

DEVELOPMENT OF CONTROLLER
SOFTWARE SYSTEM FOR CAM
MANUFACTURING MACHINE

MOGAN A/L ARASAPPAN

BACHELOR OF MECHANICAL ENGINEERING
UNIVERSITI MALAYSIA PAHANG

DEVELOPMENT OF CONTROLLER SOFTWARE SYSTEM FOR CAM
MANUFACTURING MACHINE

MOGAN A/L ARASAPPAN

Report submitted in partial fulfillment of the
requirements for the award of the degree of
Bachelor of Mechanical Engineering

Faculty of Mechanical Engineering
UNIVERSITI MALAYSIA PAHANG

NOVEMBER 2009

EXAMINER APPROVAL DOCUMENT

**UNIVERSITI MALAYSIA PAHANG
FACULTY OF MECHANICAL ENGINEERING**

We certify that the project entitled “DEVELOPMENT OF CONTROLLER SOFTWARE SYSTEM FOR CAM MANUFACTURING MACHINE” is written by MOGAN A/L ARASAPPAN. We have examined the final copy of this project and in our opinion; it is fully adequate in terms of scope and quality for the award of the degree of Bachelor of Engineering. We herewith recommend that it be accepted in partial fulfillment of the requirements for the degree of Bachelor of Mechanical Engineering.

Examiner: EN. MOHD. HAFIZI B. ZUHARI

Signature:

SUPERVISOR'S DECLARATION

I hereby declare that I have checked this project and in my opinion, this project is adequate in terms of scope and quality for the award of the degree of Bachelor of Mechanical Engineering.

Signature:

Name of Supervisor: EN. MOHD RUZAIMI BIN MAT REJAB

Position: LECTURER OF FACULTY OF MECHANICAL ENGINEERING

Date: 23/11/2009

STUDENT'S DECLARATION

I hereby declare that the work in this project is my own except for quotations and summaries which have been duly acknowledged. The project has not been accepted for any degree and is not concurrently submitted for award of other degree.

Signature:

Name: MOGAN A/L ARASAPPAN

ID Number: MA06043

Date: 23/11/2009

**Dedicated to my beloved parents,
brothers and sister**

ACKNOWLEDGEMENTS

I would like to express my sincerely gratitude to my supervisor, EN. MOHD. RUZAIMI BIN MAT REJAB, for constant guidance and providing useful information for me on my research from the beginning. I also would like to thank him for spending most of his precious time to help me with correcting my errors and all the help he has offered to finish this thesis.

My sincere thanks to ASSOC. PROF. DR. WAN AZHAR BIN WAN YUSOFF for his supports and provides me essential information on completing the program of this research. I also would like to thank him for allowing me to refer to his previous works and studies about program to develop cam profile.

Many special thanks go to I would also like to thank to all my course-mates and staffs of the Mechanical Engineering Department, UMP, who helped me in many ways and made my stay at UMP pleasant and unforgettable.

I acknowledge my sincere indebtedness and gratitude to my parents, MR. ARASAPPAN A/L SINNATHAMBY and MDM. AMMRATHAN A/P MARIMUTHU for their love, dream and sacrifice throughout my life. They were always stood by me and help me to motivate myself to overcome any obstacles. Lastly, I would like to thank my elder brother, YOGESVARAN A/L ARASAPPAN, my elder sister, SARASVATHY DEVI A/P ARASAPPAN and my younger brother, TIRUPPATHY A/L ARASAPPAN for their continuous supports and trusts.

ABSTRACT

This thesis is about developing controller software system for cam manufacturing machine by using cam parametric equations as a guide. Cam development is based on the cam shapes, follower types, and also the applications. Performance of a cam profile affects the speed of the system and the higher the speed, the system will experience more vibrations which lead to defectiveness of the system. These vibrations also may occur due to sudden acceleration of the cam surface once they were produced with unsmooth surface. In order to produce highly performance of cam, it has to have a smoothest surface of contact with the follower. The objective of this project is divided to two most important criteria which to derive a parametric equation system for both x and y axis from a relative cam parametric equation of motion and also to simulate a cam profile using parametric equation of motion by programming methods. The results of this project are from both graphical and programming methods and the results of each cam motions will be discussed. Finally, for conclusion and recommendation of this project, we will conclude the results and whether the objective of this project is achieved and we will also recommend little recommendation on how to improve the results obtain for further research to come.

ABSTRAK

Tesis ini adalah mengenai pengembangan sistem perisian pengontrol mesin produksi cam dengan menggunakan persamaan cam parametrik sebagai panduan. Pembinaan cam adalah berdasarkan kepada bentuk cam, jenis pengikut, dan juga aplikasi. Kelancaran profil cam mempengaruhi kelajuan sistem dan pada kelajuan yang lebih tinggi, sistem akan mengalami lebih banyak getaran yang menyebabkan kecacatan kepada sistem. Getaran ini juga boleh berlaku kerana percepatan permukaan cam secara tiba-tiba dan akan terhasil dengan permukaan yang tidak sempurna. Untuk menghasilkan kelancaran kualiti, cam itu harus mempunyai permukaan licin yang bertemu dengan pengikut. Objektif dari projek ini dibagi kepada dua kriteria yang paling penting untuk memperoleh sistem persamaan parametrik untuk kedua-dua paksi x dan y dari persamaan parametrik cam relatif gerak dan juga untuk mensimulasikan profil cam menggunakan persamaan parametrik gerak mengikut kaedah pengaturcaraan. Keputusan daripada projek ini adalah daripada kaedah grafik dan pengaturcaraan dan hasil dari keputusan masing-masing, gerakan cam akan dibahas. Akhirnya, kesimpulan dan cadangan untuk projek ini, kami akan menyimpulkan keputusan dan menyimpulkan tujuan dari projek ini sudah tercapai dan kami juga akan mengesyorkan sedikit cadangan mengenai bagaimana memperbaiki memperoleh hasil untuk kajian lebih lanjut yang seterusnya.

TABLE OF CONTENTS

		Page
SUPERVISOR’S DECLARATION		ii
STUDENT’S DECLARATION		iii
DEDICATION		iv
ACKNOWLEDGEMENTS		vi
ABSTRACT		vii
ABSTRAK		viii
TABLE OF CONTENTS		viii
LIST OF TABLES		xi
LIST OF FIGURES		xii
LIST OF SYMBOLS		xiv
LIST OF ABBREVIATIONS		xv
CHAPTER 1 INTRODUCTION		
1.1	Introduction of Cam	1
1.2	Project Background	2
1.3	Problem Statement	2
1.4	Project Objectives	3
1.5	Project Scopes	3
CHAPTER 2 LITERATURE REVIEW		
2.1	Introduction	4
2.2	Cam Definition	5
2.3	Cam Mechanisms	5
	2.3.1 Different types of Cam	6
	2.3.2 Cam Shape	6
	2.3.3 Follower Configurations	11
	2.3.4 Follower Types	11
2.4	Cam Nomenclature	13

2.5	Cam Motions	14
	2.5.1 Constant Velocity Motion	15
	2.5.2 Constant Acceleration Motion	16
	2.5.3 Simple Harmonics Motion	17
	2.5.4 Cycloidal Motion	18
2.6	Graphical Layout of Cam Profile	19
2.7	Programming Methods	20

CHAPTER 3 METHODOLOGY

3.1	Introduction	21
3.2	Cam Motion Concept	21
	3.2.1 Cam Parametric Equation	22
3.3	Case Study	25
	3.3.1 Example	25
3.4	Flow Chart	28
3.5	Advantages of Using Cam Motion Equation	29
	3.5.1 Comparison Between Computer Aided Design and CAM Application	29

CHAPTER 4 RESULTS AND DISCUSSION

4.1	Introduction	30
4.2	Graphical Method	30
	4.2.1 Graphical Results	31
4.3	Programming Method	36
	4.3.1 C++ Programming	36
	4.3.2 Programming Results	36

CHAPTER 5 CONCLUSION AND RECOMMENDATIONS

5.1	Introduction	42
5.2	Conclusions	42
5.3	Recommendations	44
	REFERENCES	45
	APPENDICES	
A1	Cam Profile Graphical Result for Constant Velocity Motion	47
A2	Cam Profile Graphical Result for Constant Acceleration Motion	48
A3	Cam Profile Graphical Result for Simple Harmonic Motion	49
A4	Cam Profile Graphical Result for Cycloidal Motion	50
A5	Cam Profile Graphical Result for Combined Motion	51
B1	Program for Constant Velocity Motion	53
B2	Program for Constant Acceleration Motion	56
B3	Program for Simple Harmonics Motion	59
B4	Program for Cycloidal Motion	62
B5	Program for Combined Motion	65

LIST OF TABLES

Table No.	Title	Page
3.1	Spreadsheet for Graphical Cam Layout	27
4.1	Data for Constant Velocity Motion	31
4.2	Data for Constant Acceleration Motion	32
4.3	Data for Simple Harmonics Motion	33
4.4	Data for Cycloidal Motion	34
4.5	Data for Combined Motion	35

LIST OF FIGURES

Figure No.	Title	Page
2.1	Plate cam or disc cam	6
2.2	Groove cam or closed cam	7
2.3	Cylindrical cam or barrel cam	7
2.4	End cam	8
2.5	Wedge cam	8
2.6	Spiral cam	9
2.7	Conjugate cam	9
2.8	Globoidal cam	10
2.9	Spherical cam	11
2.10	Knife-edge follower	12
2.11	Roller follower	12
2.12	Flat-faced follower	12
2.13	Spherical-faced follower	12
2.14	Offset Roller Follower	12
2.15	Swinging Roller Follower	12
2.16	Cam nomenclature	13
2.17	Series of Motion events	15
2.18	Cam Graphical Layout	20
3.1	Cam graphical layout using constant velocity motion	27
3.2	Cam graphical layout using constant velocity motion	28
4.1	Graph of Constant Velocity Motion	31
4.2	Graph of Constant Acceleration Motion	32

4.3	Graph of Simple Harmonic Motion	33
4.4	Graph of Cycloidal Motion	34
4.5	Graph of Combined Motion	35
4.6	Program result of Constant Velocity Motion	37
4.7	Program result of Constant Acceleration Motion	38
4.8	Program result of Simple Harmonics Motion	39
4.9	Program result of Cycloidal Motion	40
4.10	Program result of Combined Motions	41

LIST OF SYMBOLS

β	Rotation angle of cam during the rise or fall interval under consideration (deg)
θ	Cam rotation angle measured against the direction of cam rotation from the home position
π	Pi
ω	Speed of the cam (degrees per time)
ϕ	Angle into rise or fall interval (deg)
a	Instantaneous follower acceleration
a	Constant variables
b	Constant variables
C_x	x coordinate of cutter center
C_x'	Differential x coordinate of cutter center
C_y	y coordinate of cutter center
C_y'	Differential y coordinate of cutter center
H	Total follower displacement during the rise or fall interval under consideration
ΔR	Instantaneous follower displacement
v	Instantaneous follower velocity

LIST OF ABBREVIATIONS

MEMS	Micro-Electromechanical Systems
CAD	Computer Aided-Design
CAM	Computer Aided-Manufacturing
RDR	rise-dwell-rise
DRFD	dwell-rise-fall-dwell
RFR	rise-fall-rise

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION OF CAM

A cam is generally known as an element of the cam-follower mechanical system that reciprocates (move back and forth) when direct contact with another element known as the follower thus creates a prescribed motion. Nevertheless, cam is properly design and manufacture for variety of applications, especially in requirement for high speed operation that enhances the accuracy, precision and repeatability.

Cam can be found in almost all mechanical devices and machines which can be seen in domestic industries of cam-follower machineries. In yearly year, the genius of Leonardo da Vinci was able to produce a modern design cam which applied to a machine for pumping water. This technically inspires much new application using the same concept. The obvious prove can be seen in oil pump at oil rig refinery that been used by many largest oil company nowadays.

Even so, the application of cam-follower mechanism system are however evolves into more convenient cam-driven machines such as automobile internal combustion engines, transportation equipment, and most recently the Micro-Electromechanical Systems (MEMS).

These applications are basically based on simple cam profile motion, where it consist series of events which are rise, dwell and return. Rise is a motion when the follower starts to move away from cam center, dwell is a motion where the follower is at rest, and return is when the motion of the follower moving towards closer to the cam center.

1.2 PROJECT BACKGROUND

The development of cam software system for computer resources are widely used in today mechanical equipment application and has contributed to understanding the cam-follower system complete design. Most manufacturer of industrial machinery has developed proprietary software packages for their specific needs. This project co-consists of developing a controller software system where it has high potential to be used for any cam manufacturing machine and reduce time consumed for a cam to be manufacture if compared to other cam designing software such as Computer Aided-Design (CAD) or Computer Aided-Manufacturing (CAM).

1.3 PROBLEM STATEMENT

Cam mechanism system is developed by combining two mutually related elements which is the cam and follower. Cam development is based on the cam shapes, follower types, and also the applications. Cam shapes determine how fast or slow do the speed the follower will achieve and does the performance of cam-follower system satisfied with the application. Performance of a cam-follower system affects the speed of the system and the higher the speed, the system will experience more vibrations which lead to defectiveness of the system. These vibrations also may occur due to sudden acceleration of the cam surface once they were produced with unsmooth surface. In order to produce highly performance of cam, it has to have a smoothest surface of contact with the follower. This will be a problem for engineer or designers, because they need to reduce sudden acceleration on the surface of the cam.

1.4 PROJECT OBJECTIVES

This project sets a couple of objectives to be covered which are:

- i. To derive a parametric equation system for both x and y axis from a relative cam parametric equation of motion.
- ii. To simulate a cam profile using parametric equation of motion by programming methods.

1.5 PROJECT SCOPES

The scopes of this project are:

- i. Analyze the equation of motion of cam for constant-velocity motion, constant acceleration motion, and simple harmonics motion and cycloidal motion.
- ii. Express and perform differential equation to basic cam parametric equations. Perform derivation analysis on the equations of motions to get parametric equation system of x and y axis.
- iii. Apply the parametric equation in a source code for C++ programming to produce a cam profiles.
- iv. Compare the results between cam profiles from graphical results and programming results

CHAPTER 2

LITERATURE REVIEW

2.1 INTRODUCTION

In this chapter, the details and the basic knowledge about the concept and mechanism of a cam profile will be explained. The contents and information of this chapter are mainly come from reference books, journals, thesis, and shared views from online resources which will be included in references section. In this particular section, relevance knowledge concern about cam mechanism and its cam-follower system will be discussed.

On the further part of this chapter, mostly every part concerning cam mechanism will be thoroughly explained such as types of cam and follower, cam nomenclature, and different type of cam motion. Moreover, important aspects also will be covered such as cam parametric equation, and cam coordinate systems. The whole view of this chapter is to provide a well, better and clearer understanding of the purpose of this project.

2.2 CAM DEFINITION

Cams are used to convert rotary motion into reciprocating motion. The motion created can be simple and regular or complex and irregular. As the cam turns, it driven by the circular motion that gives a predetermined specified motion to the follower. The cam follower traces the surface of the cam transmitting its motion to the required mechanism. A cam mechanism usually consists of two moving elements, the cam and the follower, mounted on a fixed frame. [1]

A cam may be defined as an element of cam-follower mechanical system that compels the movement of the follower by direct contact. The cam has been widely used in variety type of application includes almost all mechanical devices and machinery properties such as in transportation equipment, machine tools, printing process, control systems, food processing machines, automobile internal combustion engines, and even in agriculture. The accuracy of the cam produced is frequently improved and the cost is lowered after each new design are invented or innovated. [2]

2.3 CAM MECHANISMS

We can classify cam mechanisms by the shape of the cam, the series of events of motion, the configuration and arrangement of the follower. Firstly, we can determine cam in terms of their shapes, such as wedge, radial, cylindrical, globoidal or even conical. Also, we can classify them in terms of the follower motion as the result of direct contact, such as rise-dwell-rise (RDR), dwell-rise-fall-dwell (DRFD) or rise-fall-rise (RFR). Finally, in terms of the follower constraint, this can be achieved by either positive drive or spring load. [3]

2.3.1 Different Types of Cam

There are many different types of cam existed nowadays and each of cam developed have unique way of showing their differences especially in terms of shapes, sizes and also the motion that they produce to the follower. The cam can be design in variety of shapes, and the most popular industrial production cams are [4]:

- i. Positive drive – radial groove cam
- ii. Radial conjugate dual cam
- iii. Cylindrical groove cam
- iv. Cylindrical with circumferential contour cut in the end (end cam)

2.3.2 Cam Shape

i. Plate cam (Figure 2.1):

This is the simplest and very common type of cam. It is formed on a disc or a plate. The circumference of the cam varies the radial distance and the center of the disc. The follower will ride on the outer edge and results a radial motion. Thus, this cam can be recognized as radial cam. [5]

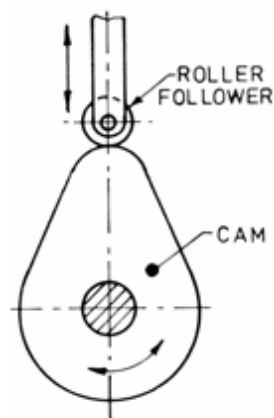


Figure 2.1: Plate cam or disc cam

ii. Grooved cam or closed cam (Figure 2.2):

This cam was basically used in sewing machine. Cutting this cam requires that the cutter be moved in the exact way relative to the spherical cam as the roller follower. [6]

