

104
81

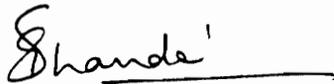
SYMBOLIC AND COMPUTATIONAL CONJUGATE GEOMETRY
FOR
DESIGN AND MANUFACTURING APPLICATIONS

by

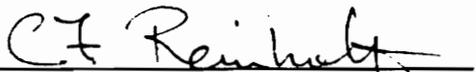
Ravinder Srinivas Voruganti

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of
Master of Science
in
Mechanical Engineering

APPROVED:



Dr. S. G. Dhande, Co-Chairman



Dr. C. F. Reinholtz, Co-Chairman



Dr. H. H. Robertshaw

August, 1990
Blacksburg, Virginia

LD

5655

V855

1990

V673

C.2

Symbolic and Computational Conjugate Geometry for Design and Manufacturing Applications

by

Ravinder Srinivas Voruganti

S. G. Dhande & C. F. Reinholtz, Co-Chairmen

Mechanical Engineering

(ABSTRACT)

Development of a methodology of applying symbolic and computational models of conjugate geometry to several problems in design and manufacturing areas forms the basis of this study. The philosophy of this work is the development of generalized symbolic models for two-dimensional and three-dimensional conjugate geometry applications. The generalized models have been implemented using what seems to be the best tool for these kind of applications - a symbolic manipulation system. The unique feature of this research is reflected in the fruitful combination of the elegance of conjugate geometry theory and the inherent versatility of a symbolic manipulation system.

The generalized conjugate geometry algorithms were programmed and run using MACSYMA. Typical cases in design of mechanisms have been studied using these symbolic programs. The manufacture of helically swept surfaces is of special interest to this work. Helically swept surfaces have been designed and three schemes of manufacturing these surfaces are presented here. Examples of these three schemes of manufacturing helically swept surfaces have been carried out using the symbolic program in MACSYMA. The results of all the examples have been presented both analytically and graphically.

Acknowledgements

I would like to express my heartfelt gratitude to Dr. Sanjay Dhande and Dr. Charles Reinholtz for their guidance and unfailing support during the course of this research work. I thank them for giving me the opportunity to work on this topic. I am also thankful to Dr. Harry Robertshaw for his help and for making MACSYMA available in the mechanical engineering department.

I would also like to thank my friends Kannan Marudachalam and Juzer Mohammedshah for their unfailing presence and moral support in some tense moments.

Finally, I sincerely thank my parents without whose full support this thesis would have been just another dream.

Table of Contents

Introduction	1
1.1 Motivation	1
1.2 Conjugate Geometry	2
1.2.1 Envelopes	3
1.2.2 Conjugate Curves/Surfaces	5
1.3 Applications of Conjugate Curves/Surfaces	5
1.3.1 Design of Mechanisms	5
1.3.2 Manufacturing	9
1.4 Literature Review	11
1.4.1 Symbolic and Algebraic Manipulation	11
1.4.2 Mechanisms	13
1.4.3 Computational Geometry and Manufacturing	13
1.5 Objective and Organization of Research	15
Symbolic Conjugate Geometry Model Development	19
2.1 Two-Dimensional Case Formulation	19
2.1.1 Condition of Contact	24
2.1.2 Symbolic Algorithm for Two-Dimensional Case	26

2.2 Three-Dimensional Case Formulation	29
2.2.1 Condition of Contact	32
2.2.2 Symbolic Algorithm for Three-Dimensional Case	34
2.3 MACSYMA	38
Case Studies in Conjugate Geometry (Computational and Graphical Models)	40
3.1 Disc Cams	40
3.1.1 Disc Cam with Translating Flat-Faced Follower (TFFF)	41
3.1.2 Disc Cam with Translating Roller-Ended Follower (TREF)	46
3.1.3 Disc Cam with Oscillating Flat-Faced Follower (OFFF)	48
3.1.4 Disc Cam with Oscillating Roller-Ended Follower (OREF)	54
3.2 Gear Tooth Profile	58
3.3 Plane Generation Using Cylinder	62
Extruder Screw Conjugate Geometries	66
4.1 Helically Swept Surfaces	66
4.2 Side-Mill Machining	69
4.3 End-Mill Machining	73
4.4 Wheel-Mill Machining	77
Conclusions	84
5.1 Summary	84
5.2 Recommendations	86
5.3 Future Work	87
Bibliography	88
Appendix A. MACSYMA Session	93

Vita 105

List of Illustrations

Figure 1.	ENVELOPES - A Few Examples	4
Figure 2.	CAM MECHANISM - An Example	8
Figure 3.	Organization of Research (Two Dimensional Cases)	17
Figure 4.	Organization of Research (Three Dimensional Cases)	18
Figure 5.	Two-Dimensional Conjugate Geometry Model	20
Figure 6.	Three-Dimensional Conjugate Geometry Model - Coordinate Systems	30
Figure 7.	DISC CAM - Translating Flat-Faced Follower	42
Figure 8.	Generated Graphical Model of a Disc Cam Profile (TFFF)	45
Figure 9.	DISC CAM - Translating Roller-Ended Follower	47
Figure 10.	Generated Graphical Model of a Disc Cam Profile (TREF)	49
Figure 11.	DISC CAM - Oscillating Flat-Faced Follower	50
Figure 12.	Generated Graphical Model of a Disc Cam Profile (OFFF)	53
Figure 13.	DISC CAM - Oscillating Roller-Ended Follower	55
Figure 14.	Generated Graphical Model of a Disc Cam Profile (OREF)	57
Figure 15.	Involute Profile - Coordinate Systems	59
Figure 16.	Generated Graphical Model of Involute Profile	61
Figure 17.	Plane Generation - Coordinate Systems	63
Figure 18.	Generated Graphical Model of Plane Surface Generation (Curve of Intersection)	65
Figure 19.	Helically Swept Surface - Definition	67
Figure 20.	Side-Mill Operation - Coordinate Systems	70
Figure 21.	Side-Mill Operation - Generatrix Curves	74

Figure 22. End-Mill Operation - Coordinate Systems	75
Figure 23. End-Mill Operation - Generatrix Curves	78
Figure 24. Wheel-Mill Operation - Coordinate Systems	79
Figure 25. Wheel-Mill Operation - Generatrix Curves	83

Chapter 1

Introduction

1.1 Motivation

This research was conceived from an industry-sponsored project on the design of cutters for the manufacture of helically swept extruder screw surfaces. Three schemes for the manufacture of extruder screws were identified, namely, side-mill machining, end-mill machining and wheel-mill machining. These three machining schemes will be discussed in detail in Chapter 4. This cutter design problem was recognized to be a conjugate geometry problem where the cutter surface and the extruder screw surface are conjugate surfaces. The question that arose at this stage was that should three separate programs be written for the three schemes of manufacture or can one general program be developed of which the three schemes of manufacture are subsets. It was realized that a general program could be written using a symbolic manipulation system. Thus the cutter design project was formulated as a problem involving three areas of interest, i.e., conjugate geometry theory, symbolic manipulation and cutter design. This particular research work is unique in the fact that it involves the implementation of generalized conjugate geometry algorithms in a symbolic

manipulation system. The advantage with this system is that a particular case study is a subset of the general program and thus a number of cases can be studied by using just one program. The necessity of writing a separate program for each case study is eliminated.

In this research work it was also shown how these general symbolic programs can be used for other conjugate geometry applications in the design environment apart from the extruder screw manufacturing problem.

1.2 Conjugate Geometry

In engineering practice one often comes across situations where two objects are in continuous contact with some kind of constrained relative motion between them. Examples of such situations would be cam-follower, gear-gear, rack-pinion in a design environment and tool-blank interaction in a manufacturing environment. The concept of conjugate geometry applies to situations like these. Conjugate geometry applies to both curves and surfaces. The conjugate geometry concept states that one curve/surface is the conjugate of the other curve/surface under constrained motion. The two curves/surfaces are in contact all the time. The origins of studies in conjugate geometry lie in the theory of envelopes. This will be shown in the next section. Conjugate geometry theory can be applied in both design and manufacturing areas. In the subsequent sections the theory of envelopes and the applications of conjugate geometry in the design and manufacturing world will be discussed.

1.2.1 Envelopes

The mathematical definition of an envelope is given in Faux and Pratt (1983). If a family of curves can be represented by an equation $f(x, y, t) = 0$, where t is time, then the envelope is found by solving the equations

$$f(x, y, t) = 0$$

and

$$\frac{\partial f}{\partial t}(x, y, t) = 0$$

simultaneously.

Consider a circle of fixed radius r . Assume the circle starts moving along a straight line l (the center of the circle lies on l). If snapshots of the circle are taken at regular small intervals, a collection of circles usually called a family of circles will be seen. The above family of circles have bounding curves which are tangent at each point to a circle of the family. The bounding curves for the above family of circles are two straight lines located at a distance r on either side of the line l (Figure 1a). These bounding curves are called envelopes. In general, if a family of curves exists, then a curve that is tangent at each of its points to some curve of the family is called envelope of the given family of curves (Boltyanskii, 1964). The envelope can consist of more than one curve. Another example would be to consider the family of circles of radius r with centers on a given circle of radius R ($R > r$). The envelope of this family is an annular ring with inner radius $R - r$ and outer radius $R + r$. This is shown in Figure 1b. Here is one last simple example. Consider the family consisting of all straight lines passing at a distance R from a given point O (Figure 1c). The envelope of this family of straight lines is clearly the circle with radius R and center O .

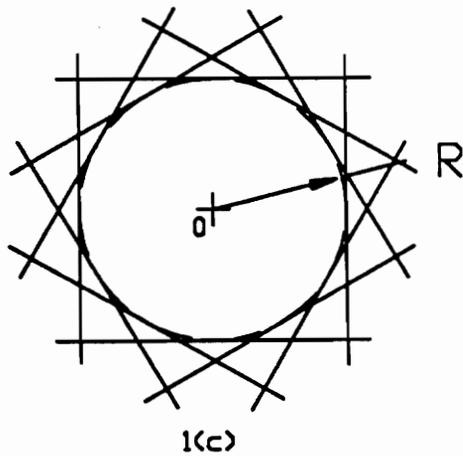
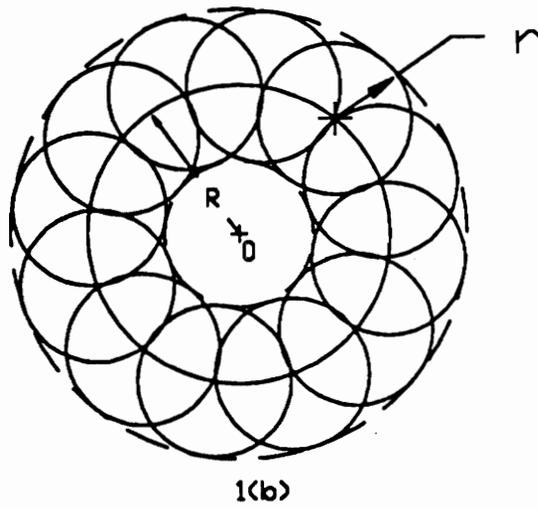
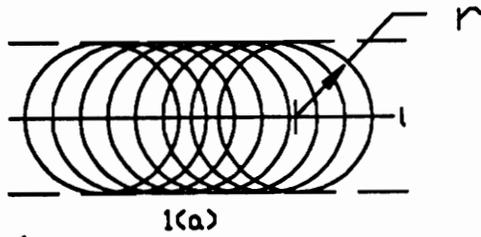


Figure 1. ENVELOPES - A Few Examples

1.2.2 Conjugate Curves/Surfaces

From the discussion on envelopes in the previous section, it is now clear that the envelope of a family of curves and any single curve of the family form a conjugate curve pair. Thus in the example of Figure 1a, the circle and the any one of the bounding straight lines are conjugate curves. Similarly the conjugate curves in the examples of Figures 1b and 1c can be identified.

The preceding discussion on envelopes was restricted to the planar or two dimensional case. The above concept can be extended to three dimensions also. A family of curves could exist in three dimensional space in which case the envelope would be a surface instead of a curve.

1.3 Applications of Conjugate Curves/Surfaces

A few applications of conjugate curves/surfaces in the design and manufacturing areas will be considered.

1.3.1 Design of Mechanisms

Study of mechanisms may be defined as that division of machine design which is concerned with the kinematic design of linkages, cams, gears and gear trains (Mabie and Reinholtz, 1987). Kinematic design is based on motion requirements in contrast to design on the basis of strength requirements. A mechanism is a combination of rigid or resistant bodies so formed and connected that they move upon each other with definite relative motion. Examples of mechanisms include cam-follower, gear-gear, etc.

In dealing with mechanisms, it is necessary to define the various types of motion produced by these mechanisms.

Plane Motion

Translation: When a rigid body moves such that the position of each straight line of the body is parallel to all of its other positions, the body has motion of translation. The two kinds of possible translation are rectilinear translation and curvilinear translation. In rectilinear translation all points of the body move in parallel straight line paths. When a body moves back and forth in this manner, it is said to reciprocate. We will deal mostly with rectilinear translation only. In curvilinear translation the paths of the points are identical curves parallel to a fixed plane.

Rotation: If each point of a rigid body having plane motion remains at a constant distance from a fixed axis that is perpendicular to the plane of motion, the body has motion of rotation. If the body rotates back and forth through a given angle, it is said to oscillate.

Many bodies have motion which is a combination of rotation and translation.

Helical Motion

When a rigid body moves so that each point of the body has motion of rotation about a fixed axis and at the same time has translation parallel to the axis, the body has helical motion. An example of helical motion is the motion of a nut as the nut is screwed onto a bolt. Later on helically swept surfaces will be dealt with, i.e., surfaces that are produced by the helical motion of a rigid body as it is in contact with a tool.

Spherical Motion

When a rigid body moves so that each point of the body has motion about a fixed point while remaining at a constant distance from it, the body has spherical motion.

Spatial Motion

A body moving with rotation about three nonparallel axes and translation along three independent directions is said to be undergoing general spatial motion.

Consider the cam mechanism. A cam mechanism, either two or three-dimensional, is a three link mechanism with the cam and follower in higher-pair contact. In mechanisms with higher-pair contact, the output kinematic quantities depend, in addition to link lengths and relative positions and orientation of the axes, on the nature of the contacting surfaces. Figure 2 shows the sketch of a cam and follower. The cam rotates at a constant angular velocity, and the follower moves up and down. On the upward motion the follower is driven by the cam, and on the return motion it is driven by the action of gravity or of a spring. So it is clear that the follower is in contact with the cam at all times and the follower motion is dependent on the rotation of the cam. Cams are used in many machines, one of the most common being the automotive engine where cams are used to operate the intake and exhaust valves in the cylinders. In a three-dimensional cam, the motion of the follower may depend on both the rotation and the axial motion of the cam.

