume of 600*600*600*mm and it’s already 30 times larger than it of other ordinary 3D printer in today mass market. Obviously, this is one of the limitation of 3D printer nowadays. Hence, if we can implant 3D printing technology on 6 DOF non-mobile robot manipulator system we may overcome this limitation.

Besides that, 3D printing using 6 DOF non-mobile robot manipulator system possess a great advantage compared with the traditional additive manufacturing method. Firstly, conventional method is restricted by both the gravity and printing environment[3]. Meanwhile, the uses of 6 DOF robot makes the 3D printing on irregular, or non-horizontal surfaces become possible. The robot can change the orientation of extrusion for the best strength and appearance of a part during 3D printing.

Lastly, the robot is using shortest distance algorithm to analyze and decide the shortest path for the 3D printing process. This energy optimizing technique reduces the time and energy consumption by the robot to the lowest as possible. Consequently, it makes the construction of light-weighted, high mobility and longer energy lasting 3D printing robot become possible.

1.2 Problem statement

The first problem of high speed 3D printing strategy using 6 DOF non-mobile robot manipulator system by using shortest distance algorithm is to design a shortest distance algorithm that can determine the sequence of the coordinates to achieve high speed 3D printing.

The designed algorithm should be able to minimize the travelled path in order to increase the speed of 3D printing. Next, the problem is to code the algorithm to the computer and conduct analysis on its performance.

The next problem is the complexity of trajectory generation for the robot. The complexity of trajectory generation of the robot is due to its multiple degree of freedom [4]. The robot has to find the way to reach a specific point and it got many ways to get a same point due to its multiple degree of freedom [5].

The performance of the robot is demonstrated and observed firstly by simulation using V-rep. This is also one of the problems we concerned about as how to simulate these trajectories and represent them in the computer.

Based on the problems described, the project need to address a method on how to achieve high speed 3D printing with different shape and dimension.

1.3 Objective

The objectives of this project are:

1. To analyse the shortest path problem of 3D printing that using a 6 DOF manipulator (IRB 4600 industrial robot).
2. To design a shortest path algorithm which can generate corresponding trajectory based on the shape and dimension given.
3. To evaluate the efficiency and speed of the 3D printing that using IRB 4600 industrial robot.

1.4 Scope

The scope of this project concentrated on:

1. Study on the shortest path algorithm to achieve a high-speed 3D printing with 6 DOF non-mobile robot manipulator system.
2. Use Scilab simulator to demonstrate the performance of the algorithm.
3. Use V-rep simulator to simulate the 3D printing process.
4. 3D printed a U-shape box but not exceed the workspace limit of the robot.
5. Use IRB 4600-40-255 industrial robot to conduct the 3D printing.
6. Represented the 3D printed object by octree and the dimensions of the object is acted as references.
7. To find the efficiency of the algorithm by using method of analysis.
8. The trajectory path that generated by the algorithm.
9. Assuming no obstacles and external factors in the simulation environment.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter is intended to review, synthesises, analyse and present the information of previous research works from literature related to 3D printing using robot. Nevertheless, the shortest path algorithm which is used to optimize the performance of the robot and resulted in high speed 3D printing is prioritized in this chapter. In line with this, a few shortest path algorithms such as Dijkstra’s algorithm, Bellman Ford's algorithm and Floyd–Warshall's algorithm will be introduced and compared in the coming section. A most suitable SP algorithm is selected as reference with the appropriate reason provided. Then, time complexity is introduced for the analysis of algorithm. Subsequently, the chapter is focus on 3D printing and technology related to it. The 3D printing and its process are enlightened in the following section. Related subjects such as STL files, slicing software and G-code are presented and explained along with 3D printing process. The information of existing 3D printing robot is gathered and shown in the later section.

2.2 Shortest path algorithm based on distance comparison

Shortest path algorithm is widely implement in various field especially in internet addressing computing, intelligent transportation systems, urban geographic information systems, and military geographical information systems. In this project, shortest distance algorithm is the main focus which is also the biggest problem urged to be solved. At present, there are several types of numerical algorithm available for solving optimization problem. Here are their names: Dijkstra algorithm, Bellman Ford's algorithm and Floyd–Warshall's algorithm.

2.2.1 Dijkstra's algorithm

Dijkstra's algorithm is an algorithm used to find the shortest distance between nodes in a graph. It was devised by computer scientist Edsger W. Dijkstra in 1956 and published 3 years later [6].

Dijkstra's algorithm applies only if [7]:

i. The link values (edges cost) must be positive values (the algorithm will be broken if the value is negative) but the links can be directional
ii. All vertices in the graph is connected (the algorithm does not work if there is unconnected part exists)

The steps to apply Dijkstra's algorithm are shown below [8]:
The starting node is called as the initial node.