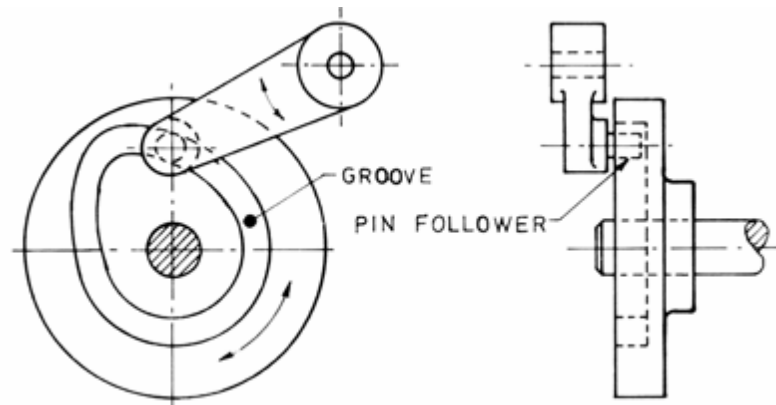


Figure 2.2: Groove cam or closed cam

iii. Cylindrical cam (Figure 2.3):

This important criterion is that this cam is formed on a cylinder. A groove is cut on the cylinder and varies for each axis of rotation. This is to give the follower motion along the axis of rotation. [7]

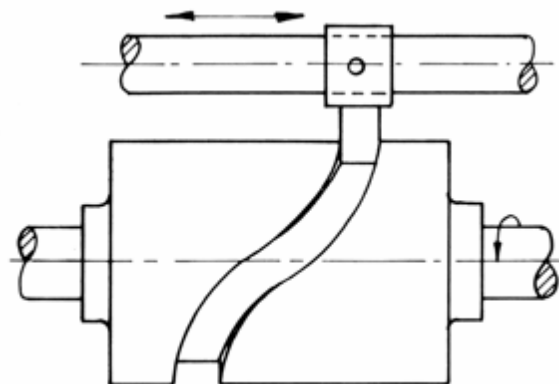


Figure 2.3: Cylindrical cam or barrel cam

iv. **End cam (Figure 2.4):**

Similar with cylinder cam but the follower contact with the cam on different periphery. The cam rotates, while the follower will translates or oscillates. [8]

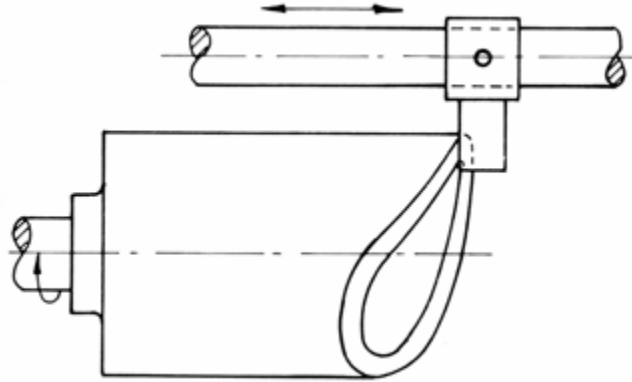


Figure 2.4: End cam

v. **Wedge or flat cam (Figure 2.5)**

A wedge cam has a wedge in general which has translational motion. The follower can either translate or oscillate. A spring is used to maintain contact between the cam and follower. Instead of wedge a flat plate with a groove is also used.

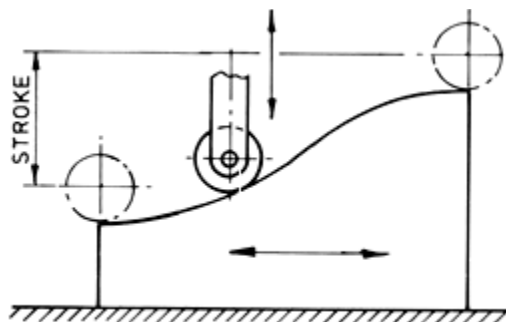


Figure 2.5: Wedge cam

vi. Spiral cam (Figure 2.6)

A spiral cam is a face cam in which a groove is cut in form of spiral. The spiral grooves consist of teeth, which mesh with a pin gear follower. The velocity of the follower is proportional to the radial distance of groove from axis of cam.

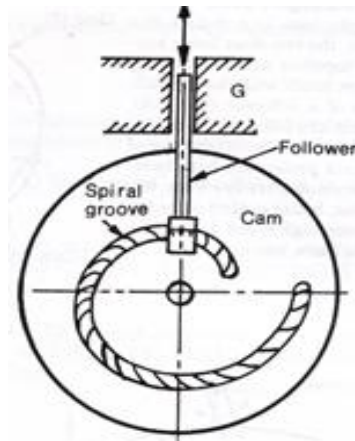


Figure 2.6: Spiral cam

vii. Conjugate cam (Figure 2.7)

A conjugate cam is a double disc cam, two discs being keyed together and is in constant touch with two rollers of the follower.

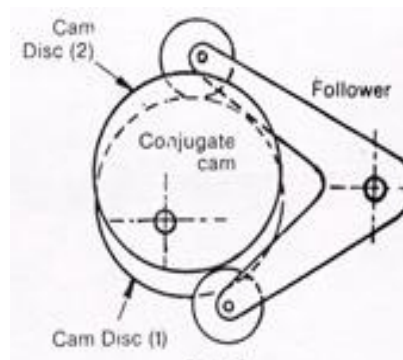


Figure 2.7: Conjugate cam

viii. Globoidal cam(Figure 2.8)

A Globoidal cam can have two types of surfaces, convex or concave. A circumferential contour is cut on the surface of the rotation of cam to impart motion to the follower that has an oscillatory motion.

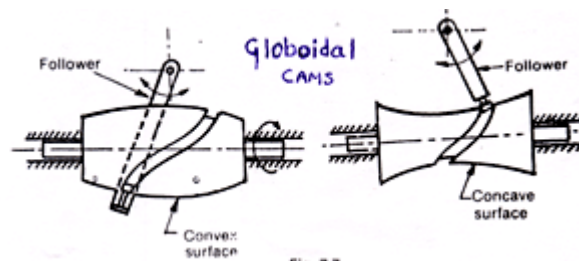


Figure 2.8: Globoidal cam

ix. Spherical cam (Figure 2.9)

In this the follower oscillates about an axis perpendicular to the axis of rotation of the cam.

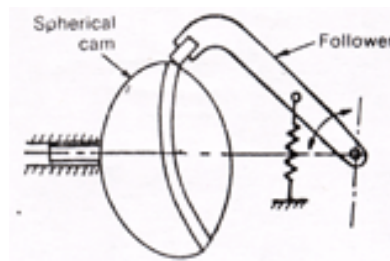


Figure 2.9: Spherical cam

2.3.3 Follower Configuration

Followers can be classified into several ways:

- i. Follower motion (translation or oscillation)
- ii. Translating follower motion with reference of cam center may be radial or offset
- iii. Shape of the follower contact surface

2.3.4 Follower Types

i. Knife-edge follower (Figure 2.10)

This is the simplest form of follower because it is formed to a point and drags on the edge of the cam. Consequently it is rarely used because the sharp edges produce high stresses and results in excessive wear of the contact point.

ii. Roller follower (Figure 2.11)

This type of follower has two elements which is the roller and the follower itself. The roller is pinned to the follower stem, and when the cam rotates the roller will roll on the surface of the cam. It is frequently used on machinery production as it has less friction and does not wear-off rapidly.

iii. Flat-faced follower (Figure 2.12)

This type has a flat surface that contact with the cam. It suitable to used with a steep motion and does not jam. Therefore, it is preferred to be used for quick motions. The follower has higher stresses than the roller follower because of intense sliding contact with the cam.

iv. Spherical-faced follower (Figure 2.13)

This type of follower has a radius face that contact with the cam. It has similar uses as flat-faced follower where it can be use for steep motion without jamming. Consequently, it has greater friction force than roller follower.

v. Offset Roller Follower (Figure 2.14)

The center of this follower does not pass through the center of the cam center.

vi. Swinging Roller Follower (Figure 2.15)

This cam can allow much more pressure angle. [8]

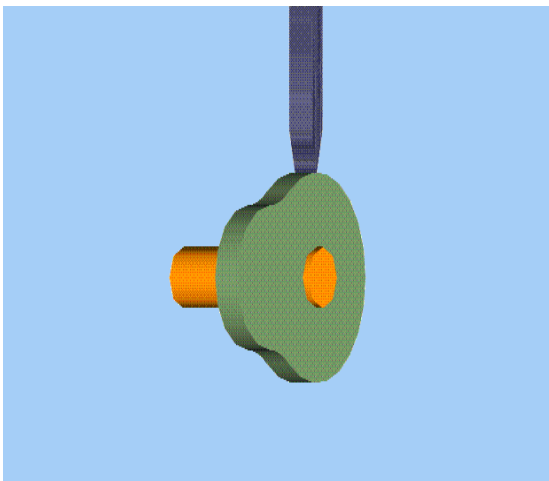


Figure 2.10: Knife-edge follower

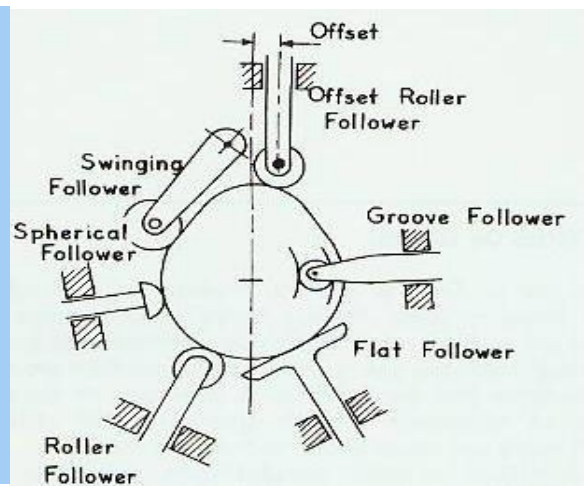


Figure 2.11, 2.12, 2.13, 2.14, and 2.15

2.4 CAM NOMENCLATURE

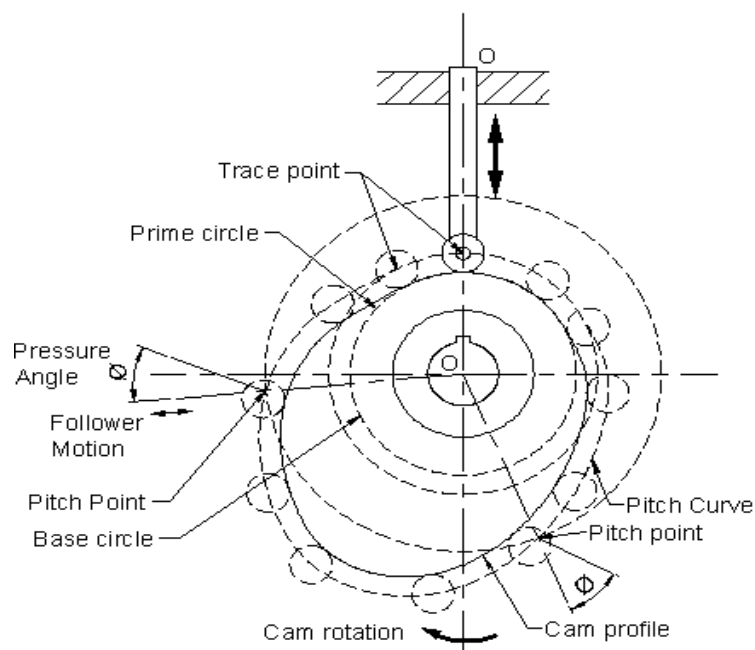


Figure 2.16: Cam nomenclature [2]

Notations:

- i. Trace point: It is a point on the follower, and its motion describes the movement of the follower. It is used to generate the pitch curve.
- ii. Pitch curve: The path generated by the trace point as the follower is rotated about a stationary cam.
- iii. Working curve: The working surface of a cam in contact with the follower. For the knife edge follower of the plate cam, the pitch curve and the working curves coincide. In a close or grooved cam there is an inner profile and an outer working curve.
- iv. Pitch circle: A circle drawn from the cam center and passes through the pitch point is called Pitch circle.
- v. Prime circle or reference circle: smallest circle from the cam center through the pitch curve.
- vi. Base circle: The circle with the shortest radius from the cam center to any part of the cam profile.

- vii. Stroke or throw: The greatest distance or angle through which the follower moves or rotates.
- viii. Follower displacement: The position of the follower from a specific zero or rest position in relation to time or the rotary angle of the cam.
- ix. Pressure angle: The angle between the direction of the follower movement and the normal to the pitch curve.

2.5 CAM MOTIONS

As the cam rotates the follower moves upward and downward. The upward motion of follower is called rise and the downward motion is called fall. When the follower is stationary or not moving upward and downward even when the cam rotates, it is called dwell. [2]

In general there are four types of motion of follower.

- i. Constant velocity motion
- ii. Constant acceleration motion
- iii. Simple harmonic motion
- iv. Cycloidal motion

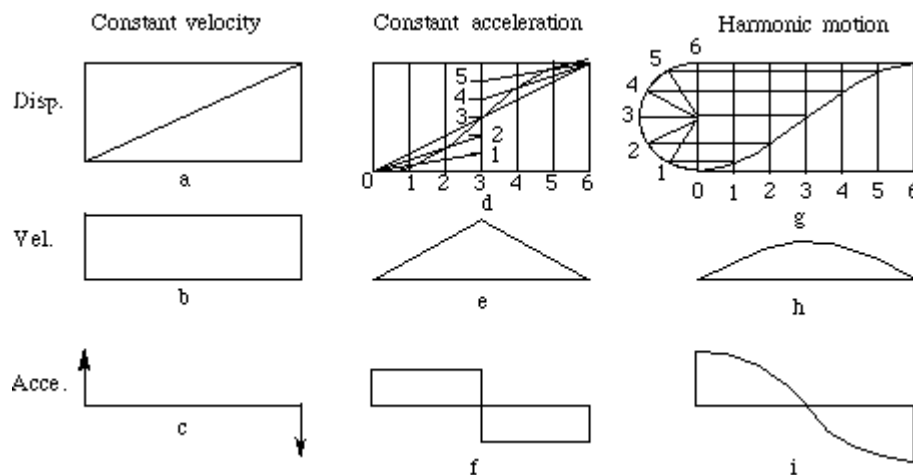


Figure 2.17: Series of Motion events

Notation:

Φ : The rotary angle of the cam (measured when motion events begin);

β : The interval of the rotary angle corresponding to the motion event;

ΔR : Instantaneous displacement of follower;

H : Total lift respects to follower motion;

v : Velocity of the follower;

a : Acceleration of the follower;

2.5.1 CONSTANT VELOCITY MOTION

This is the simplest form follower motion during rise and fall events. It is characterized as a straight line displacement diagram due to its uniform velocity (a). Constant velocity diagram is shown on (b). In theory, instantaneous jump from any constant value of velocity to another will results in infinite acceleration. The acceleration diagram is shown on (c).

Rise

$$\Delta R = H \frac{\phi}{\beta}$$

$$v = H \frac{\omega}{\beta}$$

$$a = 0$$

Fall

$$\Delta R = H \left(1 - \frac{\phi}{\beta} \right)$$

$$v = -H \frac{\omega}{\beta}$$

$$a = 0$$

2.5.2 CONSTANT ACCELERATION MOTION

This type of motion of follower is occurred during the rise and fall events that cause acceleration for a given rise and time interval [5]. The velocity diagram of this motion is shown on (e) while the acceleration diagram is shown on (f).

When $0 \leq \phi \leq \frac{\beta}{2}$,

Rise

$$\Delta R = H - 2H \left(1 - \frac{\phi}{\beta}\right)^2$$

$$v = 4H \frac{\omega}{\beta} \left(1 - \frac{\phi}{\beta}\right)$$

$$a = -4H \left(\frac{\omega}{\beta}\right)^2$$

Fall

$$\Delta R = H - 2H \left(1 - \frac{\phi}{\beta}\right)^2$$

$$v = 4H \frac{\omega}{\beta} \left(1 - \frac{\phi}{\beta}\right)$$

$$a = 4H \left(\frac{\omega}{\beta}\right)^2$$

When

$$\frac{\beta}{2} \leq \phi \leq \beta$$

Rise

$$\Delta R = H - 2H \left(1 - \frac{\phi}{\beta}\right)^2$$

$$v = 4H \frac{\omega}{\beta} \left(1 - \frac{\phi}{\beta}\right)$$

$$a = -4H \left(\frac{\omega}{\beta}\right)^2$$

Fall

$$\Delta R = 2H \left(1 - \frac{\phi}{\beta}\right)^2$$

$$v = -4H \frac{\omega}{\beta} \left(1 - \frac{\phi}{\beta}\right)$$

$$a = 4H \left(\frac{\omega}{\beta}\right)^2$$

2.5.3 SIMPLE HARMONICS MOTION

Simple harmonic motion is derived from trigonometric function and results a very smooth motion curves. It is actually the projection motion of a point on a rotating disc projected to a straight line. The velocity diagram is shown on (g) and the acceleration diagram is shown on (h).

$$0 < \Phi < \beta,$$

Rise

$$\Delta R = \frac{H}{2} \left(1 - \cos \frac{\pi\phi}{\beta} \right)$$

$$v = \frac{H}{2} \left(\frac{\pi\omega}{\beta} \right) \left[\sin \left(\frac{\pi\phi}{\beta} \right) \right]$$

$$a = \frac{H}{2} \left(\frac{\pi\omega}{\beta} \right)^2 \left[\cos \left(\frac{\pi\phi}{\beta} \right) \right]$$

Fall

$$\Delta R = \frac{H}{2} \left(1 + \cos \frac{\pi\phi}{\beta} \right)$$

$$v = -\frac{H}{2} \left(\frac{\pi\omega}{\beta} \right) \left[\sin \left(\frac{\pi\phi}{\beta} \right) \right]$$

$$a = -\frac{H}{2} \left(\frac{\pi\omega}{\beta} \right)^2 \left[\cos \left(\frac{\pi\phi}{\beta} \right) \right]$$

2.5.4 CYCLOIDIAL MOTION

This is another derived motion from trigonometric equation. It is preferred motion for high speed motion because it has very smooth motion curves and does not have sudden change of acceleration in the end of motion. Physically, it is a motion of a point on a disc rolling on a straight line.