A cam may be designed in two ways: (a) to assume the required motion for the follower and to design the cam to give this motion, or (b) to assume the shape of the cam and to determine what characteristics of displacement, velocity and acceleration this contour will give. The first method is an example of synthesis. Designing a cam mechanism from the desired motion is an application of synthesis that can be solved every time. The theory of conjugate geometry can be used here to generate the cam profile. The profile of the follower is the known curve in this case. The cam profile is the conjugate curve of the follower profile. With the relative motion between the cam and follower known we can generate the conjugate cam profile curve.

Apart from cam mechanisms, conjugate geometry theory can be applied to other mechanisms too. In considering two curved surfaces in direct contact, it has been shown that the angular-velocity ratio is inversely proportional to the segments into which the line of centers is cut by the line of action or common normal to the two surfaces in contact. If the line of action always intersects the

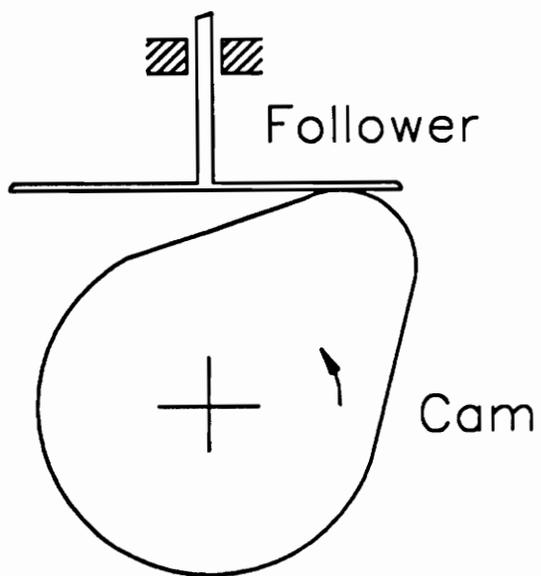


Figure 2. CAM MECHANISM - An Example

line of centers at a fixed point, then the angular-velocity remains constant. This is the condition that is desired when two gear teeth mesh together: the angular-velocity ratio must be constant. It is possible to assume the form of the tooth on one gear and, by applying the above principle (the common normal intersects the line of centers at a fixed point), to determine the outline of the mating tooth. Such teeth would be considered conjugate teeth, and the possibilities are limited only by one's ability to form the teeth. Of the many shapes possible, only the cycloid and the involute have been standardized. The cycloid was used first but has been replaced by the involute for all but a few applications. Involutes are generated from base circles. We can generate the desired involute by applying the principles of conjugate geometry to the base circles, thus producing the teeth of the gears.

Conjugate Geometry theory can be applied to any mechanism with higher pair contact. The cases cited above were a few examples of such mechanisms.

1.3.2 Manufacturing

In the previous section it was shown how conjugate geometry applications can be used in the design of mechanisms having higher pair contact. When the applications of conjugate geometry in manufacturing are explored three-dimensional cases are also considered, apart from two-dimensional cases.

To begin with, consider a two-dimensional case of manufacturing with conjugate geometry applications. Refer back to Figures 1a and 1c. In the manufacturing world, a line is realized using a circle, like in Figure 1a. An example would be to mill a plane surface using a horizontal sidemill cutter. In this example the sidemill is represented as a circle and the plane is represented as a line as the cutter and the plane are projected onto a plane perpendicular to the axis of the cutter. The converse of this case is also true, i.e., a circle is realized in the real world using a straight line. This can be seen in Figure 1c. An example of this case would be a carpenter producing a cylindrical table

leg using a carpenter's plane. Again if this process is projected onto a plane perpendicular to the axis of the table leg, the sketch of Figure 1c is achieved. In this case the carpenter's plane is the straight line and the table leg is the circle in the sketch.

The manufacturing of mechanical parts by machining processes is one of the most natural applications of conjugate geometry. We have the classic case of tool-blank interaction to which we can apply conjugate geometry theory. This would be the best case of three-dimensional conjugate geometry application. The geometry of the finished part depends on both the shape of the tool and the relative motion between the tool and the blank. The cases could involve motion of the tool with the blank stationary, or motion of the blank with the tool stationary, or both the tool and the blank could be moving. From the above discussion it is obvious that there are innumerable cases that could be discussed, but there are a few cases of special interest that will be presented here.

Consider the manufacture of screws for twin-screw extrusion machines. One possible method of manufacturing these screws would be to use the milling process on cylindrical blanks. The cutter could be a spherical endmill and it could be stationary (it would be rotating about its own axis) and the blank would be given a helical motion. As the blank is machined by the endmill, a screw would result, the parameters of which depend on the helical motion given to the blank and the size of the hemispherical end of the mill and the location of the mill. This particular example and a few others are discussed in more detail in Chapter 4.

From this discussion it is clear that manufacturing of components using machining processes is a very good example of conjugate geometry applications.

1.4 Literature Review

1.4.1 Symbolic and Algebraic Manipulation

Symbolic manipulation means manipulation of mathematical expressions which contain free parameters, usually represented by letters (Symbolics, 1989). The MACSYMA reference manual (1989) gives a good comparison of symbolic vs. numeric computation. Computers have been traditionally used to solve scientific problems that could be expressed in terms of numbers. Fortran, for example, assists scientists and mathematicians in dealing with numeric problems. This approach to problem-solving works well when a problem can be easily expressed in terms of calculations with numbers. On the other hand, some problems can be expressed best in symbolic terms, or perhaps can only be expressed that way. A symbolic algebra system can work with symbols, polynomial expressions, equations and numbers. The systems can return results in either numeric or symbolic form. Numeric systems operate using floating-point approximations, whose precision is limited by the computer hardware. By carrying out computations in symbolic form, symbolic systems can work with exact quantities rather than approximations. When you choose to convert a symbolic result to a floating-point number, you can set the precision yourself. The MACSYMA reference manual (1989) illustrates the difference between numeric and symbolic calculation with an example that contrasts the way Fortran and MACSYMA solve a simple problem. Using symbolic methods, many time-consuming jobs can be performed automatically that would otherwise have to be done manually. Symbolic systems can not only expand, factor and simplify long expressions, but they can also perform more complex tasks, such as the manipulation of matrices. Manipulation of matrices is an important function that has been used extensively in the present research work.

Wolfram (1985) demonstrates the particular use of symbolic computations in theoretical physics. But the general concept is applicable to other fields like engineering as well. Standard programming languages are inadequate for the kind of symbolic computations that physicists and engineers need

to perform. Mathematical calculations are an inevitable feature of research in mathematical sciences and engineering. Such calculations can now be done on computers due to the development of interactive computer systems embodying extensive mathematical knowledge and capable of symbolic or algebraic as well as numerical operations. Wolfram (1989) demonstrates the use of the SMP symbolic mathematics system in solving theoretical physics problems.

Raphael et al. (1966) did a survey of computer languages for symbolic and algebraic manipulation. The languages were classified into six different types. For each category the paper describes properties that the members of the group have in common and compares the features of the languages in the group. Raphael (1966), Wilkes (1964) and Green (1961) discuss a few aspects and applications of symbol manipulation. Moses (1971a) gives an examination of algebraic simplification first from the point of view of a user who needs to comprehend a large expression, and second from the point of view of a designer who wants to construct a useful and efficient system. Various techniques akin to substitution are described here. These techniques can be used to decrease the size of an expression and make it more intelligible to a user. Moses (1971b) describes three approaches to symbolic integration. Moses (1974) divides the set of algorithms employed by algebraic manipulation systems into two categories: rational function and non-rational function algorithms. Tobey (1971) presents a specific model depicting the nature of research and development work in symbolic mathematical computation. Basic concepts and issues are adumbrated from the user's point of view, from the mathematician's point of view, and from the designer/implementer's point of view. Martin and Fateman (1971) describe the MACSYMA system which was still in initial development phase then. Martin (1971) gives a very good description of computer input/output of mathematical expressions. It emphasises the importance of implementing the syntax of textbook style mathematical expressions in computer systems. Some applications of algebraic manipulation in celestial mechanics have been described by Jefferys (1971). Hearn (1976) describes the Reduce model for algebraic simplification. The results from a session with a symbolic manipulator are sometimes very lengthy and complex. Stoutemyer (1976) explains how to qualitatively analyze mathematical expressions using computer symbolic mathematics. Hamming (1962) has said "The purpose of computing is insight, not numbers", and computer symbolic mathematics helps fulfill that purpose

by providing analytical rather than numerical answers. Winston (1977) describes LISP as one of the standard vehicles for encoding intelligence-exhibiting processes. He also gives a few examples of symbol manipulations in LISP.

Reviewing the work being done in symbolic manipulations, it appears to be a natural tool for conjugate geometry problems. In the next chapter an algorithm to solve conjugate geometry problems will be presented.

1.4.2 Mechanisms

Chakraborty and Dhande (1977) use modern mathematical tools to solve the kinematic and geometric problems of planar and spatial cam mechanisms. To cover the wide variety of cams, a generalized model is constructed. All the planar and spatial mechanisms are particular cases of this generalized model. This approach makes it possible to amass all important information such as cam coordinates, pressure angle, curvature, etc., of all types of cams in closed analytical forms. Freudenstein (1973), Groot (1970) and Soni (1968) present literature survey on kinematics and cam design. Equations for determining the various disk cam contours have been presented in the analytical cam design material by Mabie and Reinholtz (1987). They have also discussed involute spur gear principles. Dyson (1969) discusses the kinematics and geometry of gears in three-dimensions.

1.4.3 Computational Geometry and Manufacturing

The theory of envelopes is discussed in a book by Boltyanskii (1964). The concept of an envelope is geometric. The book examines several interesting examples of envelopes which arise in the investigation of concrete problems. Some of these problems are taken from kinematics. In the fourth chapter an explanation is given of the general concept of an envelope, together with a general

method of finding envelopes. Forrest (1971) has called computational geometry- "the computer representation, analysis and synthesis of shape information". Faux and Pratt (1983) deal with the mathematical techniques which have been developed in computational geometry. This area of mathematics is now assuming major importance in the manufacturing industries with the increasing use of computer aided design and manufacture, which require the precise geometrical description of the product. Mortenson (1985) presents the nature, development and application of the basic concepts of geometric modeling. Geometric modeling is the technique we use to describe the shape of an object to simulate dynamic processes. Complex shapes can be described as arrangements of simpler ones by using geometric modeling. Geometric modeling provides a model that is analytical, mathematical and abstract rather than concrete. It is often easier and more practical to analyze a model than to test or measure or experiment with the real object. The model is also a useful way of conveying information; for example, a model can be used to transmit design information between engineering and manufacturing functions in industry. Rogers and Adams (1974) present mathematical principles of geometry that are used in three-dimensional geometry-based software packages. Parametric representation of curves and surfaces have been discussed by Farin (1988), Faux and Pratt (1983) and Mortenson (1985).

Of particular interest to this work is the manufacture of extruder screws. Extrusion machines are used in the production of a variety of materials, including bread dough and plastics. The extruder element or screw is the central component of any extrusion machine. Several different types of extrusion machines and a wide assortment of screw elements are in common use. The examples presented in this research work deal with long lead, fully wiped, corotating screw elements. Booy (1978) discusses the geometry of extruder screw elements. Janssen (1978) and Stevens (1985) present principles of twin screw extrusion in their books. Tadmor and Klein (1970) present a good description of the extruder screw geometry.

1.5 Objective and Organization of Research

The specific objective of this research is to develop two-dimensional and three-dimensional symbolic models for conjugate geometry and to test these models using case studies in mechanisms and in manufacturing. These would be the computational models. From these computational models graphical models can be developed. The advantage with this method is that a general algorithm would eliminate the necessity of developing separate programs for each case-study. Also, the effect of varying one parameter while keeping the others fixed can be studied with ease.

The work is split into three phases:

Symbolic Model Development

This involves the development of the algorithm for the symbolic model in MACSYMA for the two-dimensional and three-dimensional case.

Development of Computational Models

This involves the execution of the symbolic program for the specific case studies in two-dimensions (mechanisms) and three-dimensions (manufacturing). This phase is also carried out in MACSYMA.

Development of Graphical Models

The output of phase two (the computational model) is used to develop two-dimensional plots of the results to enable graphical visualization. This phase is carried out in one of the various graphics packages available.

The advantage of this research is that it presents a method of developing a general program (the symbolic model) that can be used to run different kinds of case studies.

The organization of research is shown in Figures 3 and 4. Chapter two deals with the actual development of the two-dimensional and three-dimensional symbolic algorithms and a description of MACSYMA. Chapter three deals with the case studies in mechanisms (2-D) and one 3-D manufacturing case study. A brief discussion on helically swept surfaces is presented in chapter four along with a description of three possible cases of extruder screw manufacture. Chapter five presents the results of examples on the three cases of screw manufacture discussed in chapter four. Chapter six presents conclusions with a technical summary and recommendations. Appendix A contains the transcript of a typical MACSYMA session.