1. An uncertain distance value is assigned to every node: initial node is set as zero and other nodes are set as infinity.
2. Place the initial node as current and set all other nodes as unvisited. Make the unvisited set which is a set of all the unvisited nodes.
3. Consider all the neighbors of current node and compute their tentative distances. The current assigned value will be replaced by newly calculated tentative distance if the value is smaller. For example, if the assigned value is 7 for current node A and the connected edge to neighbor node B has a length of 3, then the distance to B will be 7 + 3 = 10. If B was previously assigned as a value greater than 10 then it will be changed to 10. Otherwise, the value will be remained.
4. When all the neighbors of the current node are done considering, the current node is marked as visited and it will be removed from unvisited set. A visited node will not be visit again.
5. The algorithm will continue until the destination node has been marked visited or the smallest tentative distance among the nodes in the unvisited sets is infinity.

6. If none of the two conditions is achieved, the unvisited node with the smallest tentative distance will be selected and set as new 'current node' then repeat from step 3.

Figure 2.1 & 2.2 and Table 2.1 show the demonstration of the algorithm.

'A' is the starting node.

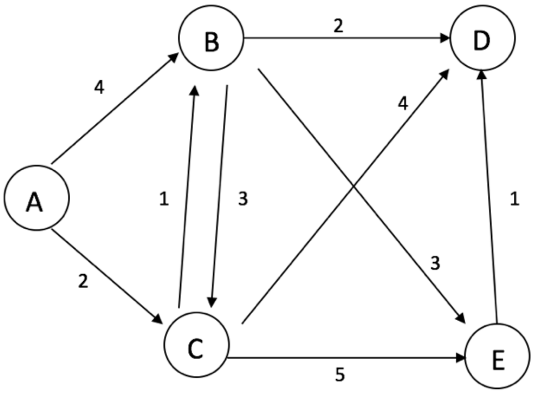

Figure 2.1: Five vertices shortest path problem example.

Table 2.1: Result of Dijkstra's algorithm on solving Figure 2.1 example.

Step	Current node	Unvisited set	A	B	C	D	E
1	A	{B,C,D,E}	0	4	2	∞	∞
2	C	{B,D,E}	0	3	2	6	7
3	B	{D,E}	0	3	2	5	6
4	D	{E}	0	3	2	5	6
5	E	{}	0	3	2	5	6

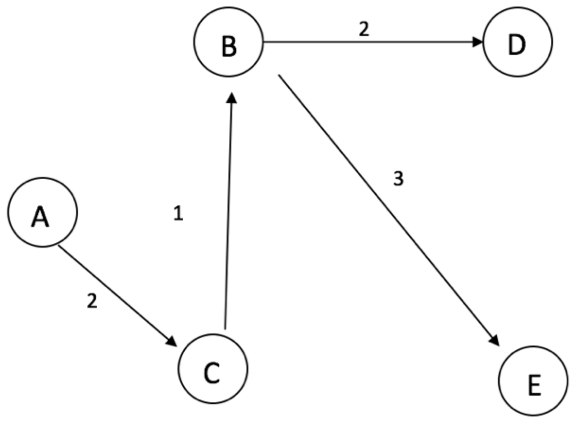

Figure 2.2: The shortest path of the Figure 2.1 example.

2.2.2 Bellman-Ford's algorithm

Bellman-Ford's algorithm is similar to Dijkstra's algorithm but it is more versatile because it can work with graph in which the edges can have negative weights [7]. This algorithm uses the principle of relaxation which is same as Dijkstra's algorithm but the only difference is Bellman-Ford algorithm relaxes all the edges instead of choosing the nearest vertex that has not been checked (also known as greedy strategy). The ordering of its relaxation makes the algorithm can handles negative weights and also detects negative cycles. It is important to note that if negative cycle occurs, there will be no shortest path exists on the graph.

Below is the step on how the algorithm works:

1. Allocate the starting node as zero and the rest of the vertices are set as infinity.
2. Start from the first point (starting node), give all the neighbors of the node a tentative distance cost.
3. Continue with the next node, the node which cannot be defined on the moment is skipped.
4. After all nodes are checked once, the first iteration is done.
5. Repeat from step 2 until 4, update the tentative distance cost when there is a smaller value get in this process.

6. The process will stop when there is no change compared with previous iteration (The algorithm takes at most |V – 1| iteration but it can be ended earlier if no alteration is found on the next iteration).