Rise

$$\Delta R = H \left[\left(\frac{\phi}{\beta} \right) - \left(\frac{1}{2\pi} \right) \sin \left(\frac{2\pi\phi}{\beta} \right) \right]$$

$$v = \left(\frac{H\omega}{\beta} \right) \left[1 - \cos \left(\frac{2\pi\phi}{\beta} \right) \right]$$

$$a = H\pi \left(\frac{\omega}{\beta} \right)^2 \left[\sin \left(\frac{2\pi\phi}{\beta} \right) \right]$$

Fall

$$\Delta R = H \left[1 - \left(\frac{\phi}{\beta} \right) + \left(\frac{1}{2\pi} \right) \sin \left(\frac{2\pi\phi}{\beta} \right) \right]$$

$$v = - \left(\frac{H\omega}{\beta} \right) \left[1 - \cos \left(\frac{2\pi\phi}{\beta} \right) \right]$$

$$a = -H\pi \left(\frac{\omega}{\beta} \right)^2 \left[\sin \left(\frac{2\pi\phi}{\beta} \right) \right]$$

2.6 GRAPHICAL CAM LAYOUT PROFILE

For every cam design, cam profiles are laid out graphically using inversion principle. That is, the cam is held stationary and the successive follower positions. The cam profile is the filled in as the envelope curve of the follower positions. [2] The important parameters need to be consider while lay-outing the cam graphically are [6];-

- i. Selecting the base circle radius of the cam and follower
- ii. Pressure angle of the cam – the angle between the contact normal and the velocity of the point on the follower at the contact position.
- iii. The size (length) of the follower face – the face has to be long enough on both sides to maintain contact with the cam on a tangent line.

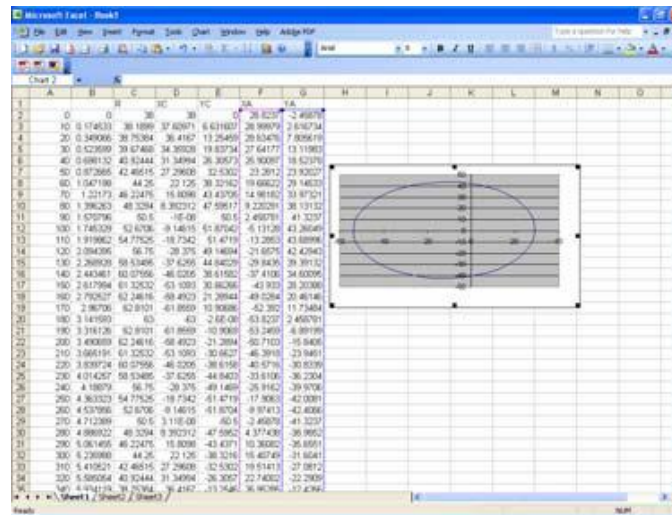


Figure 2.18: Cam Graphical Layout

2.7 PROGRAMMING METHODS

Cam controller machine uses cam algorithm to convert arbitrary equation into motion on the cutter. The software will relate and translate the motion of control to data of motion. Motion of cutter will determine the shape of cam that will be produced. [1]

i. Cam Synthesis Using Software (Programming Tools)

This method uses a series of equation of motions which define overall part trajectory positions and the series of motions in the program. The program reads the data as cutter location and consequently the cutter will follow the cutter location generated by the program. [2]

ii. Computer Aided Machining Systems (CAM)

This method has been use in almost any cam manufacturing machine. The CAM uses the source or model created in Computer Aided Design (CAD) and translates the model into trajectory motion for the cutter to follow. The cutter will automatically follow the sequence of shape that been developed in the CAM. [2]

CHAPTER 3

METHODOLOGY

3.1 INTRODUCTION

In this chapter, the methods use for this project will be clearly indicated. In developing controller software system of cam manufacturing machine, the most important criterion that needs to be considered is to understand cam motion's equations. The cam motion includes constant velocity motion, constant acceleration motion, simple harmonic motion and also cycloidal motion. This basic motion related to cam-follower mechanism need to be express into cam parametric equation in order to use as input to be able to translate into computational data for programming method. The objective is to derive the cam motion into parametric equation and test the equation by using programming method. The result is then compared and discussed with the graphical method.

3.2 CAM MOTION CONCEPT

Cam motion can basically be classified into four types of motion which are constant velocity motion, constant acceleration motion, simple harmonics motion, and cycloidal motion [3]. These four motions can determine the smoothness of the cam profile surface.

3.2.1 Cam Parametric Equation

Cam parametric equation is the equation of motion for cam-follower mechanism motion when it experience a prescribed motion due to result of direct contact between the cam and the follower, whereas the coordinate of motion of the cutter can be relate as below;

i. Constant velocity motion

Rise

$$Cx = a \sin \theta + b \sin \theta + \frac{H\theta}{\beta} \sin \theta$$

$$Cx' = a \cos \theta + b \cos \theta + \frac{H\theta}{\beta} \cos \theta + \frac{H}{\beta} \sin \theta$$

$$Cy = a \cos \theta + b \cos \theta + \frac{H\theta}{\beta} \cos \theta$$

$$Cy' = -a \sin \theta - b \sin \theta - \frac{H\theta}{\beta} \sin \theta + \frac{H}{\beta} \cos \theta$$

Fall

$$Cx = a \sin \theta + b \sin \theta + H \sin \theta - \frac{H\theta}{\beta} \sin \theta$$

$$Cx' = a \cos \theta + b \cos \theta + H \cos \theta - \frac{H\theta}{\beta} \cos \theta - \frac{H}{\beta} \sin \theta$$

$$Cy = a \cos \theta + b \cos \theta + H \cos \theta - \frac{H\theta}{\beta} \cos \theta$$

$$Cy' = -a \sin \theta - b \sin \theta - H \sin \theta + \frac{H\theta}{\beta} \sin \theta - \frac{H}{\beta} \cos \theta$$

ii. Constant acceleration motion

When, $0 \leq \phi \leq \frac{\beta}{2}$

Rise

$$Cx = a \sin \theta + b \sin \theta + 2H \left[\frac{\theta}{\beta} \right]^2 \sin \theta$$

$$Cx' = a \cos \theta + b \cos \theta + 2H \left[\frac{\theta}{\beta} \right]^2 \cos \theta + \left[\frac{4H\theta}{\beta^2} \right] \sin \theta$$

$$Cy = a \cos \theta + b \cos \theta + 2H \left[\frac{\theta}{\beta} \right]^2 \cos \theta$$

$$Cy' = -a \sin \theta - b \sin \theta - 2H \left[\frac{\theta}{\beta} \right]^2 \sin \theta + 4H \left[\frac{\theta}{\beta^2} \right] \cos \theta$$

Fall

$$Cx = a \sin \theta + b \sin \theta + H \sin \theta - 2H \left[\frac{\theta}{\beta} \right]^2 \sin \theta$$

$$Cx' = a \cos \theta + b \cos \theta + H \cos \theta - 2H \left[\frac{\theta}{\beta} \right]^2 \cos \theta - \left[\frac{4H\theta}{\beta^2} \right] \sin \theta$$

$$Cy = a \cos \theta + b \cos \theta + H \cos \theta - 2H \left[\frac{\theta}{\beta} \right]^2 \cos \theta$$

$$Cy' = -a \sin \theta - b \sin \theta - H \sin \theta + 2H \left[\frac{\theta}{\beta} \right]^2 \sin \theta - \left[\frac{4H\theta}{\beta^2} \right] \cos \theta$$

When, $\frac{\beta}{2} \leq \theta \leq \beta$

Rise

$$Cx = a \sin \theta + b \sin \theta + H \sin \theta - 2H \sin \theta + \left[\frac{4H\theta}{\beta} \right] \sin \theta - 2H \left[\frac{\theta}{\beta} \right]^2 \sin \theta$$

$$Cx' = a \cos \theta + b \cos \theta + H \cos \theta - 2H \cos \theta + \left[\frac{4H\theta}{\beta} \right] \cos \theta + \left[\frac{4H}{\beta} \right] \sin \theta - 2H \left[\frac{\theta}{\beta} \right]^2 \cos \theta - \left[\frac{4H\theta}{\beta^2} \right] \sin \theta$$

$$Cy = a \cos \theta + b \cos \theta + H \cos \theta - 2H \cos \theta + \left[\frac{4H\theta}{\beta} \right] \cos \theta - 2H \left[\frac{\theta}{\beta} \right]^2 \cos \theta$$

$$Cy' = -a \sin \theta - b \sin \theta - H \sin \theta + 2H \sin \theta - \left[\frac{4H\theta}{\beta} \right] \sin \theta + \left[\frac{4H}{\beta} \right] \cos \theta + 2H \left[\frac{\theta}{\beta} \right]^2 \sin \theta - \left[\frac{4H\theta}{\beta^2} \right] \cos \theta$$

Fall

$$Cx = a \sin \theta + b \sin \theta + 2H \sin \theta - \left[\frac{4H\theta}{\beta} \right] \sin \theta + 2H \left[\frac{\theta}{\beta} \right]^2 \sin \theta$$

$$Cx' = a \cos \theta + b \cos \theta + 2H \cos \theta - \left[\frac{4H\theta}{\beta} \right] \cos \theta - \left[\frac{4H}{\beta} \right] \sin \theta + 2H \left[\frac{\theta}{\beta} \right]^2 \cos \theta + \left[\frac{4H\theta}{\beta^2} \right] \sin \theta$$

$$Cy = a \cos \theta + b \cos \theta + 2H \cos \theta - \left[\frac{4H\theta}{\beta} \right] \cos \theta + 2H \left[\frac{\theta}{\beta} \right]^2 \cos \theta$$

$$Cy' = -a \cos \theta - b \cos \theta - 2H \sin \theta + \left[\frac{4H\theta}{\beta} \right] \sin \theta - \left[\frac{4H}{\beta} \right] \cos \theta - 2H \left[\frac{\theta}{\beta} \right]^2 \sin \theta + \left[\frac{4H\theta}{\beta^2} \right] \cos \theta$$

iii. Simple harmonic motion

Rise

$$Cx = a \sin \theta + b \sin \theta + \frac{H}{2} \sin \theta - \left[\frac{H}{2} \cos \frac{\pi\theta}{\beta} \right] \sin \theta$$

$$Cx' = a \cos \theta + b \cos \theta + \frac{H}{2} \cos \theta - \left[\frac{H}{2} \cos \frac{\pi\theta}{\beta} \right] \cos \theta + \left[\frac{H\pi}{2\beta} \sin \frac{\pi\theta}{\beta} \right] \sin \theta$$

$$Cy = a \cos \theta + b \cos \theta + \frac{H}{2} \cos \theta - \left[\frac{H}{2} \cos \frac{\pi\theta}{\beta} \right] \cos \theta$$

$$Cy' = -a \sin \theta - b \sin \theta - \frac{H}{2} \sin \theta + \left[\frac{H}{2} \cos \frac{\pi\theta}{\beta} \right] \sin \theta + \left[\frac{H\pi}{2\beta} \sin \frac{\pi\theta}{\beta} \right] \cos \theta$$

Fall

$$Cx = a \sin \theta + b \sin \theta + \frac{H}{2} \sin \theta + \left[\frac{H}{2} \cos \frac{\pi\theta}{\beta} \right] \sin \theta$$

$$Cx' = a \cos \theta + b \cos \theta + \frac{H}{2} \cos \theta + \left[\frac{H}{2} \cos \frac{\pi\theta}{\beta} \right] \cos \theta - \left[\frac{H\pi}{2\beta} \sin \frac{\pi\theta}{\beta} \right] \sin \theta$$

$$Cy = a \cos \theta + b \cos \theta + \frac{H}{2} \cos \theta + \left[\frac{H}{2} \cos \frac{\pi\theta}{\beta} \right] \cos \theta$$

$$Cy' = -a \sin \theta - b \sin \theta - \frac{H}{2} \sin \theta - \left[\frac{H}{2} \cos \frac{\pi\theta}{\beta} \right] \sin \theta - \left[\frac{H\pi}{2\beta} \sin \frac{\pi\theta}{\beta} \right] \cos \theta$$

iv. Cycloidal motion

Rise

$$Cx = a \sin \theta + b \sin \theta + \frac{H\theta}{\beta} \sin \theta - \left[\frac{H}{2\pi} \sin \frac{2\pi\theta}{\beta} \right] \sin \theta$$

$$Cx' =$$

$$a \cos \theta + b \cos \theta + \frac{H\theta}{\beta} \cos \theta + \frac{H\theta}{\beta} \sin \theta - \left[\frac{H}{2\pi} \sin \frac{2\pi\theta}{\beta} \right] \cos \theta - \left[\frac{H}{2\pi} \cos \frac{2\pi\theta}{\beta} \right] \sin \theta$$

$$Cy = a \cos \theta + b \cos \theta + \frac{H\theta}{\beta} \cos \theta - \left[\frac{H}{2\pi} \sin \frac{2\pi\theta}{\beta} \right] \cos \theta$$

$$Cy' = -a \sin \theta - b \sin \theta - \frac{H\theta}{\beta} \sin \theta + \frac{H\theta}{\beta} \cos \theta + \left[\frac{H}{2\pi} \sin \frac{2\pi\theta}{\beta} \right] \sin \theta -$$

$$\left[\frac{H}{2\pi} \cos \frac{2\pi\theta}{\beta} \right] \cos \theta$$

Fall

$$Cx = a \sin \theta + b \sin \theta + H \sin \theta - \frac{H\theta}{\beta} \sin \theta + \left[\frac{H}{2\pi} \sin \frac{2\pi\theta}{\beta} \right] \sin \theta$$

$$Cx' = a \cos \theta + b \cos \theta + H \cos \theta - \frac{H\theta}{\beta} \cos \theta - \frac{H}{\beta} \sin \theta + \left[\frac{H}{2\pi} \sin \frac{2\pi\theta}{\beta} \right] \cos \theta +$$

$$\left[\frac{H}{\beta} \cos \frac{2\pi\theta}{\beta} \right] \sin \theta$$

$$Cy = a \cos \theta + b \cos \theta + H \cos \theta - \frac{H\theta}{\beta} \cos \theta + \left[\frac{H}{2\pi} \sin \frac{2\pi\theta}{\beta} \right] \cos \theta$$

$$Cy' = -a \sin \theta - b \sin \theta - H \sin \theta + \frac{H\theta}{\beta} \sin \theta - \frac{H}{\beta} \cos \theta - \left[\frac{H}{2\pi} \sin \frac{2\pi\theta}{\beta} \right] \sin \theta +$$

$$\left[\frac{H}{\beta} \cos \frac{2\pi\theta}{\beta} \right] \cos \theta$$

3.3 CASE STUDY

3.3.1 Example: Design a plate cam profile. [Rothbart, 2003].

A radial cam rotating clockwise drives a 12.7mm diameter roller follower as follows:

1. Rise of 15.875mm in 90degree of rotation
2. Fall of 15.875mm in 90degree of cam rotation
3. It dwells for the last 180 degree of the cam cycle

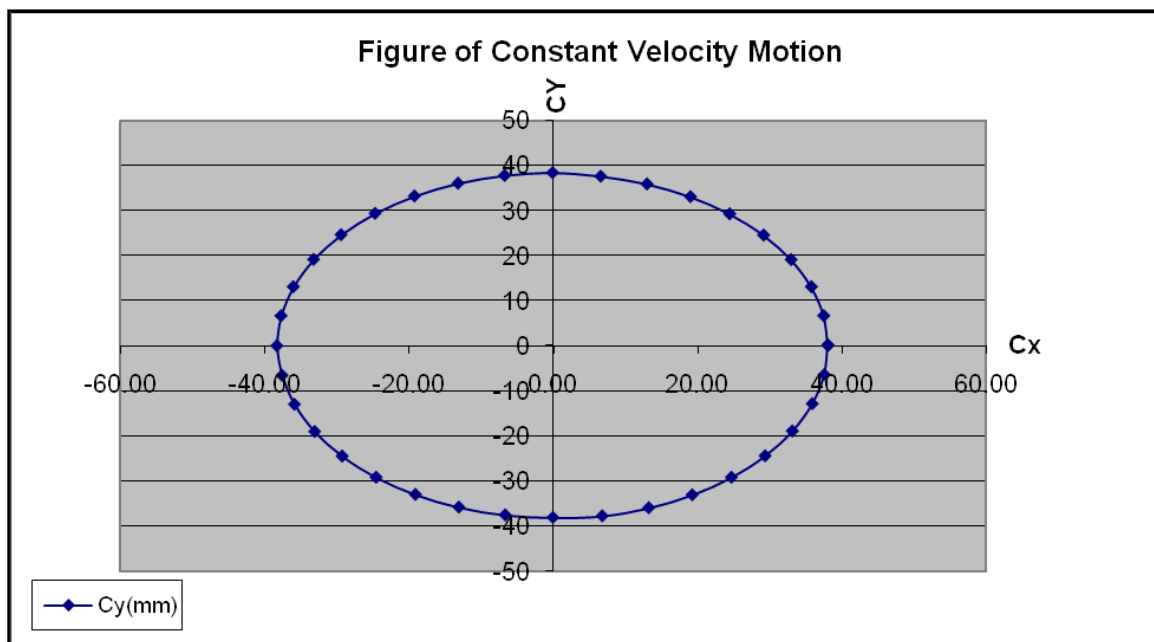
Assumption: Consider each rotation of motion is constant velocity motion