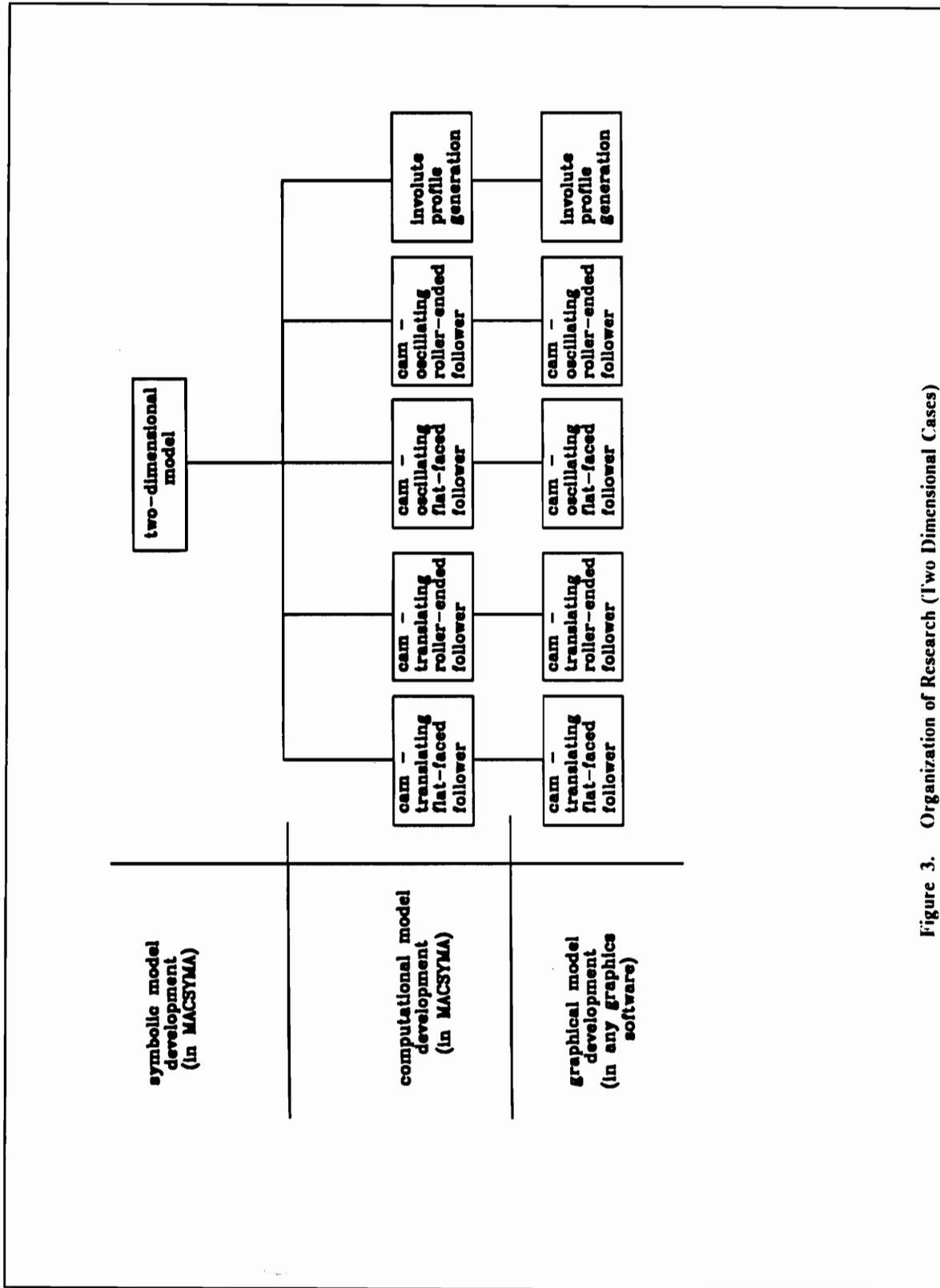


Figure 3. Organization of Research (Two Dimensional Cases)

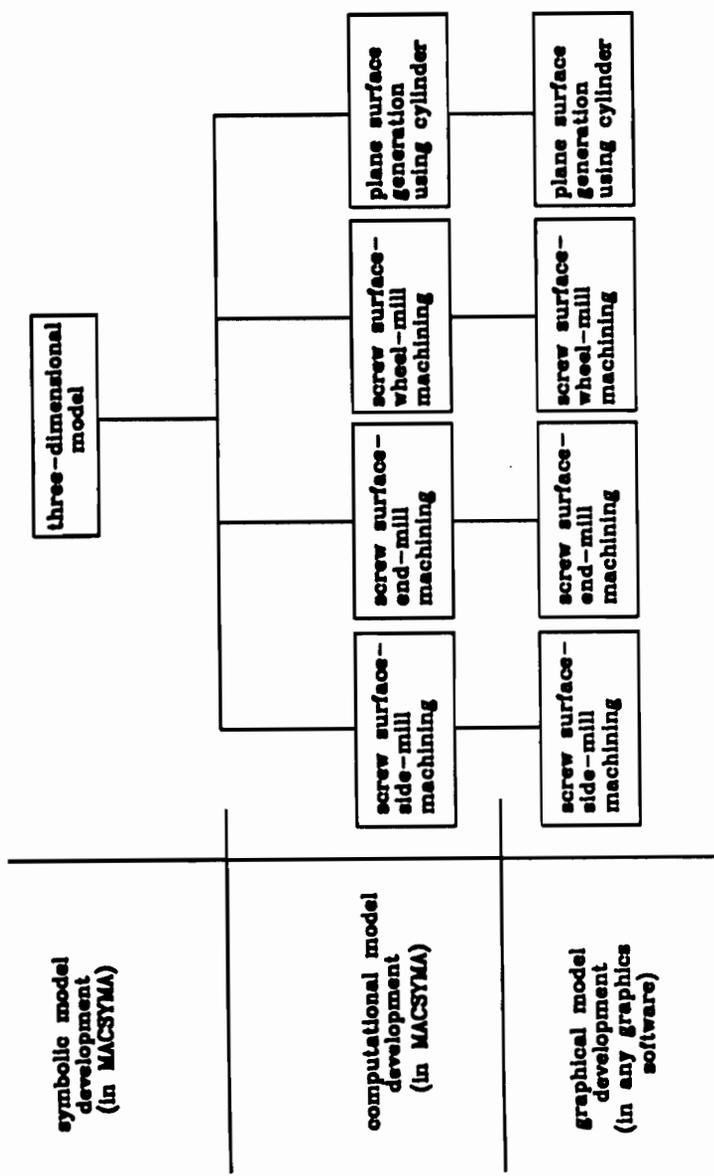


Figure 4. Organization of Research (Three Dimensional Cases)

Chapter 2

Symbolic Conjugate Geometry Model Development

2.1 Two-Dimensional Case Formulation

In this section, a symbolic model for a general two-dimensional conjugate geometry case will be developed.

Figure 5 represents the conjugate geometry model in two-dimensions. There are three coordinate systems defined in this model. The global coordinate system is identified as the 0 system, which is a fixed system. Coordinate systems 1 and 2 are moving coordinate systems.

The coordinate system 1 is attached to the tool. For the sake of convenience, throughout this manuscript the 'known' curve/surface will be referred to as the tool curve/surface and the curve/surface resulting from the action of the tool curve/surface will be known as the blank curve/surface. For example, in a cam-follower case where the cam profile is being determined from a known follower profile and motion, the known follower would be the tool and the resulting cam would be the blank. Coordinate system 2 is attached to the blank. In general, the profile of the

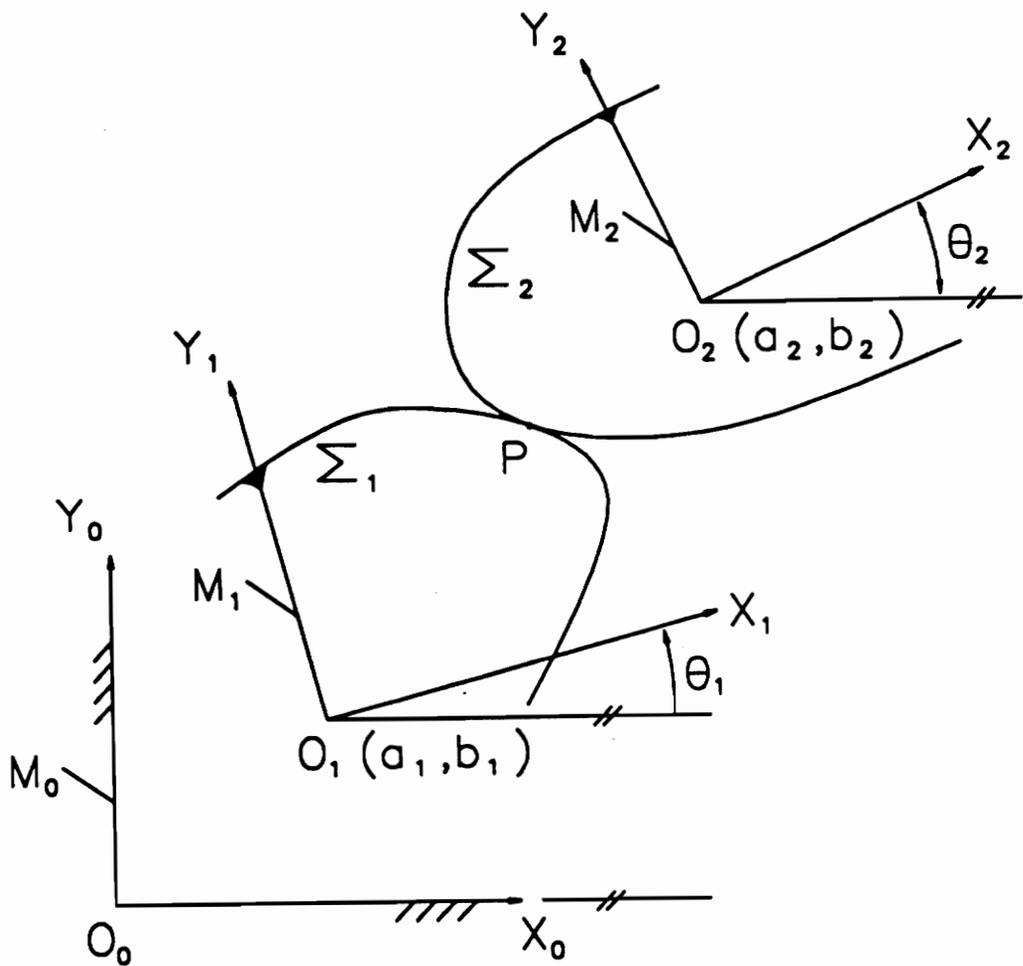


Figure 5. Two-Dimensional Conjugate Geometry Model

tool is known and the motions of the tool and blank are known. The tool and blank have three degrees-of-freedom each. (Translation in the X and Y directions and rotation about the Z axis). The tool and blank are in continuous contact at a point P. Following is a description of the nomenclature used.

M_0 : global coordinate system 0

M_1 : tool coordinate system 1

M_2 : blank coordinate system 2

Σ_1 : the tool curve/surface

Σ_2 : the blank curve/surface

X_0 : X-axis of global coordinate system M_0

Y_0 : Y-axis of global coordinate system M_0

O_0 : origin of global coordinate system M_0

X_1 : X-axis of tool coordinate system M_1

Y_1 : Y-axis of tool coordinate system M_1

O_1 : origin of tool coordinate system M_1

X_2 : X-axis of blank coordinate system M_2

Y_2 : Y-axis of blank coordinate system M_2

O_2 : origin of blank coordinate system M_2

(a_1, b_1) : coordinates of O_1 in the global coordinate system M_0

(a_2, b_2) : coordinates of O_2 in the global coordinate system M_0

θ_1 : angle of rotation of the tool coordinate system measured from the X axis in the positive sense

θ_2 : angle of rotation of the blank coordinate system measured from the X axis in the positive sense

(x_1, y_1) : coordinates of P in the tool coordinate system M_1

(x_2, y_2) : coordinates of P in the blank coordinate system M_2

u : parameter of which both x_1 and y_1 are functions

t : time - parameter of which $a_1, b_1, a_2, b_2, \theta_1$ and θ_2 are functions

${}^1\mathbf{P}$: point P with reference to the tool coordinate system M_1 in vector form

${}^2\mathbf{P}$: point P with reference to the blank coordinate system M_2 in vector form

$[{}^0_1\mathbf{T}]$: matrix for transformation from coordinate system M_1 to ψ οορδινατε συστημα M_0

$[{}^0_2\mathbf{T}]$: matrix for transformation from the coordinate system M_2 to coordinate system M_0

The point of contact P between Σ_1 and Σ_2 , when transformed to the global coordinate system M_0 from their respective M_1 and M_2 coordinate systems, will have the same coordinates, as the point is the same. This is represented by Equation [2.1].

$$[{}^0_1\mathbf{T}] \quad {}^1\mathbf{P} = [{}^0_2\mathbf{T}] \quad {}^2\mathbf{P} \quad [2.1]$$

$${}^1\mathbf{P} = \begin{bmatrix} x_1(u) \\ y_1(u) \\ 1 \end{bmatrix} \quad \text{and} \quad {}^2\mathbf{P} = \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} \quad [2.2]$$

where ${}^1\underline{P}$ is known and ${}^2\underline{P}$ is unknown. ${}^1\underline{P}$ and ${}^2\underline{P}$ are represented in homogeneous coordinates in Equation [2.2]. Therefore from Equation [2.1] we get,

$${}^2\underline{P} = [{}^0_2T]^{-1} [{}^0_1T] {}^1\underline{P} \quad [2.3]$$

$[{}^0_1T]$ and $[{}^0_2T]$ are given by Equations [2.4] and [2.5] respectively.

$$[{}^0_1T] = \begin{bmatrix} \cos \theta_1 & -\sin \theta_1 & a_1 \\ \sin \theta_1 & \cos \theta_1 & b_1 \\ 0 & 0 & 1 \end{bmatrix} \quad [2.4]$$

$$[{}^0_2T] = \begin{bmatrix} \cos \theta_2 & -\sin \theta_2 & a_2 \\ \sin \theta_2 & \cos \theta_2 & b_2 \\ 0 & 0 & 1 \end{bmatrix} \quad [2.5]$$

Input to the symbolic program would be x_1, y_1 in terms of u and $a_1, b_1, a_2, b_2, \theta_1, \theta_2$ in terms of t (time).

Output from the symbolic program would be x_2, y_2 in terms of u and t using Equation [2.3].

Now there exists an equation for the curve of the blank in terms of two parameters (u and t). To be able to uniquely define this curve another condition that relates u and t is required. This condition will be called the condition of contact.

2.1.1 Condition of Contact

For Σ_1 and Σ_2 to be in contact, the condition of contact must be satisfied.