The algorithm is demonstrated on Figure 2.3 and Table 2.2 as below.
'S' is the starting node

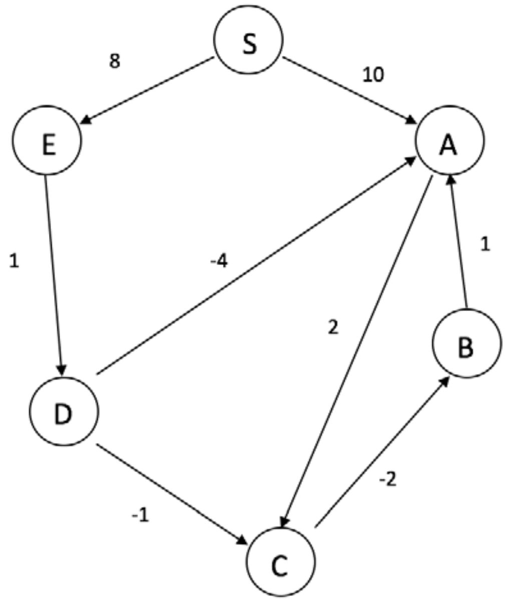

Figure 2.3: Six vertices graph with negative edge weights example.

Table 2.2: Result of Figure 2.2 solved by Bellman-Ford's algorithm.

Iteration	S	A	B	C	D	E
0	0	∞	∞	∞	∞	∞
1	0	10	10	12	9	8
2	0	5	10	8	9	8
3	0	5	5	7	9	8
4	0	5	5	7	9	8

2.2.3 Floyd-Warshall algorithm

Floyd-Warshall algorithm is an algorithm for finding shortest distance in a weighed graph same as Dijikstra's algorithm and Bellman-Ford's algorithm [7]. However, it is an all-pairs shortest path algorithm instead of single-source shortest path algorithm (Dijkstra's algorithm & Bellman-Ford's algorithm). This means that it computes the shortest distance between every pair of vertices in the graph instead of starting from a single initial node. Floyd-Warshall algorithm is an example of dynamic programming, the problem is break down into smaller sub-problems and the answers of those sub-problems are combined to solve the large, primary problem. It can solve the graph contains negative weighed edges but no negative-weighed cycles (same as Bellman-Ford algorithm). It is useful at handling multiple stops on the route because it can compute the shortest distance between all relevant nodes.

The implementation of Floyd-Warshall algorithm is shown below.

1. A distance array table is constructed to track the shortest path between nodes.
2. Fill in the corresponding weighs into the table by looking at the edges between the nodes of the graph.
3. Replace the value on the table by the newly calculated value based on equation (2.1) if the condition is met;

   $$dist[i][j] > dist[i][k] + dist[k][j] \quad (2.1)$$

   i , j and k are value from 1 until number of vertices in the graph, eg. if the graph contains 4 vertices then;

   i = 1 2 3 4 ; j = 1 2 3 4 ; k = 1 2 3 4
4. The algorithm is done as all values of i, j and k have been calculated once.

The example of Floyd-Warshall algorithm is shown in Figure 2.4 below.

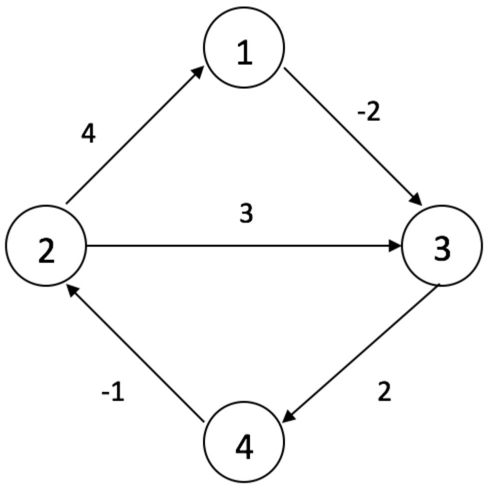

Figure 2.4: Example of four vertices graph with negative edge weights.

Step 1: Constructs a table and fill in the corresponding edges weights. The result is shown in Table 2.3.

Table 2.3: Result of the Step 1 of Floyd-Warshall algorithm.

	1	2	3	4
1	0		-2	
2	4	0	3	
3			0	2
4		-1		0

Step 2: Computes them by using the formula with different value of i, j and k

Let k = 1, i = 1 and j = 2,

dist[i][j] > dist[i][k] + dist[k][j]

dist[1][2] > dist[1][1] + dist[1][2]

$\infty > 0 + \infty$

$\infty > \infty$ (False, nothing happened)

Let k = 1, i = 2 and j = 3,

dist[i][j] > dist[i][k] + dist[k][j]

dist[2][3] > dist[2][1] + dist[1][3]

3 > 4 + (-2)