Table 3.1: Spreadsheet for graphical cam layout

a(mm)	b(mm)	$\beta(^{\circ})$	H(mm)	$\Phi(^{\circ})$	$\theta(^{\circ})$	Cx(mm)	Cy(mm)
12.7	25.4	90	15.875	0	0	38.10	0.176388889
12.7	25.4	90	15.875	10	10	37.58	-6.447632299
12.7	25.4	90	15.875	20	20	35.92	-12.88627476
12.7	25.4	90	15.875	30	30	33.16	-18.94342124
12.7	25.4	90	15.875	40	40	29.39	-24.43424079
12.7	25.4	90	15.875	50	50	24.72	-29.19082864
12.7	25.4	90	15.875	60	60	19.30	-33.06734047
12.7	25.4	90	15.875	70	70	13.27	-35.94446376
12.7	25.4	90	15.875	80	80	6.83	-37.7330895
12.7	25.4	180	0	90	90	2.33E-15	-38.1
12.7	25.4	180	0	100	100	-6.616	-37.52117539
12.7	25.4	180	0	110	110	-13.031	-35.80228885
12.7	25.4	180	0	120	120	-19.05	-32.99556788
12.7	25.4	180	0	130	130	-24.4902	-29.18629328
12.7	25.4	180	0	140	140	-29.1863	-24.49020793
12.7	25.4	180	0	150	150	-32.9956	-19.05
12.7	25.4	180	0	160	160	-35.8023	-13.03096746
12.7	25.4	180	0	170	170	-37.5212	-6.615995569
12.7	25.4	360	15.875	180	180	-38.2385	-0.044097222
12.7	25.4	360	15.875	190	190	-37.6728	6.59796119
12.7	25.4	360	15.875	200	200	-35.962	13.04217622
12.7	25.4	360	15.875	210	210	-33.1576	19.09262307
12.7	25.4	360	15.875	220	220	-29.3443	24.56526505
12.7	25.4	360	15.875	230	230	-24.6378	29.29355148
12.7	25.4	360	15.875	240	240	-19.1805	33.1334863
12.7	25.4	360	15.875	250	250	-13.1382	35.96801337
12.7	25.4	360	15.875	260	260	-6.69417	37.71058476

Table 3.1: Spreadsheet for graphical cam layout (continued)

a(mm)	b(mm)	$\beta(^{\circ})$	H(mm)	$\Phi(^{\circ})$	$\theta(^{\circ})$	Cx(mm)	Cy(mm)
12.7	25.4	360	15.875	270	270	-0.0441	38.30780326
12.7	25.4	360	0	280	280	6.615996	37.52117539
12.7	25.4	360	0	290	290	13.03097	35.80228885
12.7	25.4	360	0	300	300	19.05	32.99556788
12.7	25.4	360	0	310	310	24.49021	29.18629328
12.7	25.4	360	0	320	320	29.18629	24.49020793
12.7	25.4	360	0	330	330	32.99557	19.05
12.7	25.4	360	0	340	340	35.80229	13.03096746
12.7	25.4	360	0	350	350	37.52118	6.615995569
12.7	25.4	360	0	360	360	38.1	9.33563E-15

**Figure 3.1:** Cam graphical layout using constant velocity motion

3.4 FLOW CHART

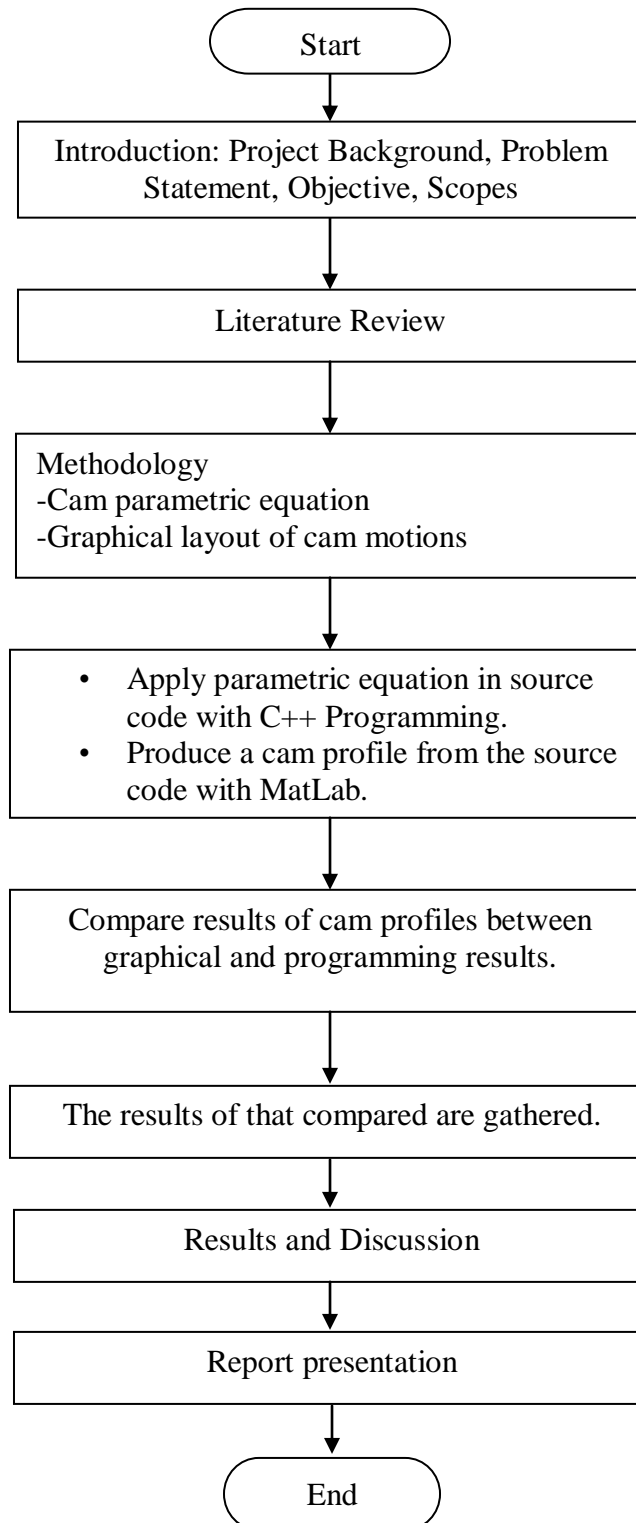


Figure 3.2: Flow chart for methodology

3.5 ADVANTAGES OF USING CAM MOTION EQUATION

There are particularly some advantages of using cam parametric equation to obtain a cam shape. One of the advantages is, by using cam motions equation as programming input, the cam shape are theoretically will be smooth according to the type of motion used. The shape can be adjustable and multiple type of motion can be applied in the cam shape.

If the shape of cam uses a model or coordinate based, it will result in a cam shape with sharp edges and that may cause sudden acceleration during the motion of cam-follower mechanism. The user who use this type of method does not have to be highly skilled in modeling because the user only have to be capable of know what type of motion or simply what type of cam shape does he/she need to produce. This methods are also time consumable because only few details such as cam parametric equation and other data needed to produce a desired cam shape.

The smoothness of the cam shape produced usually depends on the type of parametric equation used. The smoother the shape means the better performance the system can obtain hence the less vibration created.

3.5.1 Comparison between Computer Aided Design (CAD) and CAM Application

CAD has many advantages too but among those advantages, there are also disadvantages. In order to use CAD, a user must require time and skilled software knowledge to build a cam shape has the smooth surface to have better performance. This goes the same for CAM, where the user has to identify the geometry shape of cam by drawing the coordinate of cam.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 INTRODUCTION

In this chapter, we'll look further for the results of using graphical and programming method to analyze the cam mechanism system with parametric equation. In order to understand more than theory from graphical method, programming method is used to obtain profile of cam shape. The graphical method is used to determine the outcome of cam profile for the parametric equation used meanwhile the programming method are used to obtain a similar cam profile from graphical method but by applying the equation into a computational data in mathematical software such as C Programming where in this project we will use Microsoft Studio C++ Programming.

4.2 GRAPHICAL METHOD

By referring to the cam profile analysis using equation of motion it is necessary to first understand the theory concept of the cam shape respect to their equation of motion. Therefore, before experimentally we test the equation into programming, graphical of the cam need o be constructed. The graphical method can be used for four type of motion discussed in methodology chapter. The four type of motion will produce a significant parametric equation that will be used in the programming method.

4.2.1 Graphical Results

In this example of graphical results, we can see that the shape of cam can be obtained by using the exactly cam parametric equation. From the equation we can obtain four graph of cam profile for each equation and one combine of graph from combining all the motion into one graph. First, we decide to have the cam to rise for first quadrant and then fall for next quadrant. After that, the cam will have rise and fall again for other quadrants respectively. Finally, when we can see the differences of the graphical results and compare to see which have the smoothest surface.

Table 4.1: Constant Velocity Motion (Appendix A1)

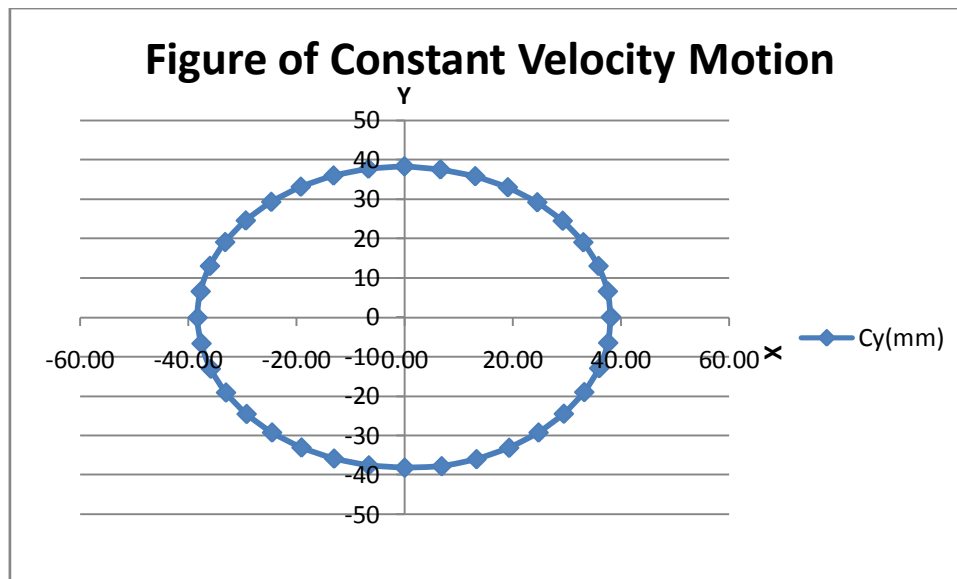
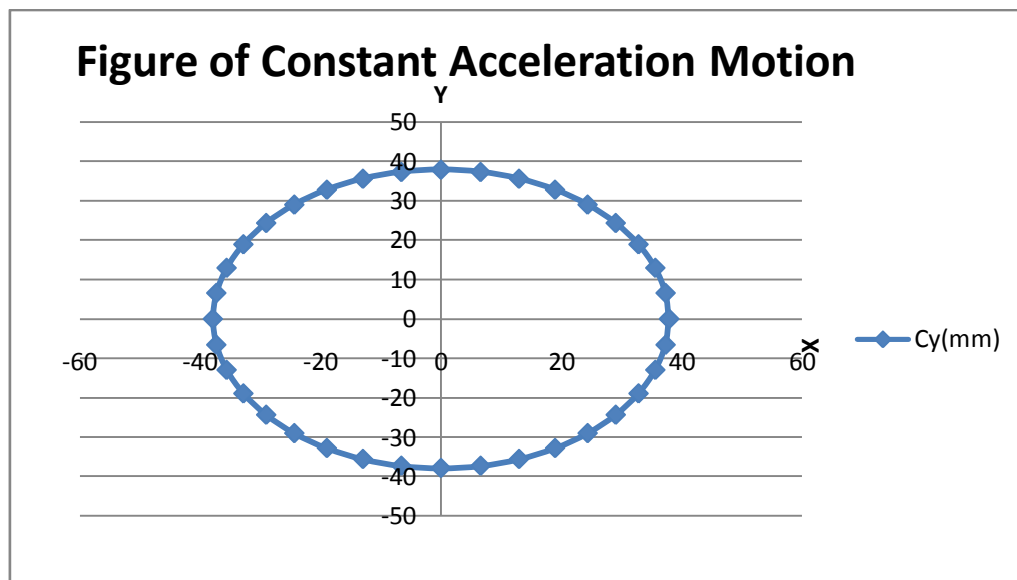
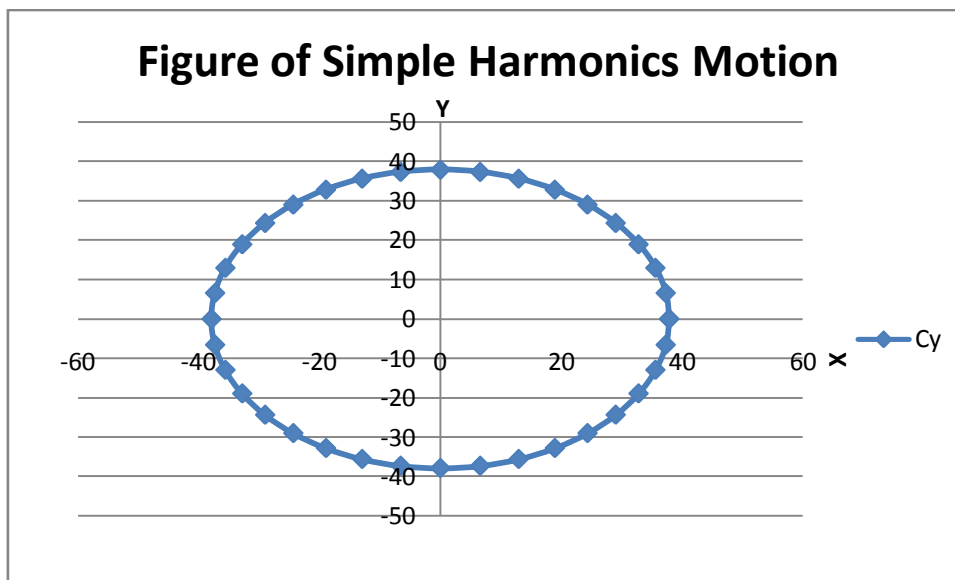


Figure 4.1: Graph of Constant Velocity Motion

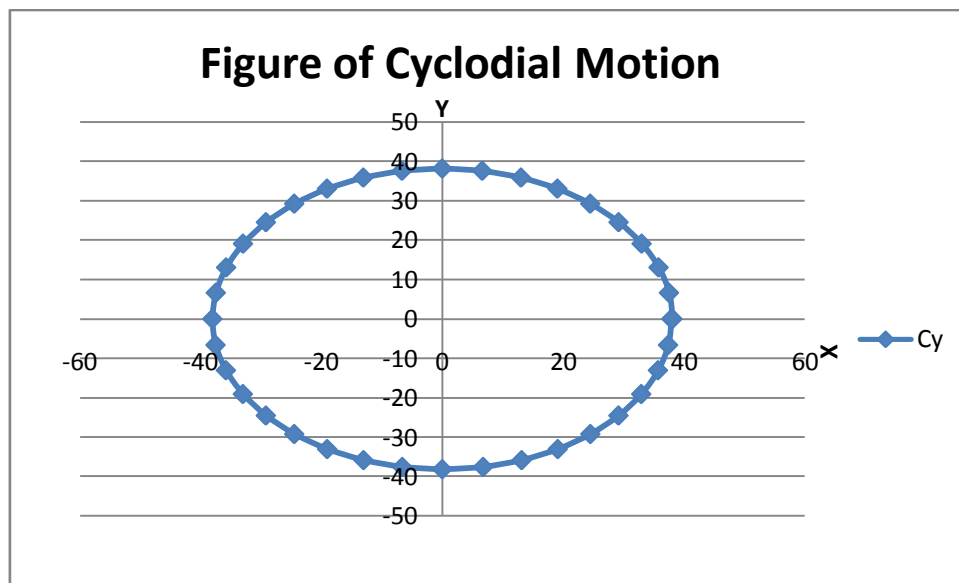
From the result obtained as above, the shape of graphical result that we obtained above is Constant Velocity Motion. We can also see that the highest value from the graph for X-axis were 38.1mm when the cam was rise and -38.2mm when it falls at 180 degrees. Whereas the highest value from the graph for Y-axis were 38.3mm when the cam was rise at 90 degrees and -38.1mm when it falls at 270 degrees. The graph above has sharper edges and the curve lines of the cam are not the smoothest because the parametric equation used doesn't concern with angular parameters.