The equation of the curve Σ_1 in the M_1 coordinate system is expressed as in Equation [2.2] where u is the parameter of the curve Σ_1 . The normal at any point on Σ_1 is given by

$${}^1\mathbf{N}_{P,1} = \begin{bmatrix} \frac{dy_1}{du} \\ -\frac{dx_1}{du} \end{bmatrix} \quad [2.6]$$

This normal vector transformed into the M_0 coordinate system is given by

$${}^0\mathbf{N}_{P,1} = [{}^0R] [{}^1\mathbf{N}_{P,1}] \quad [2.7]$$

$${}^0\mathbf{N}_{P,1} = \begin{bmatrix} \cos \theta_1 & -\sin \theta_1 \\ \sin \theta_1 & \cos \theta_1 \end{bmatrix} \begin{bmatrix} \frac{dy_1}{du} \\ -\frac{dx_1}{du} \end{bmatrix} \quad [2.8]$$

If ${}^0\mathbf{V}_{P,12}$ is the relative velocity vector between the curves Σ_1 and Σ_2 at the point of contact P then ${}^0\mathbf{V}_{P,12}$ must lie on the common tangent line at P to both Σ_1 and Σ_2 . Furthermore, if P lies on the common contact line between Σ_1 and Σ_2 , then at this point both the curves must have a common normal. It follows that at contact points, vectors ${}^0\mathbf{N}_{P,1}$ and ${}^0\mathbf{V}_{P,12}$ must be orthogonal. Therefore expressing this using the dot product we have

$${}^0\mathbf{N}_{P,1} \cdot {}^0\mathbf{V}_{P,12} = 0 \quad [2.9]$$

Equation [2.9] is the condition of contact. The relative velocity vector ${}^0\underline{V}_{P,12}$ can be found by taking the difference between the velocity vectors of Σ_1 and Σ_2 at P. Equation [2.10] gives this relation.

$${}^0\underline{V}_{P,12} = {}^0\underline{V}_{P,2} - {}^0\underline{V}_{P,1} \quad [2.10]$$

where ${}^0\underline{V}_{P,1}$ and ${}^0\underline{V}_{P,2}$ are given by

$${}^0\underline{V}_{P,1} = [{}^0_1\dot{R}] \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} + \begin{bmatrix} \dot{a}_1 \\ \dot{b}_1 \end{bmatrix} \quad [2.11]$$

$${}^0\underline{V}_{P,2} = [{}^0_2\dot{R}] \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} + \begin{bmatrix} \dot{a}_2 \\ \dot{b}_2 \end{bmatrix} \quad [2.12]$$

$$[{}^0_1R] = \begin{bmatrix} \cos \theta_1 & -\sin \theta_1 \\ \sin \theta_1 & \cos \theta_1 \end{bmatrix} \quad [2.13]$$

$$[{}^0_2R] = \begin{bmatrix} \cos \theta_2 & -\sin \theta_2 \\ \sin \theta_2 & \cos \theta_2 \end{bmatrix} \quad [2.14]$$

$$[{}^0_1\dot{R}] = \dot{\theta}_1 \begin{bmatrix} -\sin \theta_1 & -\cos \theta_1 \\ \cos \theta_1 & -\sin \theta_1 \end{bmatrix} \quad [2.15]$$

$$[{}^0_2\dot{R}] = \dot{\theta}_2 \begin{bmatrix} -\sin \theta_2 & -\cos \theta_2 \\ \cos \theta_2 & -\sin \theta_2 \end{bmatrix} \quad [2.16]$$

By simplifying Equations [2.11] and [2.12] the following relations result

$${}^0\underline{V}_{P,1} = \begin{bmatrix} -\sin \theta_1 \dot{\theta}_1 & -\cos \theta_1 \dot{\theta}_1 & \dot{a}_1 \\ \cos \theta_1 \dot{\theta}_1 & -\sin \theta_1 \dot{\theta}_1 & \dot{b}_1 \\ 0 & 0 & 0 \end{bmatrix} {}^1\underline{P} \quad [2.17]$$

$${}^0\mathcal{L}_{P,2} = \begin{bmatrix} -\sin \theta_2 \dot{\theta}_2 & -\cos \theta_2 \dot{\theta}_2 & \dot{a}_2 \\ \cos \theta_2 \dot{\theta}_2 & -\sin \theta_2 \dot{\theta}_2 & \dot{b}_2 \\ 0 & 0 & 0 \end{bmatrix} {}^2\mathcal{P} \quad [2.18]$$

Substituting Equation [2.3] in Equation [2.16] the following relation is obtained

$${}^0\mathcal{L}_{P,2} = \begin{bmatrix} -\sin \theta_2 \dot{\theta}_2 & -\cos \theta_2 \dot{\theta}_2 & \dot{a}_2 \\ \cos \theta_2 \dot{\theta}_2 & -\sin \theta_2 \dot{\theta}_2 & \dot{b}_2 \\ 0 & 0 & 0 \end{bmatrix} [{}^0_2\mathcal{T}]^{-1} [{}^0_1\mathcal{T}] {}^1\mathcal{P} \quad [2.19]$$

Equation [2.9] is reduced symbolically in terms of u and t . Using Equations [2.3] and [2.9], a unique curve (Σ_2) can be defined.

2.1.2 Symbolic Algorithm for Two-Dimensional Case

Using the derivations in Sections 2.1 and 2.1.1, a symbolic algorithm can be written. This algorithm is programmed in MACSYMA. Here \leftarrow is the symbol for 'define'. The algorithm is presented here

1. Define x_1, y_1 as functions of u (parameter) and $a_1, b_1, \theta_1, a_2, b_2, \theta_2$ as functions of t (time).
2. Define vector ${}^1\mathcal{P}$

$${}^1\mathcal{P} \leftarrow \begin{bmatrix} x_1(u) \\ y_1(u) \\ 1 \end{bmatrix}$$

3. Define matrix $[{}^1\mathcal{T}]$

$${}^0_1\mathbf{T} \leftarrow \begin{bmatrix} \cos \theta_1 & -\sin \theta_1 & a_1 \\ \sin \theta_1 & \cos \theta_1 & b_1 \\ 0 & 0 & 1 \end{bmatrix}$$

4. Define matrix ${}^0_2\mathbf{T}$

$${}^0_2\mathbf{T} \leftarrow \begin{bmatrix} \cos \theta_2 & -\sin \theta_2 & a_2 \\ \sin \theta_2 & \cos \theta_2 & b_2 \\ 0 & 0 & 1 \end{bmatrix}$$

5. Derive matrix ${}^2_0\mathbf{T}$

$${}^2_0\mathbf{T} \leftarrow [{}^0_2\mathbf{T}]^{-1}$$

6. Derive vector ${}^2\mathbf{P}$

$${}^2\mathbf{P} \leftarrow [{}^0_2\mathbf{T}]^{-1} [{}^0_1\mathbf{T}] {}^1\mathbf{P}$$

7. Define normal vector ${}^1\mathbf{N}_{P,1}$

$${}^1\mathbf{N}_{P,1} \leftarrow \begin{bmatrix} \frac{dy_1}{du} \\ -\frac{dx_1}{du} \\ 0 \end{bmatrix}$$

8. Define normal vector ${}^0\mathbf{N}_{P,1}$

$${}^0\mathbf{N}_{P,1} \leftarrow \begin{bmatrix} \cos \theta_1 & -\sin \theta_1 & 0 \\ \sin \theta_1 & \cos \theta_1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \frac{dy_1}{du} \\ -\frac{dx_1}{du} \\ 0 \end{bmatrix}$$

9. Define relative velocity vector ${}^0\mathbf{V}_{P,12}$

$${}^0\mathbf{V}_{P,12} \leftarrow {}^0\mathbf{V}_{P,2} - {}^0\mathbf{V}_{P,1}$$

where

$${}^0\mathbf{V}_{P,1} \leftarrow \begin{bmatrix} -\sin \theta_1 \dot{\theta}_1 & -\cos \theta_1 \dot{\theta}_1 & \dot{a}_1 \\ \cos \theta_1 \dot{\theta}_1 & -\sin \theta_1 \dot{\theta}_1 & \dot{b}_1 \\ 0 & 0 & 0 \end{bmatrix} {}^1\mathbf{P}$$

$${}^0\mathbf{V}_{P,2} \leftarrow \begin{bmatrix} -\sin \theta_2 \dot{\theta}_2 & -\cos \theta_2 \dot{\theta}_2 & \dot{a}_2 \\ \cos \theta_2 \dot{\theta}_2 & -\sin \theta_2 \dot{\theta}_2 & \dot{b}_2 \\ 0 & 0 & 0 \end{bmatrix} [{}^0_2\mathbf{T}]^{-1} [{}^0_1\mathbf{T}] {}^1\mathbf{P}$$

10. Reduce symbolically the condition of contact

$${}^0\mathbf{N}_{P,1} \cdot {}^0\mathbf{V}_{P,12} = 0 \quad [2.9]$$

11. Output from steps 6 and 10 will define the curve Σ_2 .

The symbolic model described in this section is programmed in MACSYMA. It can be used to test any two-dimensional case as it is capable of handling all the possible degrees-of-freedom in two-dimensions. The tool and blank could each have three degrees-of-freedom. In the next chapter a few case-studies in two-dimensions are tackled.

2.2 *Three-Dimensional Case Formulation*

This section deals with the development of a three-dimensional symbolic algorithm specific to a desired case-study. A general three-dimensional symbolic program would handle six degrees-of-freedom each for the blank and the tool, but this would involve large computations and also would take more computer time. Since there were case-studies to be conducted, and since no real engineering problems involve so many degrees-of-freedom, a symbolic program to handle a few specific examples was developed. The development of this algorithm is similar to the two-dimensional case.

The case-study that is being referred to is the manufacture of extruder screw elements. The screw elements can be manufactured in different ways, but the method of interest here is the machining by the milling method. Three possible milling methods are (1) the horizontal side-mill, (2) the spherical end-mill or (3) a grinding wheel type mill which will be referred to as the wheel-mill. The end result desired is the shape of a screw from a cylindrical blank by the action of the tool on the blank and the relative motion between the tool and blank. The relative motion can be achieved in different ways, but for the sake of simplicity the tool will be assumed to be stationary during the milling operation and the blank will have the helical motion. The axis of the cylindrical blank is along the Z_2 axis. Refer to Figure 6.

The helical motion of the blank involves rotation about the Z_2 axis and also translation along the Z_2 direction. The tool has three degrees-of-freedom, which are the location of the origin of the tool coordinate system M_1 in the global coordinate system M_0 . The following nomenclature is used in addition to the nomenclature used in developing the two-dimensional case.

Z_0 : Z-axis of global coordinate system M_0

Z_1 : Z-axis of tool coordinate system M_1

Z_2 : Z-axis of blank coordinate system M_2

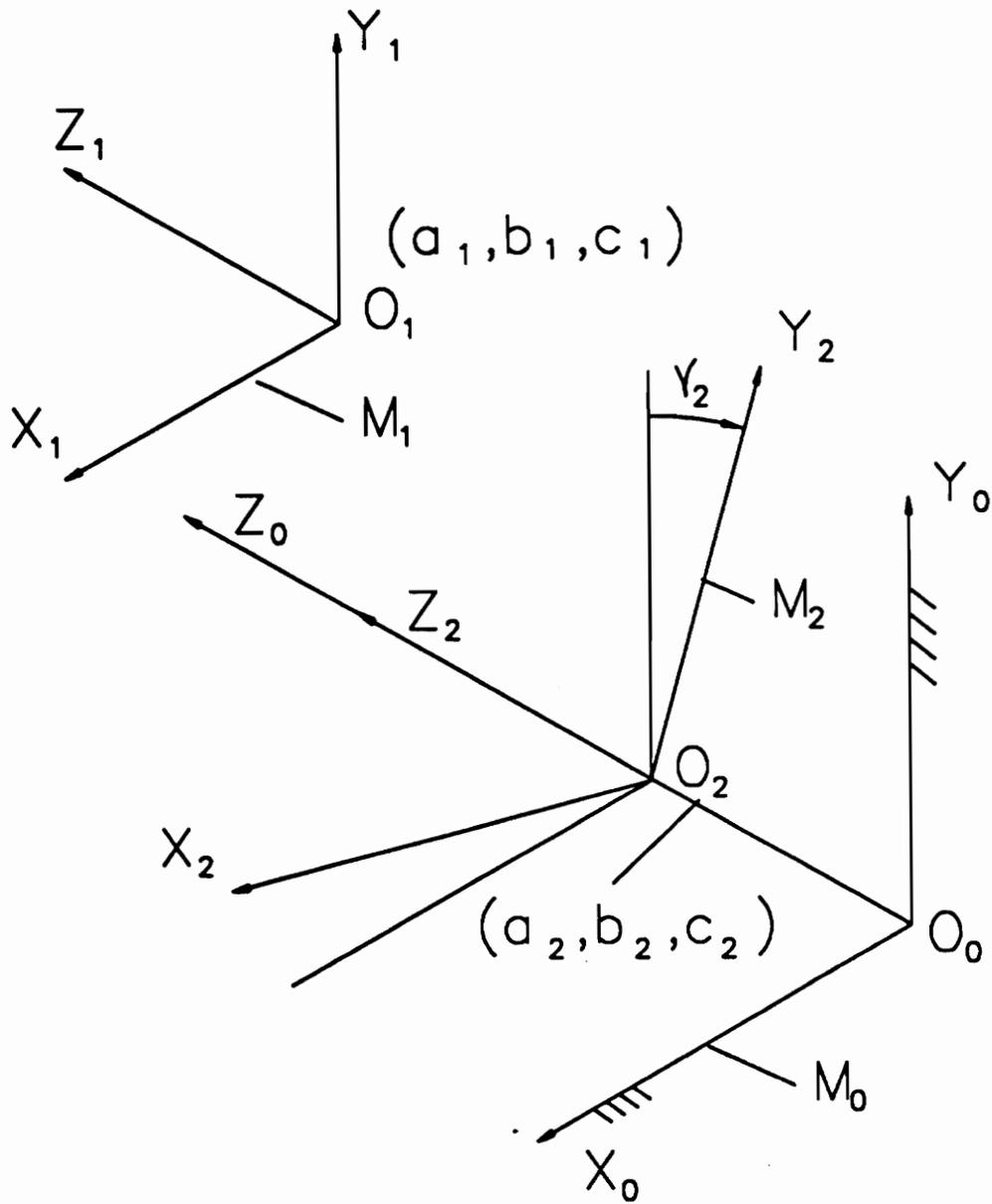


Figure 6. Three-Dimensional Conjugate Geometry Model - Coordinate Systems

c_1 : Z-coordinate of O_1 in the global coordinate system M_0

c_2 : Z-coordinate of O_2 in the global coordinate system M_0

γ_2 : angle of rotation of M_2 about the Z_2 axis measured from the X-axis in the positive sense

z_1 : Z-coordinate of P in the tool coordinate system M_1

z_2 : Z-coordinate of P in the blank coordinate system M_2

u : parameter of which x_1 , y_1 and z_1 are functions

v : parameter of which x_1 , y_1 and z_1 are functions

In this case $a_1, b_1, c_1, a_2, b_2, c_2$ and γ_2 are functions of time t and x_1, y_1 and z_1 are functions of two parameters u and v . Surfaces are being dealt with in this section instead of curves. To define a surface in parametric form completely, its coordinates must depend upon two parameters u and v .

Following will continue a formulation similar to the one presented for the two-dimensional case.

In this case, the point of contact P is represented by Equation [2.20]

$${}^1\mathbf{P} = \begin{bmatrix} x_1(u, v) \\ y_1(u, v) \\ z_1(u, v) \\ 1 \end{bmatrix} \quad \text{and} \quad {}^2\mathbf{P} = \begin{bmatrix} x_2 \\ y_2 \\ z_2 \\ 1 \end{bmatrix} \quad [2.20]$$

Equations [2.1] and [2.3] hold good for the three-dimensional formulation also. But the transformation matrices will be different, they are defined as

$${}^0_1\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & a_1 \\ 0 & 1 & 0 & b_1 \\ 0 & 0 & 1 & c_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad [2.21]$$

$${}^0_2\mathbf{T} = \begin{bmatrix} \cos \gamma_2 & -\sin \gamma_2 & 0 & a_2 \\ \sin \gamma_2 & \cos \gamma_2 & 0 & b_2 \\ 0 & 0 & 1 & c_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad [2.22]$$

2.2.1 Condition of Contact

The condition of contact for two surfaces is the same as it is for two curves as described in section 2.1.1, i.e., at the contact points the normal vector is orthogonal to the relative velocity vector. The normal vector is defined by

$${}^1\mathbf{N}_{P,1} = {}^1\mathbf{P}_u \times {}^1\mathbf{P}_v \quad [2.23]$$

where

$${}^1\mathbf{P}_u = \begin{bmatrix} \frac{\partial x_1}{\partial u} \\ \frac{\partial y_1}{\partial u} \\ \frac{\partial z_1}{\partial u} \\ 0 \end{bmatrix} \quad [2.24]$$

$${}^1\mathbf{P}_v = \begin{bmatrix} \frac{\partial x_1}{\partial v} \\ \frac{\partial y_1}{\partial v} \\ \frac{\partial z_1}{\partial v} \\ 0 \end{bmatrix} \quad [2.25]$$

The unit normal vector is defined by

$${}^1\mathbf{n}_{P,1} = \frac{{}^1\mathbf{N}_{P,1}}{\|{}^1\mathbf{N}_{P,1}\|} \quad [2.26]$$

where the double bars indicate the magnitude of the vector quantity.

The result of transforming the unit normal vector in Equation [2.26] to M_0 coordinate system is

$${}^0\mathbf{n}_{P,1} = [{}^0M] {}^1\mathbf{n}_{P,1} \quad [2.27]$$

where

$$[{}^0M] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad [2.28]$$

This is because the tool is stationary with respect to M_0 .

The relative velocity vector is given by Equation [2.10]. Since the tool is stationary, ${}^0\mathbf{V}_{P,1}$ will be zero. This means that the relative velocity vector is the velocity of the blank in the global coordinate system M_0 , which is ${}^0\mathbf{V}_{P,2}$. The equation for the relative velocity vector is follows

$${}^0\mathbf{V}_{P,2} = [{}^0\dot{T}] {}^2\mathbf{P} \quad [2.29]$$

Substituting from Equation [2.3] into Equation [2.29] the following equation is obtained

$${}^0\mathcal{L}_{P,2} = [{}^0\dot{T}_2] [{}^2_0T] [{}^0_1T] {}^1P \quad [2.30]$$

where

$$[{}^0\dot{T}_2] = \begin{bmatrix} -\sin \gamma_2 \dot{\gamma}_2 & -\cos \gamma_2 \dot{\gamma}_2 & 0 & \dot{a}_2 \\ \cos \gamma_2 \dot{\gamma}_2 & -\sin \gamma_2 \dot{\gamma}_2 & 0 & \dot{b}_2 \\ 0 & 0 & 0 & \dot{c}_2 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad [2.31]$$

Now Equation [2.9] reduces to

$${}^0n_{P,1} \cdot {}^0\mathcal{L}_{P,2} = 0 \quad [2.32]$$

Equation [2.32] can be written as

$$f(u, v, t) = 0 \quad [2.33]$$

Equation [2.3] can be written as

$${}^2P = {}^2P(u, v, t) \quad [2.34]$$

From Equations [2.32] and [2.33] it can be seen that 2P is a biparametric surface. This is the desired Σ_2 surface.

2.2.2 Symbolic Algorithm for Three-Dimensional Case

The algorithm presented in this section is similar to the algorithm in Section 2.1.2, except that this algorithm handles three-dimensional problems instead of two-dimensional ones. This algorithm is written using the derivations in Sections 2.2 and 2.2.1. The algorithm follows

1. Define x_1, y_1, z_1 as functions of u, v (parameters) and $a_1, b_1, c_1, a_2, b_2, c_2, \gamma_2$ as functions of t (time).

2. Define vector ${}^1\mathbf{P}$

$${}^1\mathbf{P} \leftarrow \begin{bmatrix} x_1(u, v) \\ y_1(u, v) \\ z_1(u, v) \\ 1 \end{bmatrix}$$

3. Define matrix $[\gamma T]$

$$[{}^0_1T] \leftarrow \begin{bmatrix} 1 & 0 & 0 & a_1 \\ 0 & 1 & 0 & b_1 \\ 0 & 0 & 1 & c_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

4. Define matrix $[\delta T]$

$$[{}^0_2T] \leftarrow \begin{bmatrix} \cos \gamma_2 & -\sin \gamma_2 & 0 & a_2 \\ \sin \gamma_2 & \cos \gamma_2 & 0 & b_2 \\ 0 & 0 & 1 & c_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

5. Derive matrix $[\delta T]$

$$[{}^2_0T] \leftarrow [{}^0_2T]^{-1}$$

6. Derive vector ${}^2\mathbf{P}$

$${}^2\mathbf{P} \leftarrow [{}^2_0T]^{-1} [{}^0_1T] {}^1\mathbf{P}$$

7. Define normal vector ${}^1\mathbf{N}_{P,1}$

$${}^1\mathbf{N}_{P,1} = {}^1\mathbf{P}_u \times {}^1\mathbf{P}_v$$

where

$${}^1\mathbf{P}_u = \begin{bmatrix} \frac{\partial x_1}{\partial u} \\ \frac{\partial y_1}{\partial u} \\ \frac{\partial z_1}{\partial u} \\ 0 \end{bmatrix}$$

$${}^1\mathbf{P}_v = \begin{bmatrix} \frac{\partial x_1}{\partial v} \\ \frac{\partial y_1}{\partial v} \\ \frac{\partial z_1}{\partial v} \\ 0 \end{bmatrix}$$

8. Define unit normal vector ${}^1\mathbf{n}_{P,1}$

$${}^1\mathbf{n}_{P,1} = \frac{{}^1\mathbf{N}_{P,1}}{\|{}^1\mathbf{N}_{P,1}\|}$$

9. Define unit normal vector ${}^0\mathbf{n}_{P,1}$

$${}^0\mathbf{n}_{P,1} = [{}^0_1M] {}^1\mathbf{n}_{P,1}$$

where

$$[{}^0_1M] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

10. Define relative velocity vector ${}^0V_{P,12}$

$${}^0V_{P,12} \leftarrow {}^0V_{P,2} - {}^0V_{P,1}$$

where, in this case,

$${}^0V_{P,1} = 0$$

$${}^0V_{P,2} \leftarrow [{}^0_2\dot{T}] [{}^2_0T] [{}^0_1T] {}^1P$$

where

$$[{}^0_2\dot{T}] = \begin{bmatrix} -\sin \gamma_2 \dot{\gamma}_2 & -\cos \gamma_2 \dot{\gamma}_2 & 0 & \dot{a}_2 \\ \cos \gamma_2 \dot{\gamma}_2 & -\sin \gamma_2 \dot{\gamma}_2 & 0 & \dot{b}_2 \\ 0 & 0 & 0 & \dot{c}_2 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

11. Reduce symbolically the condition of contact

$${}^0n_{P,1} \cdot {}^0V_{P,2} = 0$$

12. The output from steps 6 and 11 will define the surface Σ_2 .

The symbolic algorithm described in this section is programmed in MACSYMA and test cases involving the manufacture of extruder screw elements have been run with this program. These cases are described in Chapter 4.

2.3 *MACSYMA*

MACSYMA (project MAC's SYmbolic MANipulation system) is an interactive expert system and programming environment designed to assist in solving a wide spectrum of mathematical problems. It is written in LISP.

MACSYMA offers symbolic and numeric manipulation and solution capabilities in algebra, calculus and numerical analysis; two-dimensional and three-dimensional graphics; interface with mathematical text processors; and a user programming environment.

MACSYMA was developed and maintained at the Massachusetts Institute of Technology from 1969 until 1982. Since 1982, MACSYMA has been enhanced and supported by Symbolics, Inc. The MACSYMA system is the result of a development effort involving over 150 man-years of software design and programming. Including libraries of programs contributed by users, MACSYMA consists of more than 200,000 lines of source code.

The program, MACSYMA, can manipulate algebraic expressions involving constants, variables and functions. It has the power to differentiate, integrate, take limits, factor polynomials, simplify and make substitutions within symbolic expressions, expand analytic functions (both single and multivariate) in Taylor series, compute Poisson series, and take Laplace transforms and/or their inverses. It can also plot curves, solve algebraic and ordinary differential equations, and manipulate lists (vector fields, etc.), subscripted variables, tensors and matrices. MACSYMA can also solve systems of simultaneous linear and non-linear algebraic and differential equations. It also allows the user to write specialized programs for manipulating expressions.

To illustrate the utility of MACSYMA as a labor saving tool in formula simplification, consider reducing a long mathematical expression to a smaller one that requires fewer arithmetic operations to evaluate. Work of this type usually precedes computer program implementation of mathematical

formulas. Doing this may require identifying mathematical subexpressions that are identically zero, simplifying trigonometric expressions using trigonometric identities, or factoring long algebraic expressions. Many are aware that this can be an arduous task from personal experience. However, MACSYMA, with some user assistance, can identify subexpressions that vanish as well as perform other tasks in the simplification process. Furthermore, MACSYMA is more than just a symbol manipulator; it has the capability of evaluating expressions. Thus, in addition to representing a function symbolically, MACSYMA can be used to evaluate it at a particular point. This ability to combine numerical computation and symbol manipulation into one package is particularly useful since symbolic manipulators are not available in common computer languages, such as Fortran and Pascal. For more details on the capabilities of MACSYMA and guidance in using MACSYMA, the reader is referred to the MACSYMA reference manual (Symbolics, Inc. 1988).

Chapter 3

Case Studies in Conjugate Geometry

(Computational and Graphical Models)

This chapter deals with specific examples of applications of conjugate geometry in design and manufacture. These examples were run using the symbolic programs for two-dimensions and three-dimensions which were developed in Chapter 2. The cases that were run include the four disc cam cases, an involute profile case and a milling case. The milling case is an example of manufacturing applications. The other cases are examples of design applications. The four cam cases will be discussed first.

3.1 Disc Cams

Disc cams are examples of two-dimensional cams. The following are the four cases analyzed.

1. Disc cam with translating flat-faced follower (TFFF)

2. Disc cam with translating roller-ended follower (TREF)
3. Disc cam with oscillating flat-faced follower (OFFF)
4. Disc cam with oscillating roller-ended follower (OREF)

3.1.1 Disc Cam with Translating Flat-Faced Follower (TFFF)

A disc cam with translating flat-faced follower is shown in Figure 7. The flat face is perpendicular to the direction of translatory motion of the follower. Offset of the follower is permitted. The cam rotates with uniform angular velocity ω_2 . The angular rotation θ_2 is the parameter of cam motion. The follower linear displacement s_1 is measured from a reference line which is at an initial distance s_0 from another reference line from which θ_2 is measured (X_0 axis). The displacement function is given as

$$s_1 = s_1(\theta_2) \quad [3.1]$$

Coordinate systems used are shown in Figure 7 (same coordinate systems used in the formulation of the symbolic algorithm in Chapter 2). Coordinate system M_1 is attached to the follower. Coordinate system M_2 is attached to the rotating cam. The global coordinate system M_0 is fixed in space.

The input to the symbolic program is x_1, y_1 in terms of a parameter u . x_1, y_1 defines the follower curve which is in contact with the cam profile. In this particular case it is a straight line. The X_1 axis is in line with the flat face of the follower, thus the equation of the follower flat face is

$$x_1 = u \quad [3.2]$$

$$y_1 = 0$$

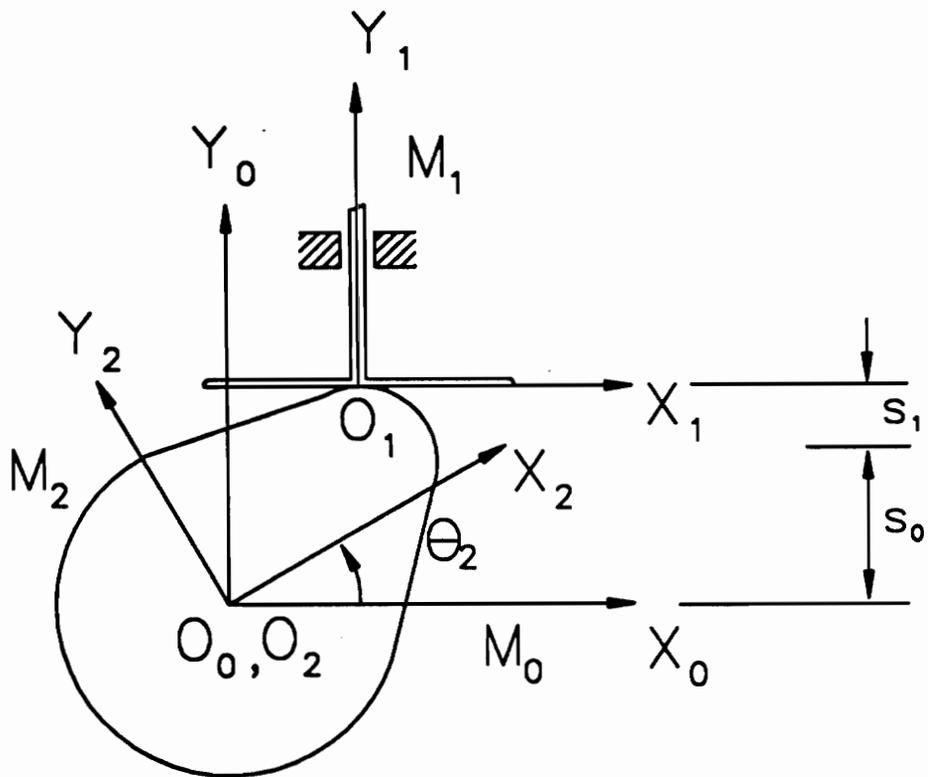


Figure 7. DISC CAM - Translating Flat-Faced Follower

The other input to the symbolic program is the location of the follower coordinate system in the fixed coordinate system (a_1, b_1) , the location of the cam coordinate system in the fixed coordinate system (a_2, b_2) , the angle of rotation of the follower coordinate system θ_1 and the angle of rotation of the cam θ_2 . The following input was selected

$$a_1 = 5 \quad [3.3]$$

which is the follower offset

$$b_1 = s_0 + s_1 \quad [3.4]$$

where

$$s_0 = 20$$

$$s_1 = 5 (1 - \cos \theta_2)$$

Thus s_1 would range from 0 to 10. This describes simple harmonic motion.

$$a_2 = 0 \quad [3.5]$$

$$b_2 = 0 \quad [3.6]$$

which means that the origin of the cam coordinate system is coincident with the origin of the fixed coordinate system.

$$\theta_1 = 0 \quad [3.7]$$

This is because the follower only translates, and does not rotate.

$$\theta_2 = \omega_2 t \quad [3.8]$$

where

$$\omega_2 = \frac{2\pi}{10}$$

Thus as t ranges from 0 to 10, θ_2 ranges from 0 to 2π .