Table 4.2: Constant Acceleration Motion (Appendix A2)**Figure 4.2:** Graph of Constant Acceleration Motion

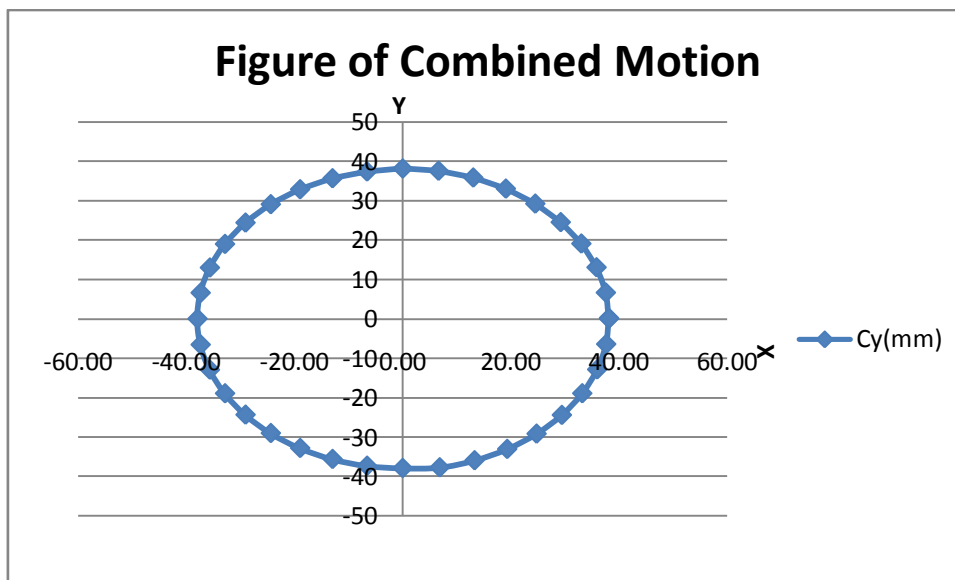
From the result obtained as above, the shape of graphical result that we obtained above is Constant Acceleration Motion. We can also see that the highest value from the graph for X-axis were 37.9mm when the cam was rise and -37.9mm when it falls at 180 degrees. Whereas the highest value from the graph for Y-axis were 37.9mm when the cam was rise at 90 degrees and -37.9mm when it falls at 270 degrees. The graph above has lower sharp edges and the curves lines of the cam are smoother. These were due to the trigonometric functions in the parametric equations.

Table 4.3: Simple Harmonic Motion (Appendix A3)**Figure 4.3:** Graph of Simple Harmonic Motion

From the result obtained as above, the shape of graphical result that we obtained above is Simple Harmonic Motion. We can also see that the highest value from the graph for X-axis were 37.9mm when the cam was rise and -37.9mm when it falls at 180 degrees. Whereas the highest value from the graph for Y-axis were 37.9mm when the cam was rise at 90 degrees and -37.9mm when it falls at 270 degrees. This graph begins to have a smoother edges since the equation that we used contain angular parameters such as pi.

Table 4.4: Cycloidal Motion (Appendix A4)**Figure 4.4:** Graph of Cycloidal Motion

From the result obtained as above, the shape of graphical result that we obtained above is Cycloidal Motion. We can also see that the highest value from the graph for X-axis were 387.92mm when the cam was rise and -38.04mm when it falls at 180 degrees. Whereas the highest value from the graph for Y-axis were 38.1mm when the cam was rise at 90 degrees and -38.1mm when it falls at 270 degrees. The graph above has sharper edges and the curve lines of the cam are the smoothest because the parametric equation used concern with angular parameters such as pi and trigonometric functions.

Table 4.5: Combined Motions (Appendix A5)**Figure 4.5:** Graph of Combined Motion

From the result obtained as above, the shape of graphical result that we obtained above is Combined Motion. We can also see that the highest value from the graph for X-axis were 38.10mm when the cam was rise and -37.90mm when it falls at 180 degrees. Whereas the highest value from the graph for Y-axis were 38.1mm when the cam was rise at 90 degrees and -37.9mm when it falls at 270 degrees. The graph above has smooth and also sharper edges and the curve lines of the cam are combined so it appears that it has many features of surface on the cam profile which gives different type of motion as it turns. The combined motion is the easiest way to identify the differences between all other four types of motions.

4.3 PROGRAMMING METHOD

Programming method is where we use the cam parametric equation to translate it into C++ software. The essential of doing so is to translate numerical data into computational data that is use to run the cam manufacturing machine. By inserting the cam parametric equation into the software, the shape of cam that will be produced will not be much different than the graphical results if all specification is required.

4.3.1 C++ PROGRAMMING

There are four type of C++ coding that developed, which are the Constant Velocity Motion, Constant Acceleration Motion, Simple Harmonic Motion, and also Cycloidal Motion. These four motions which have been translated into computational data can easily run with C Programming software in order to obtain a cam profile.

4.3.2 Programming Results

In these results we obtained from programming method shows a possible shape of graphs according to their specified motion but there are few differences from the graphical results as we obtained before. The results that we obtained were basically came from the same parametric equation that we used in graphical results but it is translated in computational data through C++ Programming where the parametric equation was embedded into a source code used in the programming. The result of the C++ Programming was only a portion of obtaining the cam profile. Further method is used by using MatLab software to get the data of cam profile from C++ Programming results. Once the data is obtained from MatLab, a cam profile can be obtained and the shape can be analyzed to identify whether the correctness of computational data in the C++ Programming.

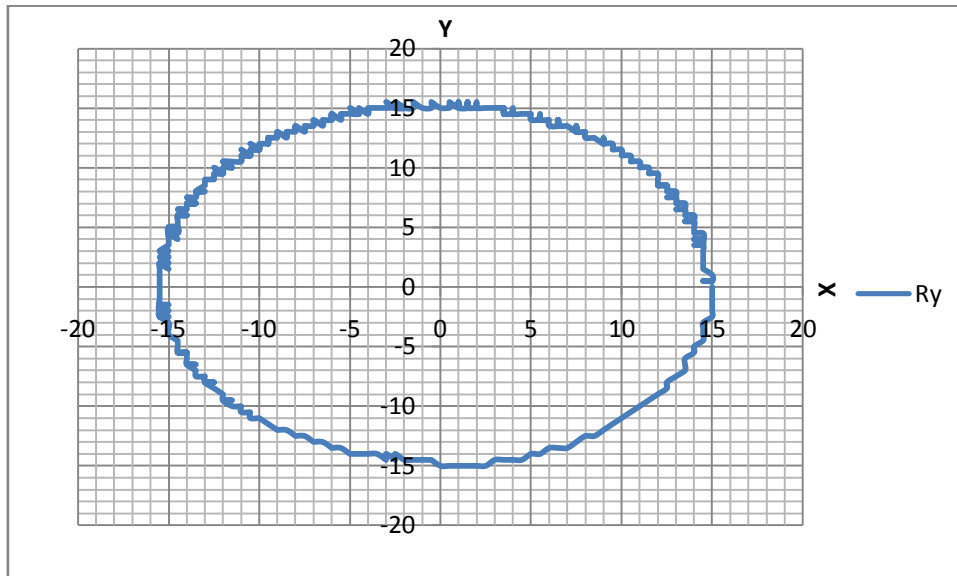


Figure 4.6: Program result of Constant Velocity Motion

The result above is obtained from combining the results from C++ Programming and MatLab software. From the result, the type of graph that we obtained is Constant Velocity Motion. We can also see that the highest value from the graph for X-axis were 14.5mm when the cam was rise and -15.5mm when it falls at 180 degrees. Whereas the highest value from the graph for Y-axis were 15.5mm when the cam was rise at 90 degrees and -14.5mm when it falls at 270 degrees. The graph above has sharper edges and the curve lines of the cam are not the smoothest because the parametric equation used doesn't concern with angular parameters.

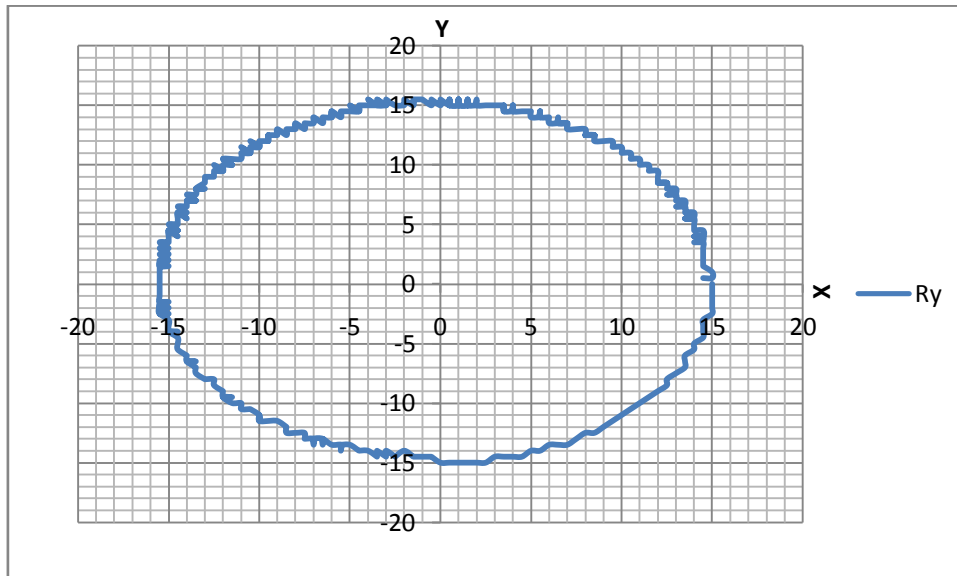


Figure 4.7: Program result of Constant Acceleration Motion

The result above is obtained from combining the results from C++ Programming and MatLab software. From the result, the type of graph that we obtained is Constant Acceleration Motion. We can also see that the highest value from the graph for X-axis were 14.5mm when the cam was rise and -15.5mm when it falls at 180 degrees. Whereas the highest value from the graph for Y-axis were 15mm when the cam was rise at 90 degrees and -14.5mm when it falls at 270 degrees. The graph above has lower sharp edges and the curves lines of the cam are smoother. These were due to the trigonometric functions in the parametric equations.

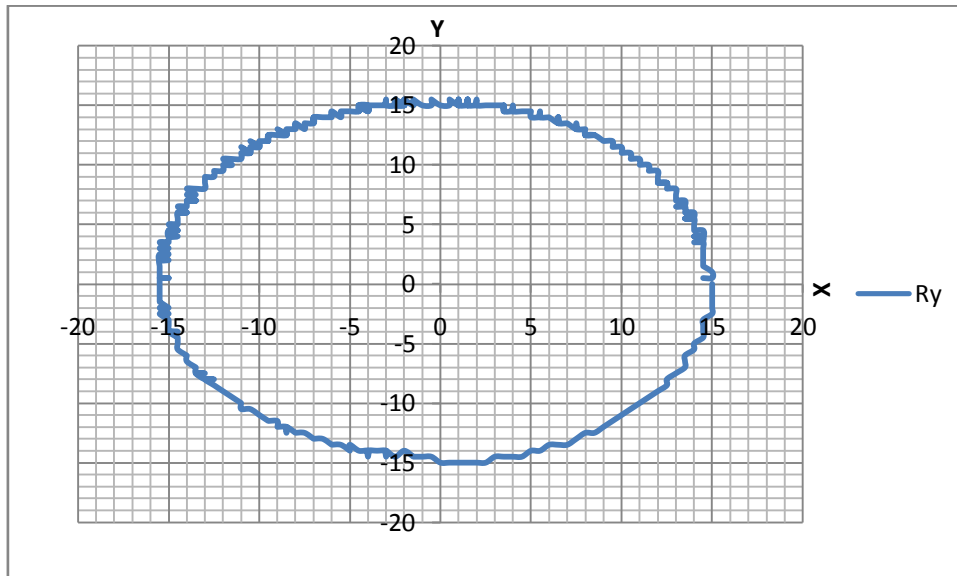


Figure 4.8: Program result of Simple Harmonics Motion

The result above is obtained from combining the results from C++ Programming and MatLab software. From the result, the type of graph that we obtained is Simple Harmonics Motion. We can also see that the highest value from the graph for X-axis were 15mm when the cam was rise and -15.5mm when it falls at 180 degrees. Whereas the highest value from the graph for Y-axis were 15.5mm when the cam was rise at 90 degrees and -15mm when it falls at 270 degrees. This graph begins to have a smoother edges since the equation that we used contain angular parameters such as pi.

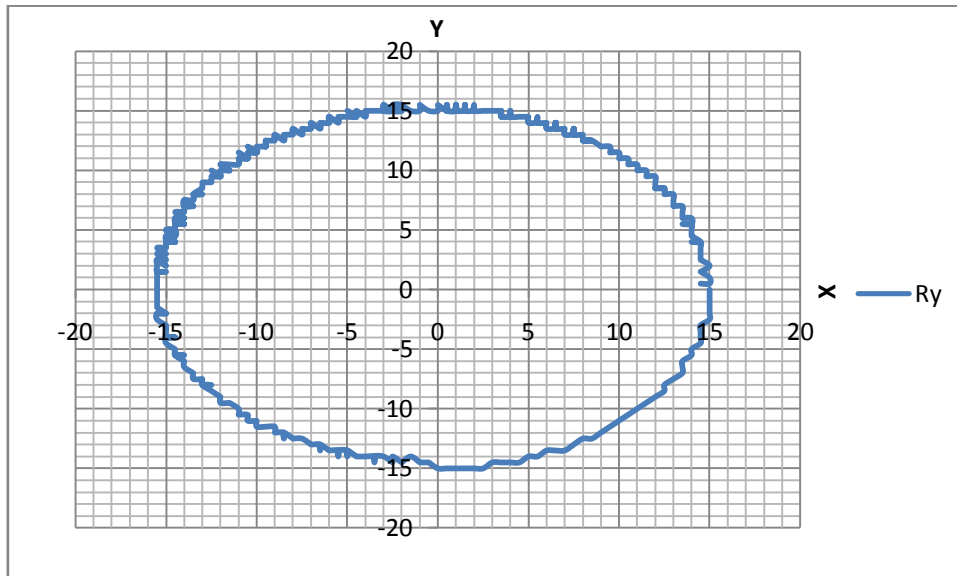


Figure 4.9: Program result of Cycloidal Motion

The result above is obtained from combining the results from C++ Programming and MatLab software. From the result, the type of graph that we obtained is Cycloidal Motion. We can also see that the highest value from the graph for X-axis were 15mm when the cam was rise and -15.5mm when it falls at 180 degrees. Whereas the highest value from the graph for Y-axis were 15.5mm when the cam was rise at 90 degrees and -15mm when it falls at 270 degrees. The graph above has sharper edges and the curve lines of the cam are the smoothest because the parametric equation used concern with angular parameters such as pi and trigonometric functions.

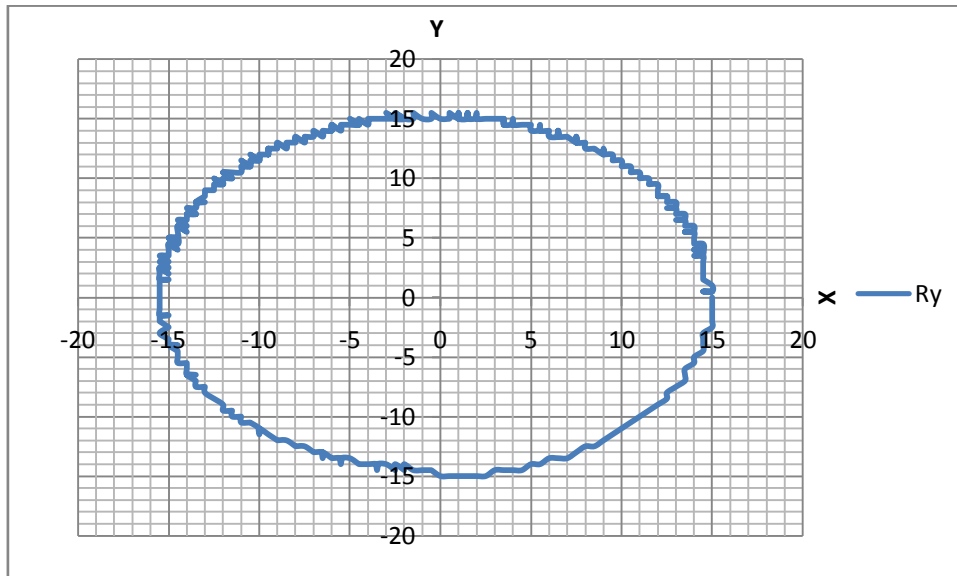


Figure 4.10: Program result of Combined Motions

The result above is obtained from combining the results from C++ Programming and MatLab software. From the result, the type of graph that we obtained is Constant Velocity Motion. We can also see that the highest value from the graph for X-axis were 15mm when the cam was rise and -15.5mm when it falls at 180 degrees. Whereas the highest value from the graph for Y-axis were 15.5mm when the cam was rise at 90 degrees and -15mm when it falls at 270 degrees. The graph above has smooth and also sharper edges and the curve lines of the cam are combined so it appears that it has many features of surface on the cam profile which gives different type of motion as it turns. The combined motion is the easiest way to identify the differences between all other four types of motions.

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 INTRODUCTION

In this final chapter, the purpose is to summarize about the overall process related to the development of controller software for cam manufacturing system of this thesis. The conclusion includes several layers such as the expected outcomes and also the recommendations that relate to previous chapters, especially from Chapter 4. For a further study, we should made conclusion to emphasize the results we obtained to have recommendations for better outcome.