The symbolic program, when given the above input, generates output in the form of two equations. One of the equations is the condition of contact (Equation [2.9]). The other equation is the equation describing the profile of the cam (r_2). The condition of contact relates the two parameters u and t . The equation for r_2 is in terms of u and t . Substituting for one of the parameters in terms of the other from the condition of contact relation into the equation for the profile of the cam, a unique cam profile can be defined. This is the desired cam curve.

The results of the program for the input defined earlier are shown here

$$u = 5 \left(\sin \left(\frac{\pi t}{10} \right) - 1 \right) \quad [3.9]$$

The relation in Equation [3.9] results after simplifying the MACSYMA output for the condition of contact.

The equation of the cam profile is

$$\begin{aligned} x_2 &= \cos \left(\frac{\pi t}{5} \right) (u + 5) + \left(-5 \cos \left(\frac{\pi t}{5} \right) + 25 \right) \sin \left(\frac{\pi t}{5} \right) \\ y_2 &= -\sin \left(\frac{\pi t}{5} \right) (u + 5) - \left(5 \cos \left(\frac{\pi t}{5} \right) - 25 \right) \cos \left(\frac{\pi t}{5} \right) \end{aligned} \quad [3.10]$$

The computational output of Equation [3.10] can be used to generate a graphical model to enable easy visualization of the cam profile. The generated cam profile is shown in Figure 8.

The next three disc cam cases are similar to the case described in Section 3.1.1.

3.1.2 Disc Cam with Translating Roller-Ended Follower (TREF)

This case is similar to the TFFF case, the only difference being the follower surface in contact with the cam. The follower in this case is a roller, so it is described as a circle. Refer to Figure 9.

The displacement function is also the same as the TFFF case (Equation [3.1]) The input to the symbolic program is described as

$$x_1 = r \cos(u) \quad [3.11]$$

$$y_1 = r \sin(u)$$

where $r = 2$

The rest of the input is

$$a_1 = 0 \quad [3.12]$$

$$b_1 = s_0 + s_1 \quad [3.13]$$

where

$$s_0 = 20$$

$$s_1 = 5 (1 - \cos \theta_2)$$

Thus s_1 would range from 0 to 10. This describes simple harmonic motion.

$$a_2 = 0 \quad [3.14]$$

$$b_2 = 0 \quad [3.15]$$

which means that the origin of the cam coordinate system is coincident with the origin of the fixed coordinate system.

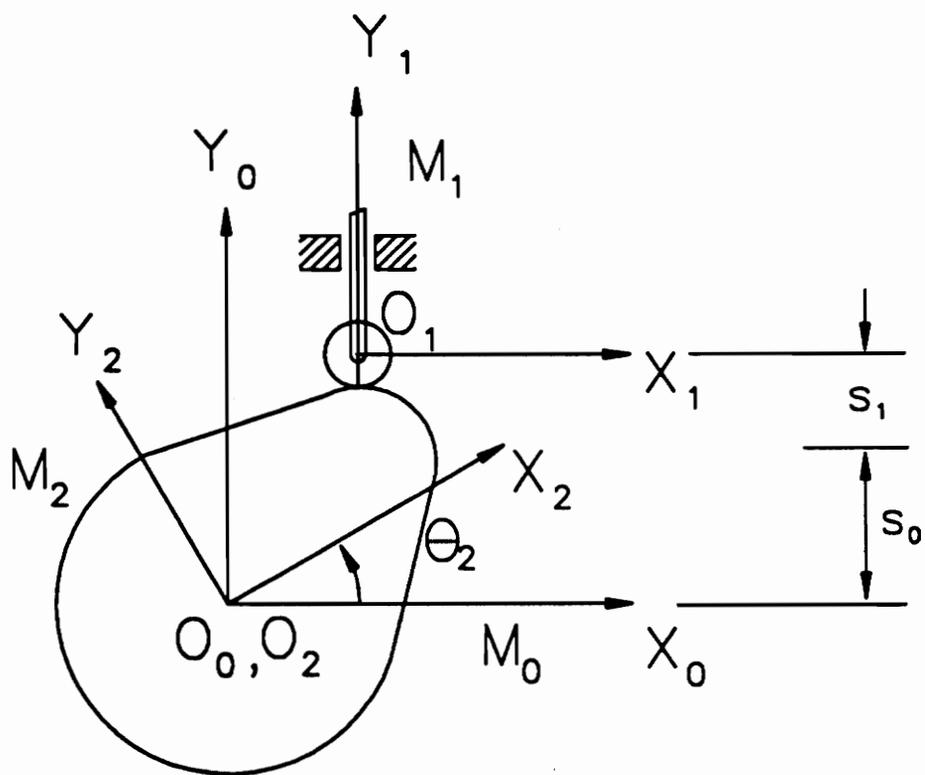


Figure 9. DISC CAM - Translating Roller-Ended Follower

$$\theta_1 = 0 \quad [3.16]$$

This is because the follower only translates, and does not rotate.

$$\theta_2 = \omega_2 t \quad [3.17]$$

where

$$\omega_2 = \frac{2\pi}{10}$$

Thus as t ranges from 0 to 10, θ_2 ranges from 0 to 2π . The output from the symbolic program for the above input is

$$\tan u = \frac{\cos\left(\frac{\pi t}{5}\right) - 5}{\sin\left(\frac{\pi t}{5}\right)} \left(\frac{\pi t}{5}\right) \quad [3.18]$$

Equation [3.18] is the condition of contact. It relates the parameters u and t

The equation of the cam profile is given by

$$x_2 = 2 \cos\left(\frac{\pi t}{5} - u\right) + \left(-5 \cos\left(\frac{\pi t}{5}\right) + 25\right) \sin\left(\frac{\pi t}{5}\right) \quad [3.19]$$

$$y_2 = 2 \sin\left(u - \frac{\pi t}{5}\right) + \left(-5 \cos\left(\frac{\pi t}{5}\right) + 25\right) \cos\left(\frac{\pi t}{5}\right)$$

The generated cam profile from Equation [3.19] is shown in Figure 10.

3.1.3 Disc Cam with Oscillating Flat-Faced Follower (OFFF)

A disc cam with an oscillating flat-faced follower is shown in Figure 11.

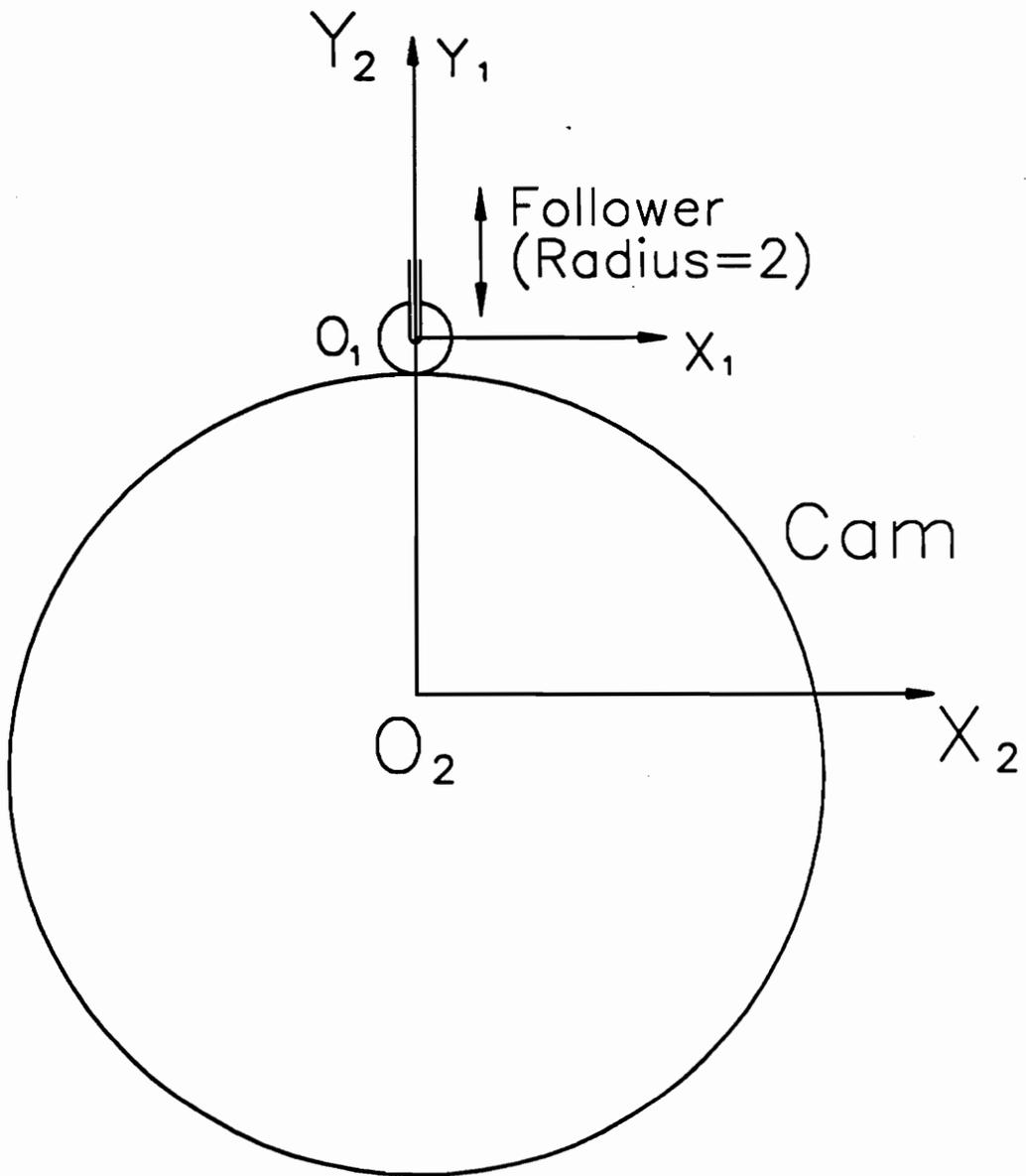


Figure 10. Generated Graphical Model of a Disc Cam Profile (TREF)

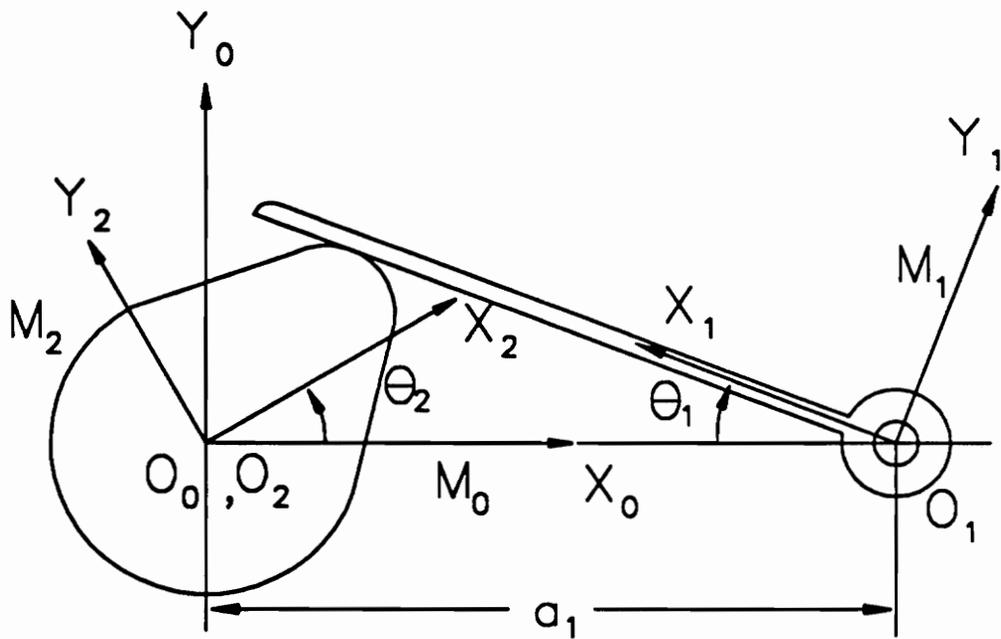


Figure 11. DISC CAM - Oscillating Flat-Faced Follower

By the arrangement of the coordinate systems it can be seen that the distance between the centers of rotation of the cam and the follower is a_1 . The cam rotates with a uniform angular velocity ω_2 as in the previous cases. The instantaneous angular velocity of the follower is ω_1 since in this case the follower is not translating but it is oscillating about the origin of the follower coordinate system. Cam angular rotation θ_2 , and follower angular rotation θ_1 are measured from the line of centers which is along the X_0 axis. The displacement equation is given by

$$\theta_1 = \theta_1(\theta_2) \quad [3.20]$$

The coordinate systems are shown in Figure 10. M_1 is attached to the follower and M_2 is attached to the cam as in previous cases.

The input to the symbolic program is given as

$$x_1 = u \quad [3.21]$$

$$y_1 = 0$$

This is the equation of the follower flat face (same as that in the TFFF case).

The rest of the input is

$$a_1 = -50 \quad [3.22]$$

which is the distance between the centers of rotation of the cam and the follower.

$$b_1 = 0 \quad [3.23]$$

$$a_2 = 0 \quad [3.24]$$

$$b_2 = 0 \quad [3.25]$$

From the above input it is clear that the cam and follower centers of rotation do not translate.

$$\theta_2 = \omega_2 t \quad [3.26]$$

where

$$\omega_2 = \frac{2\pi}{10}$$

Thus as t ranges from 0 to 10, θ_2 ranges from 0 to 2π .

$$\theta_1 = \left(\frac{\pi}{6}\right) \left(\sin\left(\frac{\theta_2}{2}\right) + 1\right) \quad [3.27]$$

Thus θ_1 would range from $\frac{\pi}{6}$ to $\frac{\pi}{3}$ as θ_2 ranges from 0 to 2π .

The equation for the condition of contact as a result of the above input is

$$u = \frac{600 \cos\left(\frac{\pi \sin\left(\frac{\pi t}{10}\right) + \pi}{6}\right)}{12 - \pi \cos\left(\frac{\pi t}{10}\right)} \quad [3.28]$$

The equation of the cam profile is

$$x_2 = u \cos\left(\frac{\pi t}{5} - \frac{\pi \sin\left(\frac{\pi t}{10}\right) + \pi}{6}\right) - 50 \cos\left(\frac{\pi t}{5}\right) \quad [3.29]$$

$$y_2 = u \sin\left(-\frac{\pi t}{5} + \frac{\pi \sin\left(\frac{\pi t}{10}\right) + \pi}{6}\right) + 50 \sin\left(\frac{\pi t}{5}\right)$$

using Equations [3.28] and [3.29] a graphical model of the cam profile can be generated as shown in Figure 12.

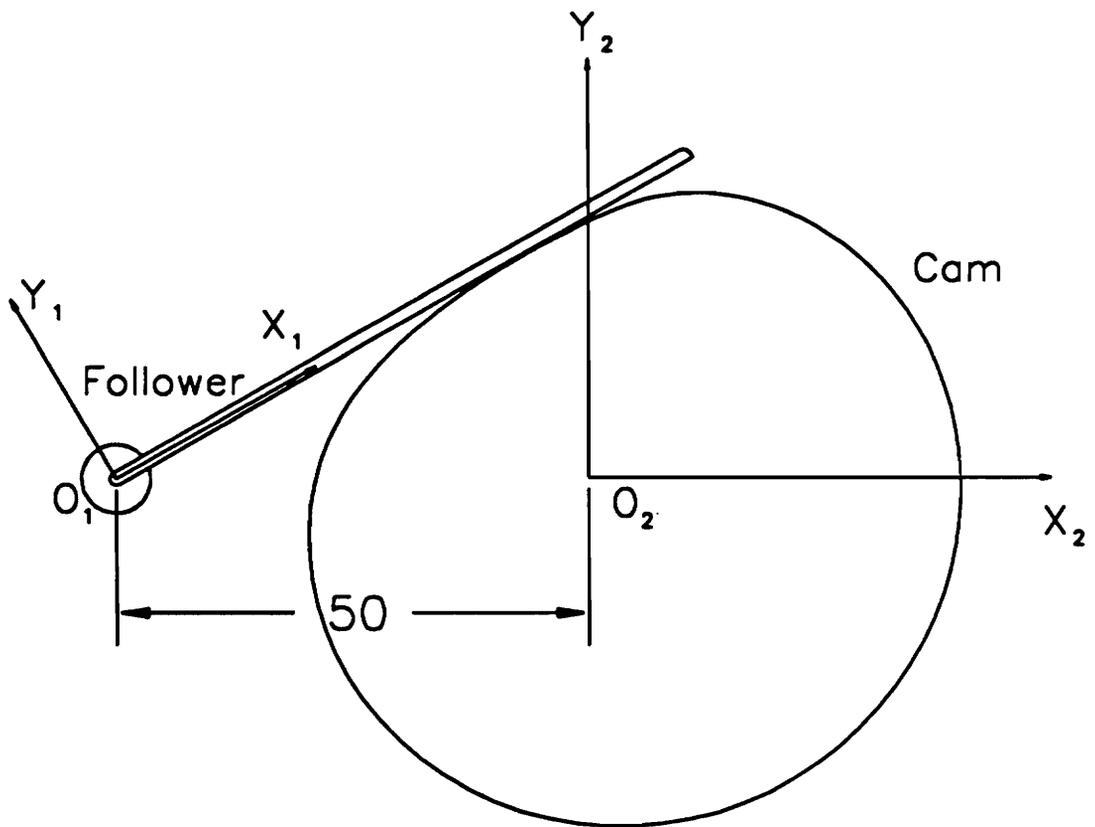


Figure 12. Generated Graphical Model of a Disc Cam Profile (OFFF)

3.1.4 Disc Cam with Oscillating Roller-Ended Follower (OREF)

Figure 13 shows a disc cam with an oscillating roller-ended follower.

This case is a combination of the TREF and OFFF cases. As the physical significance of a roller-ended follower and an oscillating follower have been described separately in the TREF and OFFF cases before, the input and output relations will be presented directly

The follower roller end is defined as

$$x_1 = 30 + 2 \cos(u) \quad [3.30]$$

$$y_1 = 2 \sin(u)$$

which means that the roller is located 30 units away from the follower center of rotation, or in other words, the follower link is 30 units long. u varies from 0 to 2π .

The rest of the input is similar to the OFFF case

$$a_1 = -50 \quad [3.31]$$

which is the distance between the centers of rotation of the cam and the follower.

$$b_1 = 0 \quad [3.32]$$

$$a_2 = 0 \quad [3.33]$$

$$b_2 = 0 \quad [3.34]$$

From the above input it is clear that the cam and follower centers of rotation do not translate.

$$\theta_2 = \omega_2 t \quad [3.35]$$

where

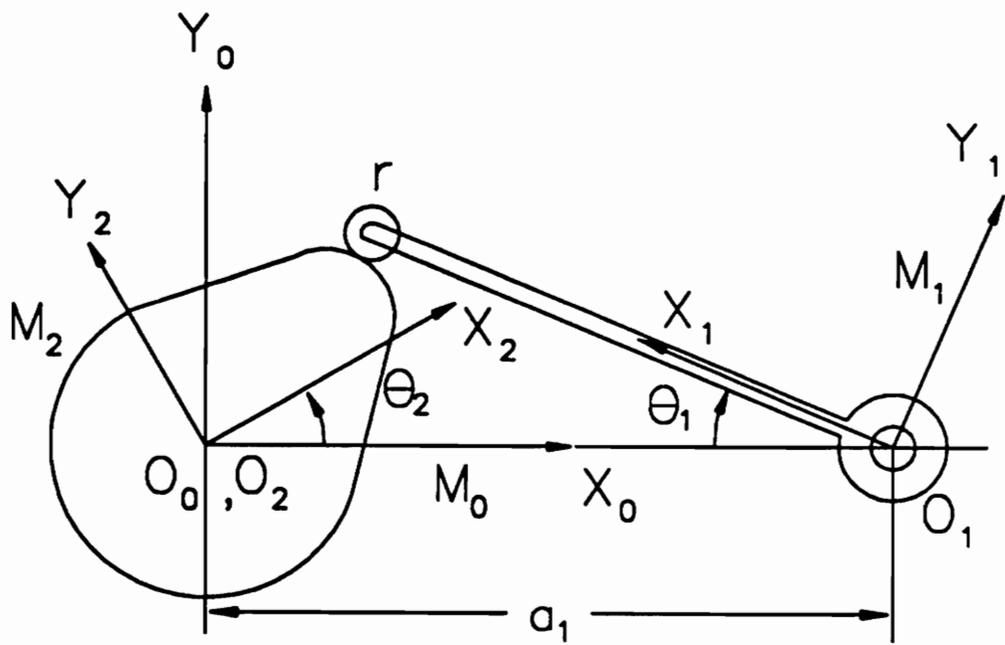


Figure 13. DISC CAM - Oscillating Roller-Ended Follower

$$\omega_2 = \frac{2\pi}{10}$$

Thus as t ranges from 0 to 10, θ_2 ranges from 0 to 2π .

$$\theta_1 = \left(\frac{\pi}{6}\right) \left(\sin\left(\frac{\theta_2}{2}\right) + 1\right) \quad [3.36]$$

Thus θ_1 would range from $\frac{\pi}{6}$ to $\frac{\pi}{3}$ as θ_2 ranges from 0 to 2π .

The equation for the condition of contact as a result of the above input is

$$\tan(u) = \frac{-20 \sin\left(\frac{\pi \sin\left(\frac{\pi t}{10}\right) + \pi}{6}\right)}{-12 + \pi \cos\left(\frac{\pi t}{10}\right) + 20 \cos\left(\frac{\pi \sin\left(\frac{\pi t}{10}\right) + \pi}{6}\right)} \quad [3.37]$$

The equation of the cam profile is

$$x_2 = 2 \cos\left(\frac{\pi t}{5} - \frac{\pi \sin\left(\frac{\pi t}{10}\right) + \pi}{6} - u\right) + 30 \cos\left(\frac{\pi t}{5} - \frac{\pi \sin\left(\frac{\pi t}{10}\right) + \pi}{6}\right) - 50 \cos\left(\frac{\pi t}{5}\right) \quad [3.38]$$

$$y_2 = 2 \sin\left(-\frac{\pi t}{5} + \frac{\pi \sin\left(\frac{\pi t}{10}\right) + \pi}{6} + u\right) + 30 \sin\left(-\frac{\pi t}{5} + \frac{\pi \sin\left(\frac{\pi t}{10}\right) + \pi}{6}\right) + 50 \sin\left(\frac{\pi t}{5}\right)$$

using Equations [3.37] and [3.38] a graphical model of the cam profile can be generated as shown in Figure 14.

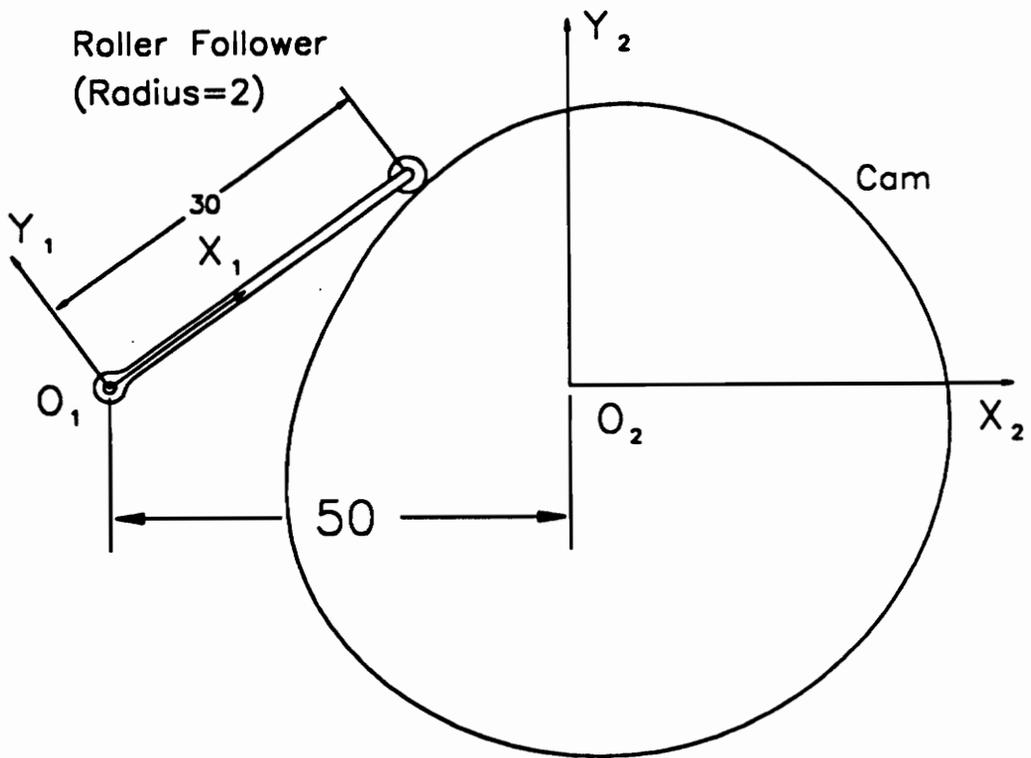


Figure 14. Generated Graphical Model of a Disc Cam Profile (OREF)

3.2 Gear Tooth Profile

The condition that is required when two gear teeth mesh together is that the angular-velocity ratio remain constant (Mabie and Reinholtz, 1987). This condition is satisfied when the line of action (the common normal to the two gear teeth surfaces) always intersects the line of centers at a fixed point. Of the many shapes possible that satisfy the above condition only the cycloid and the involute have been standardized. In this example, the involute profile of an example gear tooth will be generated. The involute profile in this example is generated by the line of action being given a uniform velocity in the X direction. Refer to Figure 15 for a setup of the coordinate system.

The angle the line of action makes with the Y-axis is the pressure angle α . The coordinate system M_1 is attached to the line of action and the coordinate system M_2 is attached to the tooth profile. The coordinate system M_2 rotates with a uniform angular velocity ω as the line of action moves in the X direction. The input to the symbolic program is given as

$$x_1 = -u \sin \alpha \quad [3.39]$$

$$y_1 = u \cos \alpha$$

where $\alpha = 20$ degrees, the pressure angle.