5.2 CONCLUSIONS

In summary for this thesis, we can conclude that;

It is best to first understand the concept theoretically of cam profile before running into experimental process. The necessary of understanding theoretically is to avoid mistakes and casualties towards experiments during hand works. The process of understanding cam profile includes reading from previous research, journals, books and also references that related closely to the topic itself. Other than that, graphical method is the source of understanding cam profile, motions and equations related to it.

After few research and development of theoretically understanding, the process of obtaining results of experimental by using multiple sources such as C++ program, MatLab and also journals and reference to produce real and consistent results.

The process of experimental should be separated in few ways, which are;

- i. Produce a parametric equation from cam's equation of motions (i.e.; Constant Velocity Motion, Constant Acceleration Motion, Simple Harmonic Motion, and Cycloidal Motion)
- ii. Translate parametric equations of mathematical data into computational data via C++ programming.
- iii. Associate the cam parametric equation into the C++ coding obtained into to have desired cam profile with type of motions chosen.
- iv. Comparing the results from graphical and programming method to understand and discuss the differences in order to build a better cam profile.

Finally, we have to compare graphical result with experimental results and the error of differences will be discussed.

Contributions from this research;

- i. This cam controller software can widely used in engineering field to generated smooth-edge cam profile
- ii. Smooth edge cam profile can produce effective results to the follower and thus obtain better results.
- iii. This method can be collaborated with typical software and mass produce to benefit in production of cam profile for engineering field.

5.3 RECOMMENDATIONS

In future research, the potentiality of this method can be specified into few areas such as;

- i. Using standard and more user friendly coding programs that specially made for developing cam profile.
- ii. Enhance the quality and surface roughness of cam profile using more engineering design cutting tools.

REFERENCE:

1. Norton,R.L. and Mosier,R.G. 1988. CAM Design and Manufacturing Handbook. New York: Industrial Press Inc.
2. Rothbart,H.A. 2003. CAM Design Handbook. New York: McGraw Hill Handbooks.
3. Jensen,P.W. 1987. CAM Design And Manufacture. Faulkner,L.L. Menkes,S.B. New York: Marcel Dekker Inc.
4. Erdman,A.G. Sandor, G.N. 1984. Mechanism Design: Analysis and Synthesis. New Jersey: Prenice Hall Inc.
5. Myszka,D.H. 2005,2002,1999, Machines and Mechanisms: Applied Kinematic Analysis,New Jersey,Pearson Prentice Hall
- 6 H. S. Yan And M. C. Tsai, M. H. Hsu, 1995, An Experimental Study Of The Effects Of Cam Speeds On Cam-Follower Systems, Pergamon, Great Britain
7. Mihai Viteazu, R. Militaru, E.-Ch. Lovasz,. Functional Optimization Of The Cam Mechanisms Through Using Helical Springs With Variable Geometry, "Politehnica" University of Timișoara, 1, 1900 Timișoara, Romania
8. Caldwell,J., Ram,Y.M., (1990). Mathematical Modelling: Concepts and Case Studies, Springer
9. Parmley, R.O.,2000. Illustrated sourcebook of mechanical components. New York: McGraw-Hill Professional
10. Yi Zhang, Susan Finger, Stephannie Behrens, 2006, Rapid Design through Virtual and Physical Prototyping, Carnegie Mellon University

APPENDIX A

APPENDIX A1

Cam Profile Graphical Result for Constant Velocity Motion

a(mm)	b(mm)	$\beta(^{\circ})$	H(mm)	$\Phi(^{\circ})$	$\theta(^{\circ})$	Cx(mm)	Cy(mm)
12.7	25.4	90	15.875	0	0	38.10	0.176388889
12.7	25.4	90	15.875	10	10	37.58	-6.447632299
12.7	25.4	90	15.875	20	20	35.92	-12.88627476
12.7	25.4	90	15.875	30	30	33.16	-18.94342124
12.7	25.4	90	15.875	40	40	29.39	-24.43424079
12.7	25.4	90	15.875	50	50	24.72	-29.19082864
12.7	25.4	90	15.875	60	60	19.30	-33.06734047
12.7	25.4	90	15.875	70	70	13.27	-35.94446376
12.7	25.4	90	15.875	80	80	6.83	-37.7330895
12.7	25.4	180	0	90	90	2.33E-15	-38.1
12.7	25.4	180	0	100	100	-6.616	-37.52117539
12.7	25.4	180	0	110	110	-13.031	-35.80228885
12.7	25.4	180	0	120	120	-19.05	-32.99556788
12.7	25.4	180	0	130	130	-24.4902	-29.18629328
12.7	25.4	180	0	140	140	-29.1863	-24.49020793
12.7	25.4	180	0	150	150	-32.9956	-19.05
12.7	25.4	180	0	160	160	-35.8023	-13.03096746
12.7	25.4	180	0	170	170	-37.5212	-6.615995569
12.7	25.4	360	15.875	180	180	-38.2385	-0.044097222
12.7	25.4	360	15.875	190	190	-37.6728	6.59796119
12.7	25.4	360	15.875	200	200	-35.962	13.04217622
12.7	25.4	360	15.875	210	210	-33.1576	19.09262307
12.7	25.4	360	15.875	220	220	-29.3443	24.56526505
12.7	25.4	360	15.875	230	230	-24.6378	29.29355148
12.7	25.4	360	15.875	240	240	-19.1805	33.1334863
12.7	25.4	360	15.875	250	250	-13.1382	35.96801337
12.7	25.4	360	15.875	260	260	-6.69417	37.71058476
12.7	25.4	360	15.875	270	270	-0.0441	38.30780326
12.7	25.4	360	0	280	280	6.615996	37.52117539
12.7	25.4	360	0	290	290	13.03097	35.80228885
12.7	25.4	360	0	300	300	19.05	32.99556788
12.7	25.4	360	0	310	310	24.49021	29.18629328
12.7	25.4	360	0	320	320	29.18629	24.49020793
12.7	25.4	360	0	330	330	32.99557	19.05
12.7	25.4	360	0	340	340	35.80229	13.03096746
12.7	25.4	360	0	350	350	37.52118	6.615995569
12.7	25.4	360	0	360	360	38.1	9.33563E-15

Appendix A2

Cam Profile Graphical Result for Constant Acceleration Motion

a(mm)	b(mm)	$\beta(^{\circ})$	H(mm)	$\Phi(^{\circ})$	$\theta(^{\circ})$	Cx(mm)	Cy(mm)
12.5	25.4	90	15.875	0	0	37.9	0
12.5	25.4	90	15.875	10	10	37.32457	-6.57994
12.5	25.4	90	15.875	20	20	35.61574	-12.9602
12.5	25.4	90	15.875	30	30	32.82535	-18.947
12.5	25.4	90	15.875	40	40	29.03807	-24.3587
12.5	25.4	90	15.875	50	50	24.36881	-29.031
12.5	25.4	90	15.875	60	60	18.95926	-32.822
12.5	25.4	90	15.875	70	70	12.97356	-35.6166
12.5	25.4	90	15.875	80	80	6.593373	-37.3298
12.5	25.4	180	0	90	90	2.32E-15	-37.9
12.5	25.4	180	0	100	100	-6.58127	-37.3242
12.5	25.4	180	0	110	110	-12.9626	-35.6144
12.5	25.4	180	0	120	120	-18.95	-32.8224
12.5	25.4	180	0	130	130	-24.3617	-29.0331
12.5	25.4	180	0	140	140	-29.0331	-24.3617
12.5	25.4	180	0	150	150	-32.8224	-18.95
12.5	25.4	180	0	160	160	-35.6144	-12.9626
12.5	25.4	180	0	170	170	-37.3242	-6.58127
12.5	25.4	270	15.875	180	180	-37.9	-4.60E-15
12.5	25.4	270	15.875	190	190	-37.3242	6.581266
12.5	25.4	270	15.875	200	200	-35.6144	12.96256
12.5	25.4	270	15.875	210	210	-32.8224	18.95
12.5	25.4	270	15.875	220	220	-29.0331	24.36165
12.5	25.4	270	15.875	230	230	-24.3617	29.03308
12.5	25.4	270	15.875	240	240	-18.95	32.82236
12.5	25.4	270	15.875	250	250	-12.9626	35.61435
12.5	25.4	270	15.875	260	260	-6.58127	37.32421
12.5	25.4	360	0	270	270	-7.00E-15	37.9
12.5	25.4	360	0	280	280	6.581266	37.32421
12.5	25.4	360	0	290	290	12.96256	35.61435
12.5	25.4	360	0	300	300	18.95	32.82236
12.5	25.4	360	0	310	310	24.36165	29.03308
12.5	25.4	360	0	320	320	29.03308	24.36165
12.5	25.4	360	0	330	330	32.82236	18.95
12.5	25.4	360	0	340	340	35.61435	12.96256
12.5	25.4	360	0	350	350	37.32421	6.581266
12.5	25.4	360	0	360	360	37.9	9.29E-15

Appendix A3

Cam Profile Graphical Result for Simple Harmonic Motion

a	b	β	h	Φ	θ	Cx	Cy
12.5	25.4	90	15.875	0	0	37.9	0
12.5	25.4	90	15.875	10	10	37.32465	-6.57963
12.5	25.4	90	15.875	20	20	35.61606	-12.9596
12.5	25.4	90	15.875	30	30	32.82604	-18.9463
12.5	25.4	90	15.875	40	40	29.03922	-24.358
12.5	25.4	90	15.875	50	50	24.37047	-29.0305
12.5	25.4	90	15.875	60	60	18.96141	-32.8219
12.5	25.4	90	15.875	70	70	12.97612	-35.6171
12.5	25.4	90	15.875	80	80	6.596181	-37.3311
12.5	25.4	180	0	90	90	2.32E-15	-37.9
12.5	25.4	180	0	100	100	-6.58127	-37.3242
12.5	25.4	180	0	110	110	-12.9626	-35.6144
12.5	25.4	180	0	120	120	-18.95	-32.8224
12.5	25.4	180	0	130	130	-24.3617	-29.0331
12.5	25.4	180	0	140	140	-29.0331	-24.3617
12.5	25.4	180	0	150	150	-32.8224	-18.95
12.5	25.4	180	0	160	160	-35.6144	-12.9626
12.5	25.4	180	0	170	170	-37.3242	-6.58127
12.5	25.4	270	15.875	180	180	-37.9053	-0.00337
12.5	25.4	270	15.875	190	190	-37.3306	6.578786
12.5	25.4	270	15.875	200	200	-35.6218	12.96128
12.5	25.4	270	15.875	210	210	-32.8306	18.9502
12.5	25.4	270	15.875	220	220	-29.0418	24.36358
12.5	25.4	270	15.875	230	230	-24.3705	29.03694
12.5	25.4	270	15.875	240	240	-18.9586	32.82827
12.5	25.4	270	15.875	250	250	-12.9705	35.62235
12.5	25.4	270	15.875	260	260	-6.58798	37.33425
12.5	25.4	360	0	270	270	-7.00E-15	37.9
12.5	25.4	360	0	280	280	6.581266	37.32421
12.5	25.4	360	0	290	290	12.96256	35.61435
12.5	25.4	360	0	300	300	18.95	32.82236
12.5	25.4	360	0	310	310	24.36165	29.03308
12.5	25.4	360	0	320	320	29.03308	24.36165
12.5	25.4	360	0	330	330	32.82236	18.95
12.5	25.4	360	0	340	340	35.61435	12.96256
12.5	25.4	360	0	350	350	37.32421	6.581266
12.5	25.4	360	0	360	360	37.9	9.29E-15

Appendix A4

Cam Profile Graphical Result for Cycloidal Motion

a	b	β	h	Φ	θ	Cx	Cy
12.7	25.4	90	15.875	0	0	37.92361	0
12.7	25.4	90	15.875	10	10	37.37811	-6.61598
12.7	25.4	90	15.875	20	20	35.69692	-13.0309
12.7	25.4	90	15.875	30	30	32.93112	-19.0499
12.7	25.4	90	15.875	40	40	29.16475	-24.4901
12.7	25.4	90	15.875	50	50	24.51222	-29.1862
12.7	25.4	90	15.875	60	60	19.11488	-32.9955
12.7	25.4	90	15.875	70	70	13.1367	-35.8023
12.7	25.4	90	15.875	80	80	6.759288	-37.5214
12.7	25.4	180	0	90	90	2.33E-15	-38.1
12.7	25.4	180	0	100	100	-6.616	-37.5212
12.7	25.4	180	0	110	110	-13.031	-35.8023
12.7	25.4	180	0	120	120	-19.05	-32.9956
12.7	25.4	180	0	130	130	-24.4902	-29.1863
12.7	25.4	180	0	140	140	-29.1863	-24.4902
12.7	25.4	180	0	150	150	-32.9956	-19.05
12.7	25.4	180	0	160	160	-35.8023	-13.031
12.7	25.4	180	0	170	170	-37.5212	-6.616
12.7	25.4	270	15.875	180	180	-38.0415	-0.00016
12.7	25.4	270	15.875	190	190	-37.4738	6.615857
12.7	25.4	270	15.875	200	200	-35.7675	13.03086
12.7	25.4	270	15.875	210	210	-32.9745	19.04995
12.7	25.4	270	15.875	220	220	-29.1795	24.49022
12.7	25.4	270	15.875	230	230	-24.4978	29.18639
12.7	25.4	270	15.875	240	240	-19.0719	32.99577
12.7	25.4	270	15.875	250	250	-13.0664	35.8026
12.7	25.4	270	15.875	260	260	-6.66383	37.52161
12.7	25.4	360	0	270	270	-7.00E-15	38.1
12.7	25.4	360	0	280	280	6.615996	37.52118
12.7	25.4	360	0	290	290	13.03097	35.80229
12.7	25.4	360	0	300	300	19.05	32.99557
12.7	25.4	360	0	310	310	24.49021	29.18629
12.7	25.4	360	0	320	320	29.18629	24.49021
12.7	25.4	360	0	330	330	32.99557	19.05
12.7	25.4	360	0	340	340	35.80229	13.03097
12.7	25.4	360	0	350	350	37.52118	6.615996
12.7	25.4	360	0	360	360	38.1	9.34E-15

Appendix A5

Cam Profile Graphical Result for Combined Motion

a(mm)	b(mm)	$\beta(^{\circ})$	H(mm)	$\Phi(^{\circ})$	$\theta(^{\circ})$	Cx(mm)	Cy(mm)
12.7	25.4	90	15.875	0	0	38.10	0.176389
12.7	25.4	90	15.875	10	10	37.58	-6.44763
12.7	25.4	90	15.875	20	20	35.92	-12.8863
12.7	25.4	90	15.875	30	30	33.16	-18.9434
12.7	25.4	90	15.875	40	40	29.39	-24.4342
12.7	25.4	90	15.875	50	50	24.72	-29.1908
12.7	25.4	90	15.875	60	60	19.30	-33.0673
12.7	25.4	90	15.875	70	70	13.27	-35.9445
12.7	25.4	90	15.875	80	80	6.83	-37.7331
12.7	25.4	180	0	90	90	2.32E-15	-37.9
12.7	25.4	180	0	100	100	-6.58127	-37.3242
12.7	25.4	180	0	110	110	-12.9626	-35.6144
12.7	25.4	180	0	120	120	-18.95	-32.8224
12.7	25.4	180	0	130	130	-24.3617	-29.0331
12.7	25.4	180	0	140	140	-29.0331	-24.3617
12.7	25.4	180	0	150	150	-32.8224	-18.95
12.7	25.4	180	0	160	160	-35.6144	-12.9626
12.7	25.4	180	0	170	170	-37.3242	-6.58127
12.7	25.4	360	15.875	180	180	-37.9053	-0.00337
12.7	25.4	360	15.875	190	190	-37.3306	6.578786
12.7	25.4	360	15.875	200	200	-35.6218	12.96128
12.7	25.4	360	15.875	210	210	-32.8306	18.9502
12.7	25.4	360	15.875	220	220	-29.0418	24.36358
12.7	25.4	360	15.875	230	230	-24.3705	29.03694
12.7	25.4	360	15.875	240	240	-18.9586	32.82827
12.7	25.4	360	15.875	250	250	-12.9705	35.62235
12.7	25.4	360	15.875	260	260	-6.58798	37.33425
12.7	25.4	360	15.875	270	270	-7.00E-15	38.1
12.7	25.4	360	0	280	280	6.615996	37.52118
12.7	25.4	360	0	290	290	13.03097	35.80229
12.7	25.4	360	0	300	300	19.05	32.99557
12.7	25.4	360	0	310	310	24.49021	29.18629
12.7	25.4	360	0	320	320	29.18629	24.49021
12.7	25.4	360	0	330	330	32.99557	19.05
12.7	25.4	360	0	340	340	35.80229	13.03097
12.7	25.4	360	0	350	350	37.52118	6.615996
12.7	25.4	360	0	360	360	38.1	9.34E-15

APPENDIX B

Appendix B1

Program for Constant Velocity Motion

```

#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
const double TWOPIE = 6.2831853072;
const double STEP = 0.5;
const double Rmin = 15.0;
const double Rtool = 5.0;
_inline double SQR(double a) {return pow(a,2);}
void main()
{
FILE *fpOut;
fpOut = fopen("CVM.DAT","w");
double gx,gx1,gy,gy1;
double a=12.7,b=25.4;
double H=15.875,H1=0;
double B1=1.571,B2=3.142,B3=4.712,B4=6.283;
double ang1=1.86,ang2=2.645,ang3=2.911,ang4=4.712;
double xtool=Rmin+Rtool,ytool,ztool;
double X = Rmin,Y=0.0,Z = 0.0;
double x = Rmin,y = 0.0,z = 0.0;
double xp,yp,zp;
double xDist,yDist,zDist;
double u = 0.0;
int XF,YF,ZF;
int GF1,GF2;
int SelectedPlane;
double DX,DY,DZ;
unsigned long TP = 0;
printf("Starting Thesis Program ... \n\n");
while(u<=TWOPIE)
{
if((u>=0)&&(u<=ang1))
{gx=a*sin(u)+b*sin(u)+(H*u/B1)*sin(u);
gx1=a*cos(u)+b*cos(u)+(H/B1)*sin(u)+(H*u/B1)*cos(u);
gy=a*cos(u)+b*cos(u)+(H*u/B1)*cos(u);
gy1=-a*sin(u)-b*sin(u)+(H/B1)*cos(u)-(H*u/B1)*sin(u);}

if((u>ang1)&&(u<=ang2))
{gx=a*sin(u)+b*sin(u)+H1*(1-u/B2)*sin(u);
gx1=a*cos(u)+b*cos(u)-(H1/B2)*sin(u)+H1*cos(u)-(H1*u/B2)*cos(u);
gy=a*cos(u)+b*cos(u)+H1*(1-u/B2)*cos(u);
gy1=-a*sin(u)-b*sin(u)-(H1/B2)*cos(u)-H1*sin(u)+(H1*u/B2)*sin(u);}

if((u>ang2)&&(u<=ang3))

```

```

{gx=a*sin(u)+b*cos(u)+(H*u/B3)*sin(u);
gx1=a*cos(u)+b*cos(u)+(H/B3)*sin(u)+(H*u/B3)*cos(u);
gy=a*cos(u)+b*cos(u)+(H*u/B3)*cos(u);
gy1=-a*sin(u)-b*sin(u)+(H/B3)*cos(u)-(H*u/B3)*sin(u);}

if((u>ang3)&&(u<=ang4))
{gx=a*sin(u)+b*sin(u)+H1*(1-u/B4)*sin(u);
gx1=a*cos(u)+b*cos(u)-(H1/B4)*sin(u)+H1*cos(u)-(H1*u/B4)*cos(u);
gy=a*cos(u)+b*cos(u)+H1*(1-u/B4)*cos(u);
gy1=-a*sin(u)-b*sin(u)-(H1/B4)*cos(u)-H1*sin(u)+(H1*u/B4)*sin(u);}

if((u>ang4)&&(u<=TWOPIE))
{gx=0.0;gx1=0.0;gy=0.0;gy1=0.0;}

xp=-Rmin*sin(u)-gx*sin(u)+gx1*cos(u)-gy*sin(u)+gy1*cos(u);
yp=Rmin*cos(u)+gx*cos(u)+gx1*sin(u)+gy*cos(u)+gy1*sin(u);
zp=0.0;
if(xp>=0){ xDist=fabs(x-X-STEP);XF=10000;}
else{ xDist=fabs(x-X+STEP);XF=0;}
if(yp>=0){ yDist=fabs(y-Y-STEP);YF=1000;}
else{ yDist=fabs(y-Y+STEP);YF=0;}
if(zp>=0){ zDist=fabs(z-Z-STEP);ZF=100;}
else{ zDist=fabs(z-Z+STEP);ZF=0;}
if(fabs(yp*xDist)>=fabs(xp*yDist))GF1=10;
else GF1=0;
if(SQR(zp)*(SQR(xDist)+SQR(yDist))>=SQR(zDist)*(SQR(xp)+SQR(yp)))
GF2=1;
else
GF2=0;
SelectedPlane=XF+YF+ZF+GF1+GF2;
switch(SelectedPlane)
{
case(11000):case(11100):case(10100):case(10000):u=u+(xDist/xp);break;//Plane1
case(11010):case(11110):case(11110):u=u+(yDist/yp);break;
//Plane2
case(1100):case(1000):case(100):case(0):u=u-(xDist/xp);break;
//Plane3
case(10110):case(10010):case(1010):case(110):case(10):u=u-
(yDist/yp);break;//Plane4
case(11111):case(11101):case(10111):case(10101):case(1111):case(1101):case(
1011):
case(1001):case(111):case(101):u=u+(zDist/zp);break; //Plane5
case(11001):case(10011):case(10001):case(11):case(1):case(11011):u=u-
(zDist/zp);break;//Plane6
default :u=1000;break;
}

x=Rmin*cos(u)+gx*cos(u)-gy*sin(u);

```



```

y=Rmin*sin(u)+gy*sin(u)-gx*cos(u);
z=0.0;
xtool=x+(yp*Rtool)/sqrt(SQR(xp)+SQR(yp));
ytool=y-(xp*Rtool)/sqrt(SQR(xp)+SQR(yp));
ztool=0.0;

DX=x-X;DY=y-Y;DZ=z-Z;
if(DX>0.5*STEP)          X=X+STEP;
else if(DX<-0.5*STEP)    X=X-STEP;
else                      X=X;
if(DY>0.5*STEP)          Y=Y+STEP;
else if(DY<-0.5*STEP)    Y=Y-STEP;
else                      Y=Y;
if(DZ>0.5*STEP)          Z=Z+STEP;
else if(DZ<-0.5*STEP)    Z=Z-STEP;
else                      Z=Z;
fprintf (fpOut,"%f\t%f\n",X,Y);
TP++;
}
fclose(fpOut);
printf("Iteration number:%u iterations\n",TP);
}

```

Appendix B2

Program for Constant Acceleration Motion

```

#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
const double TWOPIE = 6.2831853072;
const double STEP = 0.5;
const double Rmin = 15.0;
const double Rtool = 5.0;
_inline double SQR(double a) {return pow(a,2);}
void main()
{
FILE *fpOut;
fpOut = fopen("CAM.DAT","w");
double gx,gx1,gy,gy1;
double a=12.7,b=25.4;
double H=15.875,H1=0;
double B1=1.571,B2=3.142,B3=4.712,B4=6.283;
double ang1=1.86,ang2=2.645,ang3=2.911,ang4=4.712;
double xtool=Rmin+Rtool,ytool,ztool;
double X = Rmin,Y=0.0,Z = 0.0;
double x = Rmin,y = 0.0,z = 0.0;
double xp,yp,zp;
double xDist,yDist,zDist;
double u = 0.0;
int XF,YF,ZF;
int GF1,GF2;
int SelectedPlane;
double DX,DY,DZ;
unsigned long TP = 0;
printf("Starting Thesis Program ... \n\n");
while(u<=TWOPIE)
{
if((u>=0)&&(u<=ang1))
{gx=a*sin(u)+b*sin(u)+((2*H*pow(u,2))/pow(B1,2))*sin(u);

gx1=a*cos(u)+b*cos(u)+((2*H*pow(u,2))/pow(B1,2))*cos(u)+((4*H*u)/pow(B1,2))*sin(u);
gy=a*cos(u)+b*cos(u)+((2*H*pow(u,2))/pow(B1,2))*cos(u);
gy1=-a*sin(u)-b*sin(u)-((2*H*pow(u,2))/pow(B1,2))*sin(u)+((4*H*u)/pow(B1,2))*cos(u);}

if((u>ang1)&&(u<=ang2))
{gx=a*sin(u)+b*sin(u)+H1*sin(u)-(2*H1*pow(u,2)/pow(B2,2))*sin(u);
gx1=a*cos(u)+b*cos(u)+H1*cos(u)-(2*H1*pow(u,2)/pow(B2,2))*cos(u)-((4*H1*u)/pow(B2,2))*sin(u);

```

```

gy=a*cos(u)+b*cos(u)+H1*cos(u)-(2*H1*pow(u,2)/pow(B2,2))*cos(u);
gy1=-a*sin(u)-b*sin(u)-H1*sin(u)+((2*H1*pow(u,2))/pow(B2,2))*sin(u)-
((4*H1*u)/pow(B2,2))*cos(u);}

if((u>ang2)&&(u<=ang3))
{gx=a*sin(u)+b*sin(u)+H*sin(u)-2*H*sin(u)+((4*H*u)/B3)*sin(u)-
((2*H*pow(u,2))/pow(B3,2))*sin(u);
gx1=a*cos(u)+b*cos(u)+H*cos(u)-
2*H*cos(u)+((4*H*u)/B3)*cos(u)+((4*H)/B3)*sin(u)
-((2*H*pow(u,2))/pow(B3,2))*cos(u)-((4*H*u)/pow(B3,2))*sin(u);
gy=a*cos(u)+b*cos(u)+H*cos(u)-2*H*cos(u)+((4*H*u)/B3)*cos(u)-
((2*H*pow(u,2))/pow(B3,2))*cos(u);
gy1=-a*sin(u)-b*sin(u)-H*sin(u)+2*H*sin(u)-((4*H*u)/B3)*sin(u)+((4*H)/B3)*cos(u)
+((2*H*pow(u,2))/pow(B3,2))*sin(u)-((4*H*u)/pow(B3,2))*cos(u);}

if((u>ang3)&&(u<=ang4))
{gx=a*sin(u)+b*sin(u)+2*H1*sin(u)-
((4*H1*u)/B4)*sin(u)+((2*H1*pow(u,2))/pow(B4,2))*sin(u);
gx1=a*cos(u)+b*cos(u)+2*H1*cos(u)-((4*H1*u)/B4)*cos(u)-((4*H1)/B4)*sin(u)
+((2*H1*pow(u,2))/pow(B4,2))*cos(u)+((4*H1*u)/pow(B4,2))*sin(u);
gy=a*cos(u)+b*cos(u)+2*H1*cos(u)-
((4*H1*u)/B4)*cos(u)+((2*H1*pow(u,2))/pow(B4,2))*cos(u);
gy1=-a*sin(u)-b*sin(u)-2*H1*sin(u)+((4*H1*u)/B4)*sin(u)-((4*H1)/B4)*cos(u)
-((2*H1*pow(u,2))/pow(B4,2))*sin(u)+((4*H1*u)/pow(B4,2))*cos(u);}

if((u>ang4)&&(u<=TWOPIE))
{gx=0.0;gx1=0.0;gy=0.0;gy1=0.0;}

xp=-Rmin*sin(u)-gx*sin(u)+gx1*cos(u)-gy*sin(u)+gy1*cos(u);
yp=Rmin*cos(u)+gx*cos(u)+gx1*sin(u)+gy*cos(u)+gy1*sin(u);
zp=0.0;
if(xp>=0){xDist=fabs(x-X-STEP);XF=10000;}
else{xDist=fabs(x-X+STEP);XF=0;}
if(yp>=0){yDist=fabs(y-Y-STEP);YF=1000;}
else{yDist=fabs(y-Y+STEP);YF=0;}
if(zp>=0){zDist=fabs(z-Z-STEP);ZF=100;}
else{zDist=fabs(z-Z+STEP);ZF=0;}
if(fabs(yp*xDist)>=fabs(xp*yDist))GF1=10;
else GF1=0;
if(SQR(zp)*(SQR(xDist)+SQR(yDist))>=SQR(zDist)*(SQR(xp)+SQR(yp)))
GF2=1;
else
GF2=0;
SelectedPlane=XF+YF+ZF+GF1+GF2;
switch(SelectedPlane)
{
case(11000):case(11100):case(10100):case(10000):u=u+(xDist/xp);break;
//Plane1

```

```

        case(11010):case(1110):case(11110):u=u+(yDist/yp);break;
                //Plane2
        case(1100):case(1000):case(100):case(0):u=u-(xDist/xp);break;
        //Plane3
        case(10110):case(10010):case(1010):case(110):case(10):u=u-
(yDist/yp);break;//Plane4
        case(11111):case(11101):case(10111):case(10101):case(1111):case(1101):case(
1011):
        case(1001):case(111):case(101):u=u+(zDist/zp);break;    //Plane5
        case(11001):case(10011):case(10001):case(11):case(1):case(11011):u=u-
(zDist/zp);break;//Plane6
        default :u=1000;break;
}

x=Rmin*cos(u)+gx*cos(u)-gy*sin(u);
y=Rmin*sin(u)+gy*sin(u)-gx*cos(u);
z=0.0;
xtool=x+(yp*Rtool)/sqrt(SQR(xp)+SQR(yp));
ytool=y-(xp*Rtool)/sqrt(SQR(xp)+SQR(yp));
ztool=0.0;

DX=x-X;DY=y-Y;DZ=z-Z;
if(DX>0.5*STEP)          X=X+STEP;
else if(DX<-0.5*STEP)   X=X-STEP;
else                     X=X;
if(DY>0.5*STEP)          Y=Y+STEP;
else if(DY<-0.5*STEP)   Y=Y-STEP;
else                     Y=Y;
if(DZ>0.5*STEP)          Z=Z+STEP;
else if(DZ<-0.5*STEP)   Z=Z-STEP;
else                     Z=Z;
fprintf (fpOut,"%f\t%f\n",X,Y);
TP++;
}
fclose(fpOut);
printf("Iteration number:%u iterations\n",TP);
}

```

Appendix B3

Program for Simple Harmonics Motion

```

#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
const double TWOPIE = 6.2831853072;
const double STEP = 0.5;
const double Rmin = 15.0;
const double Rtool = 5.0;
_inline double SQR(double a) {return pow(a,2);}
void main()
{
FILE *fpOut;
fpOut = fopen("SHM.DAT","w");
double gx,gx1,gy,gy1;
double a=12.7,b=25.4;
double H=15.875,H1=0;
double pi=3.142;
double B1=1.571,B2=3.142,B3=4.712,B4=6.283;
double ang1=1.86,ang2=2.645,ang3=2.911,ang4=4.712;
double xtool=Rmin+Rtool,ytool,ztool;
double X = Rmin,Y=0.0,Z = 0.0;
double x = Rmin,y = 0.0,z = 0.0;
double xp,yp,zp;
double xDist,yDist,zDist;
double u = 0.0;
int XF,YF,ZF;
int GF1,GF2;
int SelectedPlane;
double DX,DY,DZ;
unsigned long TP = 0;
printf("Starting Thesis Program ... \n\n");
while(u<=TWOPIE)
{
if((u>=0)&&(u<=ang1))
{gx=a*sin(u)+b*sin(u)+(H/2)*sin(u)-((H/2)*cos(pi*u/B1))*sin(u);
gx1=a*cos(u)+b*cos(u)+(H/2)*cos(u)-
((H/2)*cos(pi*u/B1))*cos(u)+(((H*pi)/(2*B1))*sin(pi*u/B1))*sin(u);
gy=a*cos(u)+b*cos(u)+(H/2)*cos(u)-((H/2)*cos(pi*u/B1))*cos(u);
gy1=-a*sin(u)-b*sin(u)-
(H/2)*sin(u)+((H/2)*cos(pi*u/B1))*sin(u)+(((H*pi)/(2*B1))*sin(pi*u/B1))*cos(u);}

if((u>ang1)&&(u<=ang2))
{gx=a*sin(u)+b*sin(u)+(H1/2)*sin(u)+((H/2)*cos(pi*u/B2))*sin(u);
gx1=a*cos(u)+b*cos(u)+(H1/2)*cos(u)+((H/2)*cos(pi*u/B2))*cos(u)-
(((H1*pi)/(2*B2))*sin(pi*u/B2))*sin(u);

```

```

gy=a*cos(u)+b*cos(u)+(H1/2)*cos(u)+((H/2)*cos(pi*u/B2))*cos(u);
gy1=-a*sin(u)-b*sin(u)-(H1/2)*sin(u)-((H/2)*cos(pi*u/B2))*sin(u)-
(((H1*pi)/(2*B2))*sin(pi*u/B2))*cos(u);}

if((u>ang2)&&(u<=ang3))
{gx=a*sin(u)+b*sin(u)+(H/2)*sin(u)-((H/2)*cos(pi*u/B3))*sin(u);
gx1=a*cos(u)+b*cos(u)+(H/2)*cos(u)-
((H/2)*cos(pi*u/B3))*cos(u)+(((H*pi)/(2*B3))*sin(pi*u/B3))*sin(u);
gy=a*cos(u)+b*cos(u)+(H/2)*cos(u)-((H/2)*cos(pi*u/B3))*cos(u);
gy1=-a*sin(u)-b*sin(u)-
(H/2)*sin(u)+((H/2)*cos(pi*u/B3))*sin(u)+(((H*pi)/(2*B3))*sin(pi*u/B3))*cos(u);}

if((u>ang3)&&(u<=ang4))
{gx=a*sin(u)+b*sin(u)+(H1/2)*sin(u)+((H/2)*cos(pi*u/B4))*sin(u);
gx1=a*cos(u)+b*cos(u)+(H1/2)*cos(u)+((H/2)*cos(pi*u/B4))*cos(u)-
(((H1*pi)/(2*B4))*sin(pi*u/B4))*sin(u);
gy=a*cos(u)+b*cos(u)+(H1/2)*cos(u)+((H/2)*cos(pi*u/B4))*cos(u);
gy1=-a*sin(u)-b*sin(u)-(H1/2)*sin(u)-((H/2)*cos(pi*u/B4))*sin(u)-
(((H1*pi)/(2*B4))*sin(pi*u/B4))*cos(u);}

if((u>ang4)&&(u<=TWOPIE))
{gx=0.0;gx1=0.0;gy=0.0;gy1=0.0;}

xp=-Rmin*sin(u)-gx*sin(u)+gx1*cos(u)-gy*sin(u)+gy1*cos(u);
yp=Rmin*cos(u)+gx*cos(u)+gx1*sin(u)+gy*cos(u)+gy1*sin(u);
zp=0.0;
if(xp>=0){xDist=fabs(x-X-STEP);XF=10000;}
else{xDist=fabs(x-X+STEP);XF=0;}
if(yp>=0){yDist=fabs(y-Y-STEP);YF=1000;}
else{yDist=fabs(y-Y+STEP);YF=0;}
if(zp>=0){zDist=fabs(z-Z-STEP);ZF=100;}
else{zDist=fabs(z-Z+STEP);ZF=0;}
if(fabs(yp*xDist)>=fabs(xp*yDist))GF1=10;
else GF1=0;
if(SQR(zp)*(SQR(xDist)+SQR(yDist))>=SQR(zDist)*(SQR(xp)+SQR(yp)))
GF2=1;
else
GF2=0;
SelectedPlane=XF+YF+ZF+GF1+GF2;
switch(SelectedPlane)
{
case(11000):case(11100):case(10100):case(10000):u=u+(xDist/xp);break;
//Plane1
case(11010):case(1110):case(11110):u=u+(yDist/yp);break;
//Plane2
case(1100):case(1000):case(100):case(0):u=u-(xDist/xp);break;
//Plane3
case(10110):case(10010):case(1010):case(110):case(10):u=u-
(yDist/yp);break;//Plane4

```

```

        case(11111):case(11101):case(10111):case(10101):case(1111):case(1101):case(
1011):
        case(1001):case(111):case(101):u=u+(zDist/zp);break;    //Plane5
        case(11001):case(10011):case(10001):case(11):case(1):case(11011):u=u-
(zDist/zp);break;//Plane6
        default :u=1000;break;
    }

    x=Rmin*cos(u)+gx*cos(u)-gy*sin(u);
    y=Rmin*sin(u)+gy*sin(u)-gx*cos(u);
    z=0.0;
    xtool=x+(yp*Rtool)/sqrt(SQR(xp)+SQR(yp));
    ytool=y-(xp*Rtool)/sqrt(SQR(xp)+SQR(yp));
    ztool=0.0;

    DX=x-X;DY=y-Y;DZ=z-Z;
    if(DX>0.5*STEP)          X=X+STEP;
    else if(DX<-0.5*STEP)    X=X-STEP;
    else                      X=X;
    if(DY>0.5*STEP)          Y=Y+STEP;
    else if(DY<-0.5*STEP)    Y=Y-STEP;
    else                      Y=Y;
    if(DZ>0.5*STEP)          Z=Z+STEP;
    else if(DZ<-0.5*STEP)    Z=Z-STEP;
    else                      Z=Z;
    fprintf (fpOut,"%f\t%f\n",X,Y);
    TP++;
}
fclose(fpOut);
printf("Iteration number:%u iterations\n",TP);
}

```

Appendix B4

Program for Cycloidal Motion

```

#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
const double TWOPIE = 6.2831853072;
const double STEP = 0.5;
const double Rmin = 15.0;
const double Rtool = 5.0;
_inline double SQR(double a) {return pow(a,2);}
void main()
{
FILE *fpOut;
fpOut = fopen("CYM.DAT","w");
double gx,gx1,gy,gy1;
double a=12.7,b=25.4;
double H=15.875,H1=0;
double pi=3.142;
double B1=1.571,B2=3.142,B3=4.712,B4=6.283;
double ang1=1.86,ang2=2.645,ang3=2.911,ang4=4.712;
double xtool=Rmin+Rtool,ytool,ztool;
double X = Rmin,Y=0.0,Z = 0.0;
double x = Rmin,y = 0.0,z = 0.0;
double xp,yp,zp;
double xDist,yDist,zDist;
double u = 0.0;
int XF,YF,ZF;
int GF1,GF2;
int SelectedPlane;
double DX,DY,DZ;
unsigned long TP = 0;
printf("Starting Thesis Program ... \n\n");
while(u<=TWOPIE)
{
if((u>=0)&&(u<=ang1))
{gx=a*sin(u)+b*sin(u)+(H*u/B1)*sin(u)-((H/2*pi)*sin(2*pi*u/B1))*sin(u);
gx1=a*cos(u)+b*cos(u)+(H*u/B1)*cos(u)+(H/B1)*sin(u)-
((H/2*pi)*sin(2*pi*u/B1))*cos(u)-((H/B1)*cos(2*pi*u/B1))*sin(u);
gy=a*cos(u)+b*cos(u)+(H*u/B1)*cos(u)-((H/2*pi)*sin(2*pi*u/B1))*cos(u);
gy1=-a*sin(u)-b*sin(u)-
(H*u/B1)*sin(u)+(H/B1)*cos(u)+((H/2*pi)*sin(2*pi*u/B1))*sin(u)-
((H/B1)*cos(2*pi*u/B1))*cos(u);}

if((u>ang1)&&(u<=ang2))
{gx=a*sin(u)+b*sin(u)+H1*sin(u)-
(H1*u/B2)*sin(u)+((H1/2*pi)*sin(2*pi*u/B2))*sin(u);

```



```

gx1=a*cos(u)+b*cos(u)+H1*cos(u)-(H1*u/B2)*cos(u)-
(H1/B2)*sin(u)+((H1/2*pi)*sin(2*pi*u/B2))*cos(u)+((H1/B2)*cos(2*pi*u/B2))*sin(u);
gy=a*cos(u)+b*cos(u)+H1*cos(u)-
(H1*u/B2)*cos(u)+((H1/2*pi)*sin(2*pi*u/B2))*cos(u);
gy1=-a*sin(u)-b*sin(u)-H1*sin(u)+(H1*u/B2)*sin(u)-(H1/B2)*cos(u)-
((H1/2*pi)*sin(2*pi*u/B2))*sin(u)+((H1/B2)*cos(2*pi*u/B2))*cos(u);

if((u>ang2)&&(u<=ang3))
{gx=a*sin(u)+b*sin(u)+(H*u/B3)*sin(u)-((H/2*pi)*sin(2*pi*u/B3))*sin(u);
gx1=a*cos(u)+b*cos(u)+(H*u/B3)*cos(u)+(H/B3)*sin(u)-
((H/2*pi)*sin(2*pi*u/B3))*cos(u)-((H/B3)*cos(2*pi*u/B3))*sin(u);
gy=a*cos(u)+b*cos(u)+(H*u/B3)*cos(u)-((H/2*pi)*sin(2*pi*u/B3))*cos(u);
gy1=-a*sin(u)-b*sin(u)-
(H*u/B3)*sin(u)+(H/B3)*cos(u)+((H/2*pi)*sin(2*pi*u/B3))*sin(u)-
((H/B3)*cos(2*pi*u/B3))*cos(u);}

if((u>ang3)&&(u<=ang4))
{gx=a*sin(u)+b*sin(u)+H1*sin(u)-
(H1*u/B4)*sin(u)+((H1/2*pi)*sin(2*pi*u/B4))*sin(u);
gx1=a*cos(u)+b*cos(u)+H1*cos(u)-(H1*u/B4)*cos(u)-
(H1/B4)*sin(u)+((H1/2*pi)*sin(2*pi*u/B4))*cos(u)+((H1/B4)*cos(2*pi*u/B4))*sin(u);
gy=a*cos(u)+b*cos(u)+H1*cos(u)-
(H1*u/B4)*cos(u)+((H1/2*pi)*sin(2*pi*u/B4))*cos(u);
gy1=-a*sin(u)-b*sin(u)-H1*sin(u)+(H1*u/B4)*sin(u)-(H1/B4)*cos(u)-
((H1/2*pi)*sin(2*pi*u/B4))*sin(u)+((H1/B4)*cos(2*pi*u/B4))*cos(u);}

if((u>ang4)&&(u<=TWOPIE))
{gx=0.0;gx1=0.0;gy=0.0;gy1=0.0;}

xp=-Rmin*sin(u)-gx*sin(u)+gx1*cos(u)-gy*sin(u)+gy1*cos(u);
yp=Rmin*cos(u)+gx*cos(u)+gx1*sin(u)+gy*cos(u)+gy1*sin(u);
zp=0.0;
if(xp>=0){xDist=fabs(x-X-STEP);XF=10000;}
else{xDist=fabs(x-X+STEP);XF=0;}
if(yp>=0){yDist=fabs(y-Y-STEP);YF=1000;}
else{yDist=fabs(y-Y+STEP);YF=0;}
if(zp>=0){zDist=fabs(z-Z-STEP);ZF=100;}
else{zDist=fabs(z-Z+STEP);ZF=0;}
if(fabs(yp*xDist)>=fabs(xp*yDist))GF1=10;
else GF1=0;
if(SQR(zp)*(SQR(xDist)+SQR(yDist))>=SQR(zDist)*(SQR(xp)+SQR(yp)))
GF2=1;
else
GF2=0;
SelectedPlane=XF+YF+ZF+GF1+GF2;
switch(SelectedPlane)
{
case(11000):case(11100):case(10100):case(10000):u=u+(xDist/xp);break;
//Plane1

```

```

        case(11010):case(1110):case(11110):u=u+(yDist/yp);break;
                //Plane2
        case(1100):case(1000):case(100):case(0):u=u-(xDist/xp);break;
        //Plane3
        case(10110):case(10010):case(1010):case(110):case(10):u=u-
(yDist/yp);break;//Plane4
        case(11111):case(11101):case(10111):case(10101):case(1111):case(1101):case(
1011):
        case(1001):case(111):case(101):u=u+(zDist/zp);break;    //Plane5
        case(11001):case(10011):case(10001):case(11):case(1):case(11011):u=u-
(zDist/zp);break;//Plane6
        default :u=1000;break;
}

x=Rmin*cos(u)+gx*cos(u)-gy*sin(u);
y=Rmin*sin(u)+gy*sin(u)-gx*cos(u);
z=0.0;
xtool=x+(yp*Rtool)/sqrt(SQR(xp)+SQR(yp));
ytool=y-(xp*Rtool)/sqrt(SQR(xp)+SQR(yp));
ztool=0.0;

DX=x-X;DY=y-Y;DZ=z-Z;
if(DX>0.5*STEP)          X=X+STEP;
else if(DX<-0.5*STEP)    X=X-STEP;
else                      X=X;
if(DY>0.5*STEP)          Y=Y+STEP;
else if(DY<-0.5*STEP)    Y=Y-STEP;
else                      Y=Y;
if(DZ>0.5*STEP)          Z=Z+STEP;
else if(DZ<-0.5*STEP)    Z=Z-STEP;
else                      Z=Z;
fprintf (fpOut,"%f\t%f\n",X,Y);
TP++;
}
fclose(fpOut);
printf("Iteration number:%u iterations\n",TP);
}

```

Appendix B5

Program for Combined Motion

```

#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
const double TWOPIE = 6.2831853072;
const double STEP = 0.5;
const double Rmin = 15.0;
const double Rtool = 5.0;
_inline double SQR(double a) {return pow(a,2);}
void main()
{
FILE *fpOut;
fpOut = fopen("4in1.DAT","w");
double gx,gx1,gy,gy1;
double a=12.7,b=25.4;
double H=15.875,H1=0;
double pi=3.142;
double B1=1.571,B2=3.142,B3=4.712,B4=6.283;
double ang1=1.86,ang2=2.645,ang3=2.911,ang4=4.712;
double xtool=Rmin+Rtool,ytool,ztool;
double X = Rmin,Y=0.0,Z = 0.0;
double x = Rmin,y = 0.0,z = 0.0;
double xp,yp,zp;
double xDist,yDist,zDist;
double u = 0.0;
int XF,YF,ZF;
int GF1,GF2;
int SelectedPlane;
double DX,DY,DZ;
unsigned long TP = 0;
printf("Starting Thesis Program ... \n\n");
while(u<=TWOPIE)
{
if((u>=0)&&(u<=ang1))

{gx=a*sin(u)+b*cos(u)+(H*u/B1)*sin(u);gx1=a*cos(u)+b*cos(u)+(H/B1)*sin(u)+(H*u/
B1)*cos(u);
gy=a*cos(u)+b*cos(u)+(H*u/B1)*cos(u);gy1=-a*sin(u)-b*sin(u)+(H/B1)*cos(u)-
(H*u/B1)*sin(u);}

if((u>ang1)&&(u<=ang2))
{gx=a*sin(u)+b*cos(u)+H1*sin(u)-(2*H1*pow(u,2)/pow(B2,2))*sin(u);
gx1=a*cos(u)+b*cos(u)+H1*cos(u)-(2*H1*pow(u,2)/pow(B2,2))*cos(u)-
((4*H1*u)/pow(B2,2))*sin(u);
gy=a*cos(u)+b*cos(u)+H1*cos(u)-(2*H1*pow(u,2)/pow(B2,2))*cos(u);

```

```

    gy1=-a*sin(u)-b*sin(u)-H1*sin(u)+((2*H1*pow(u,2))/pow(B2,2))*sin(u)-
    ((4*H1*u)/pow(B2,2))*cos(u);}

    if((u>ang2)&&(u<=ang3))
    {gx=a*sin(u)+b*sin(u)+(H/2)*sin(u)-((H/2)*cos(pi*u/B3))*sin(u);
    gx1=a*cos(u)+b*cos(u)+(H/2)*cos(u)-
    ((H/2)*cos(pi*u/B3))*cos(u)+((H*pi)/(2*B3))*sin(pi*u/B3))*sin(u);
    gy=a*cos(u)+b*cos(u)+(H/2)*cos(u)-((H/2)*cos(pi*u/B3))*cos(u);
    gy1=-a*sin(u)-b*sin(u)-
    (H/2)*sin(u)+((H/2)*cos(pi*u/B3))*sin(u)+((H*pi)/(2*B3))*sin(pi*u/B3))*cos(u);}

    if((u>ang3)&&(u<=ang4))
    {gx=a*sin(u)+b*sin(u)+H1*sin(u)-
    (H1*u/B4)*sin(u)+((H1/2*pi)*sin(2*pi*u/B4))*sin(u);
    gx1=a*cos(u)+b*cos(u)+H1*cos(u)-(H1*u/B4)*cos(u)-
    (H1/B4)*sin(u)+((H1/2*pi)*sin(2*pi*u/B4))*cos(u)+((H1/B4)*cos(2*pi*u/B4))*sin(u);
    gy=a*cos(u)+b*cos(u)+H1*cos(u)-
    (H1*u/B4)*cos(u)+((H1/2*pi)*sin(2*pi*u/B4))*cos(u);
    gy1=-a*sin(u)-b*sin(u)-H1*sin(u)+(H1*u/B4)*sin(u)-(H1/B4)*cos(u)-
    ((H1/2*pi)*sin(2*pi*u/B4))*sin(u)+((H1/B4)*cos(2*pi*u/B4))*cos(u);}

    if((u>ang4)&&(u<=TWOPIE))
    {gx=0.0;gx1=0.0;gy=0.0;gy1=0.0;}

    xp=-Rmin*sin(u)-gx*sin(u)+gx1*cos(u)-gy*sin(u)+gy1*cos(u);
    yp=Rmin*cos(u)+gx*cos(u)+gx1*sin(u)+gy*cos(u)+gy1*sin(u);
    zp=0.0;
    if(xp>=0){xDist=fabs(x-X-STEP);XF=10000;}
    else{xDist=fabs(x-X+STEP);XF=0;}
    if(yp>=0){yDist=fabs(y-Y-STEP);YF=1000;}
    else{yDist=fabs(y-Y+STEP);YF=0;}
    if(zp>=0){zDist=fabs(z-Z-STEP);ZF=100;}
    else{zDist=fabs(z-Z+STEP);ZF=0;}
    if(fabs(yp*xDist)>=fabs(xp*yDist))GF1=10;
    else GF1=0;
    if(SQR(zp)*(SQR(xDist)+SQR(yDist))>=SQR(zDist)*(SQR(xp)+SQR(yp)))
    GF2=1;
    else
    GF2=0;
    SelectedPlane=XF+YF+ZF+GF1+GF2;
    switch(SelectedPlane)
    {
        case(11000):case(11100):case(10100):case(10000):u=u+(xDist/xp);break;
        //Plane1
        case(11010):case(1110):case(11110):u=u+(yDist/yp);break;
        //Plane2
        case(1100):case(1000):case(100):case(0):u=u-(xDist/xp);break;
        //Plane3
    }

```

```

        case(10110):case(10010):case(1010):case(110):case(10):u=u-
(yDist/yp);break;//Plane4
        case(11111):case(11101):case(10111):case(10101):case(1111):case(1101):case(
1011):
        case(1001):case(111):case(101):u=u+(zDist/zp);break;    //Plane5
        case(11001):case(10011):case(10001):case(11):case(1):case(11011):u=u-
(zDist/zp);break;//Plane6
        default :u=1000;break;
    }

x=Rmin*cos(u)+gx*cos(u)-gy*sin(u);
y=Rmin*sin(u)+gy*sin(u)-gx*cos(u);
z=0.0;
xtool=x+(yp*Rtool)/sqrt(SQR(xp)+SQR(yp));
ytool=y-(xp*Rtool)/sqrt(SQR(xp)+SQR(yp));
ztool=0.0;

DX=x-X;DY=y-Y;DZ=z-Z;
if(DX>0.5*STEP)        X=X+STEP;
else if(DX<-0.5*STEP)  X=X-STEP;
else                    X=X;
if(DY>0.5*STEP)        Y=Y+STEP;
else if(DY<-0.5*STEP)  Y=Y-STEP;
else                    Y=Y;
if(DZ>0.5*STEP)        Z=Z+STEP;
else if(DZ<-0.5*STEP)  Z=Z-STEP;
else                    Z=Z;
fprintf (fpOut,"%f\t%f\n",X,Y);
TP++;
}
fclose(fpOut);
printf("Iteration number:%u iterations\n",TP);
}

```