Equation [3.39] is the equation of the line of action. The rest of the input is as follows

$$a_1 = a_0 - 2vt \quad [3.40]$$

where $a_0 = 10$ and $v = b_1 \omega$

$$b_1 = 20 \quad [3.41]$$

So the line of action moves along a line parallel to and at a distance of 20 from the X_0 axis.

$$a_2 = 0 \quad [3.42]$$

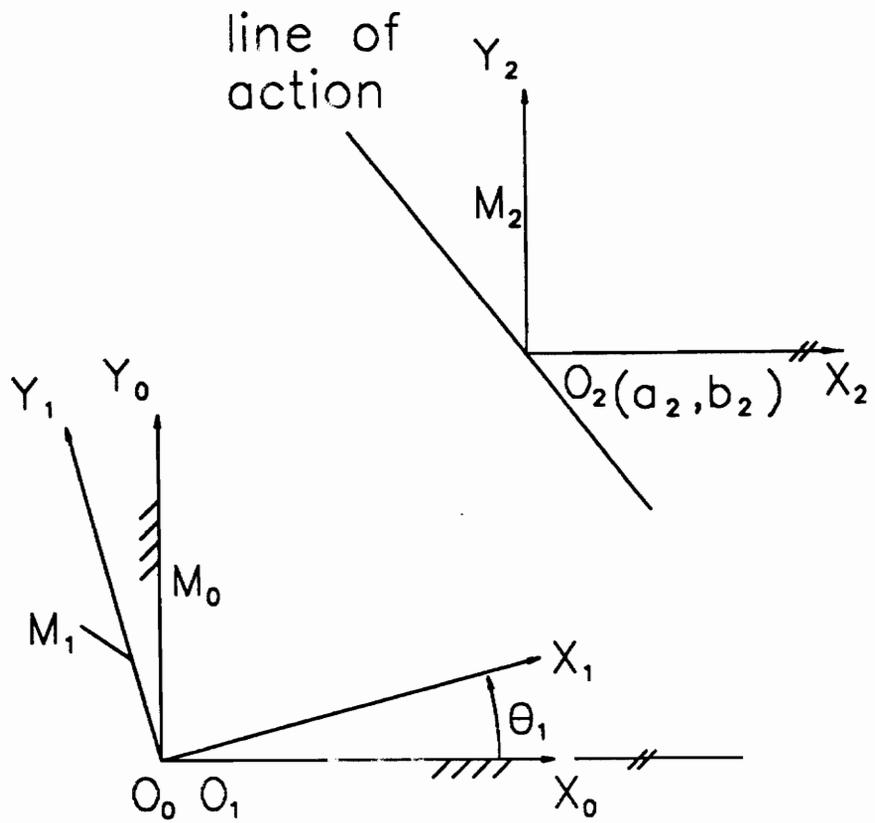


Figure 15. Involute Profile - Coordinate Systems

$$b_2 = 0 \quad [3.43]$$

The origin of M_2 is coincident with the origin of M_0 .

$$\theta_1 = 0 \quad [3.44]$$

$$\theta_2 = \omega t \quad [3.45]$$

where $\omega = 10$.

The condition of contact that results from executing the symbolic program with the above input is

$$u = \frac{2.63680916 \text{ E } 17 - 1.07852371 \text{ E } 17 - (9.59523468 \text{ E } 17) t}{7.01491619 \text{ E } 15} \quad [3.46]$$

The coefficients have such large values because MACSYMA substitutes for floating-point numbers with fractions.

The involute profile generated as a result of the above input is represented by the following equation

$$x_2 = \frac{((2.49 \text{ E } 7) \sin(10t) - (9.06 \text{ E } 6) \cos(10t)) u + (2.65 \text{ E } 8 - (1.06 \text{ E } 10) t) \cos(10t) + (5.3 \text{ E } 8) \sin(10t)}{2.65 \text{ E } 7} \quad [3.47]$$

$$y_2 = \frac{((2.49 \text{ E } 7) \cos(10t) + (9.06 \text{ E } 6) \sin(10t)) u + (-2.65 \text{ E } 8 + (1.06 \text{ E } 10) t) \sin(10t) + (5.3 \text{ E } 8) \cos(10t)}{2.65 \text{ E } 7}$$

Equations [3.46] and [3.47] are used to generate a graphical model of the involute profile. This model is shown in Figure 16.

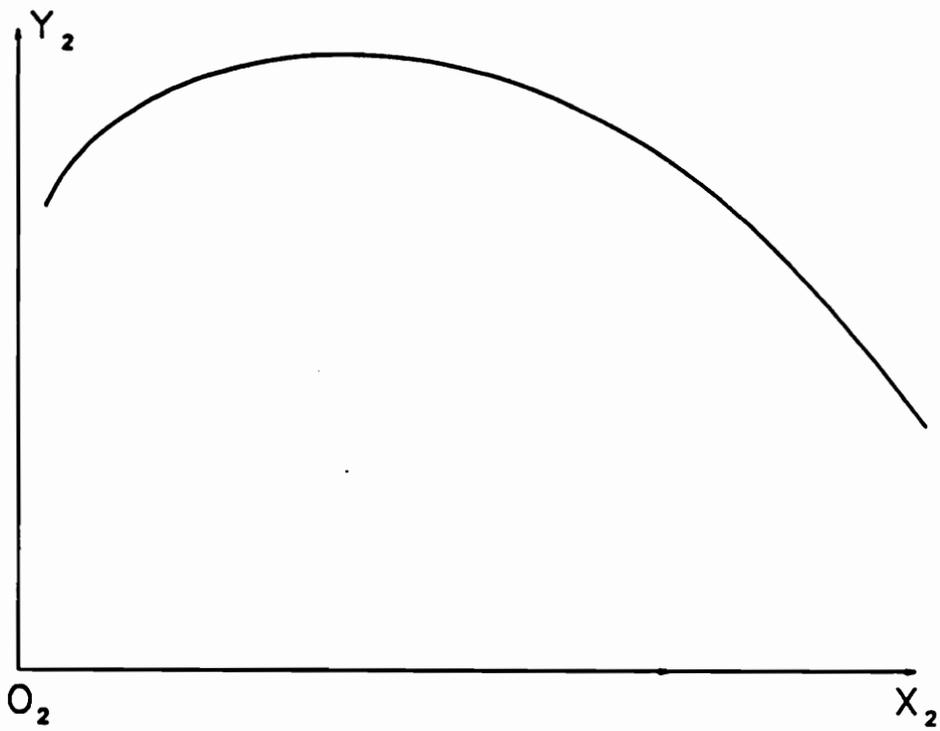


Figure 16. Generated Graphical Model of Involute Profile

3.3 Plane Generation Using Cylinder

This particular example is a three-dimensional case. It is a typical manufacturing situation. The example deals with the generation of a plane surface using a horizontal sidemill (cylinder). The blank is attached to coordinate system M_2 . The mill is attached to coordinate system M_1 . The arrangement of the coordinate systems is shown in Figure 17 along with the sidemill position.

As the sidemill is given motion along the Z direction it will produce a plane surface both at the top and bottom. This example was run using the three-dimensional symbolic algorithm. The input to the program is as follows

$$x_1 = v \quad [3.48]$$

$$y_1 = r \cos(u)$$

$$z_1 = r \sin(u)$$

This is the equation of the surface of the sidemill, which is a cylinder. The axis of the mill is along the X_1 axis. The parameter v varies along the length of the cylinder. r is the radius of the cylinder and in this case is selected as 30 units.

The rest of the input is as follows

$$a_1 = 0 \quad [3.49]$$

$$b_1 = 0 \quad [3.50]$$

$$c_1 = 5t \quad [3.51]$$

The mill moves in the Z direction with a velocity of 5 units/sec.

$$a_2 = 0 \quad [3.52]$$

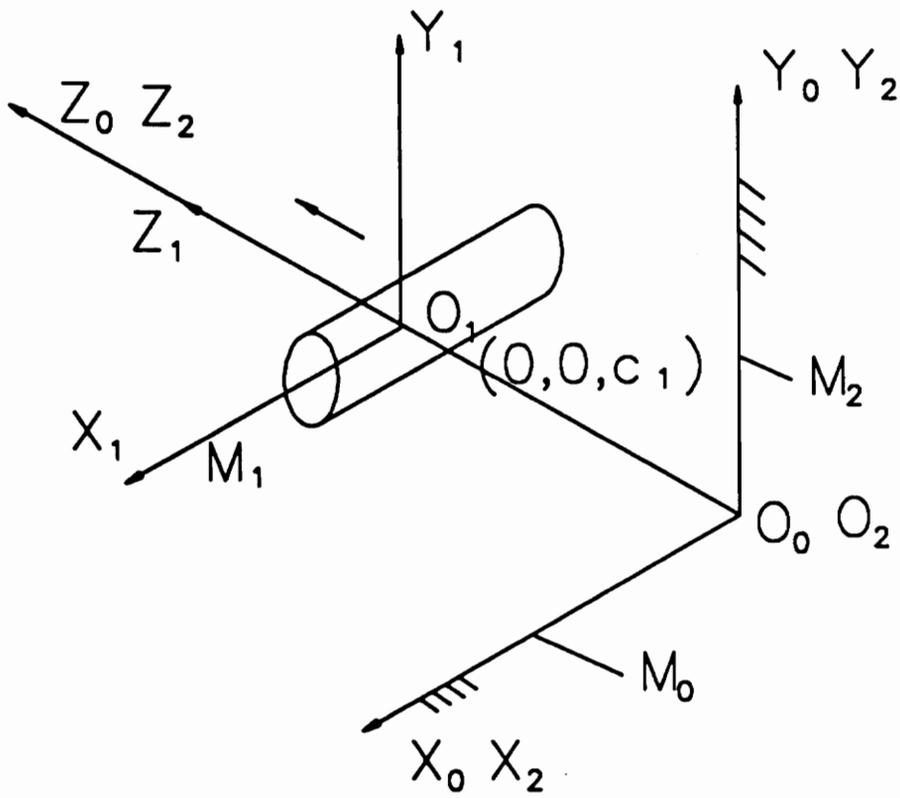


Figure 17. Plane Generation - Coordinate Systems

$$b_2 = 0 \quad [3.53]$$

$$c_2 = 0 \quad [3.54]$$

$$y_2 = 0 \quad [3.55]$$

The blank remains stationary with respect to the fixed coordinate system M_0 . There is no rotation of the blank either.

The following result for the condition of contact is obtained from the symbolic program.

$$\sin(u) = 0 \quad [3.56]$$

which means that $u = 0, \pi$.

The equation of the plane resulting from the sweep of the mill is given as

$$x_2 = v \quad [3.57]$$

$$y_2 = 30 \cos(u)$$

$$z_2 = 30 \sin(u) + 5t$$

By substituting the values of u obtained from the condition of contact into Equation [3.57], two plane surfaces result parallel to the XZ plane at a distance of 30 units on either side of the XZ plane. If the resulting plane surfaces are intersected with a plane perpendicular to the X_0 axis and passing through the YZ plane ($v = 0$), the resulting curves of intersection are straight lines and these are illustrated in Figure 18.

This chapter dealt with a few examples from the design areas (the cams and the involute profile generation) and one example from the manufacturing area (generation of a plane surface using a sidemill). Helically swept surface generation will be dealt with in Chapter 4.

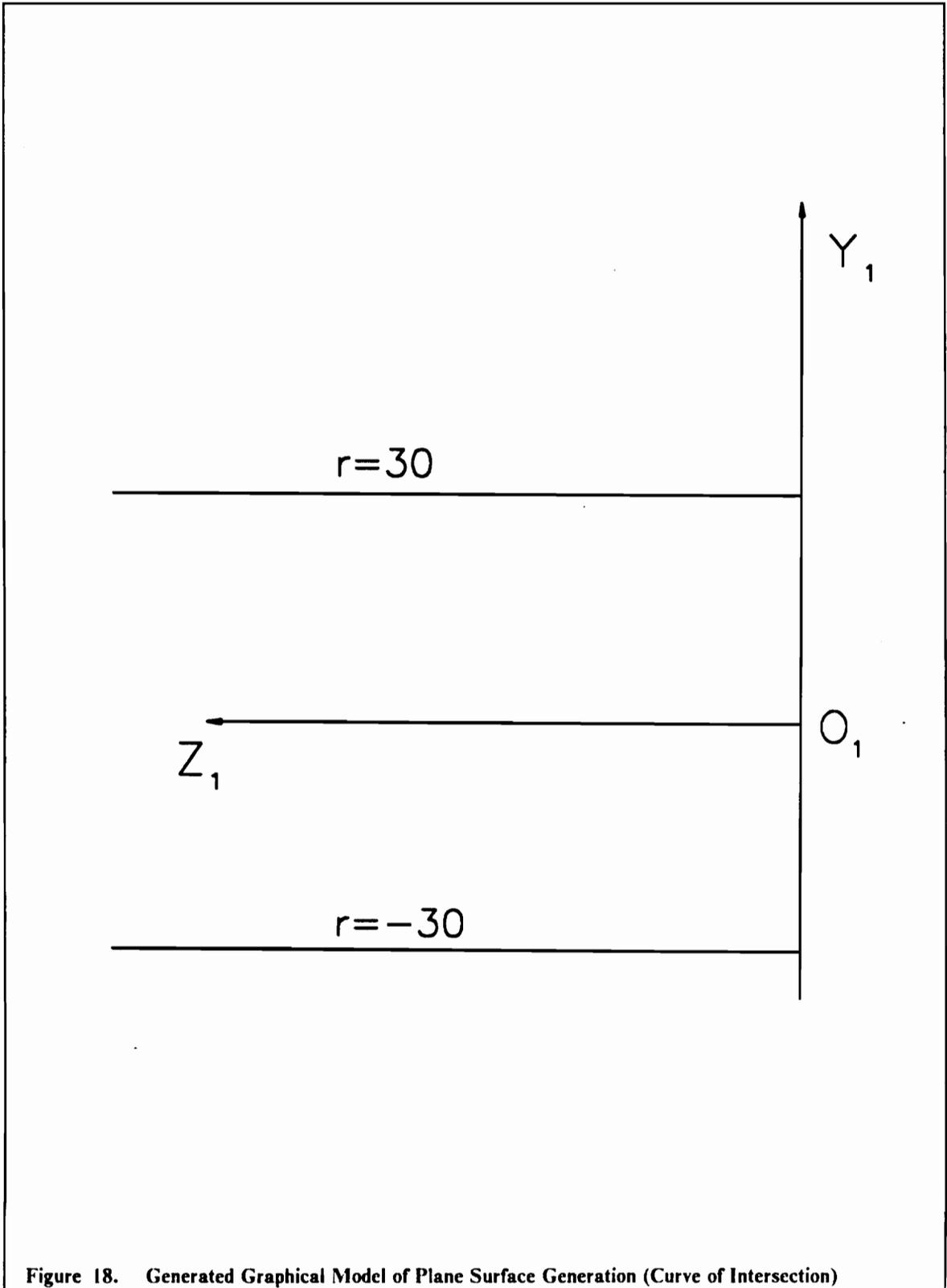


Figure 18. Generated Graphical Model of Plane Surface Generation (Curve of Intersection)

Chapter 4

Extruder Screw Conjugate Geometries

4.1 Helically Swept Surfaces

Flutes of a drill, a screw or a helical gear can be termed as helically swept surfaces. A helically swept surface is defined as the sweep of a generatrix curve when it is rotated around an axis as well as simultaneously translated along the axis of rotation. Consider a generatrix curve σ lying in the XOY plane (Figure 19).

Let u be the parameter ($u_{\min} \leq u \leq u_{\max}$) defining a generic point P_1 on this curve. The radius vector of P_1 can be defined as

$$\mathcal{L}_1(u) = \begin{bmatrix} x(u) \\ y(u) \\ 0 \\ 1 \end{bmatrix} \quad [4.1]$$

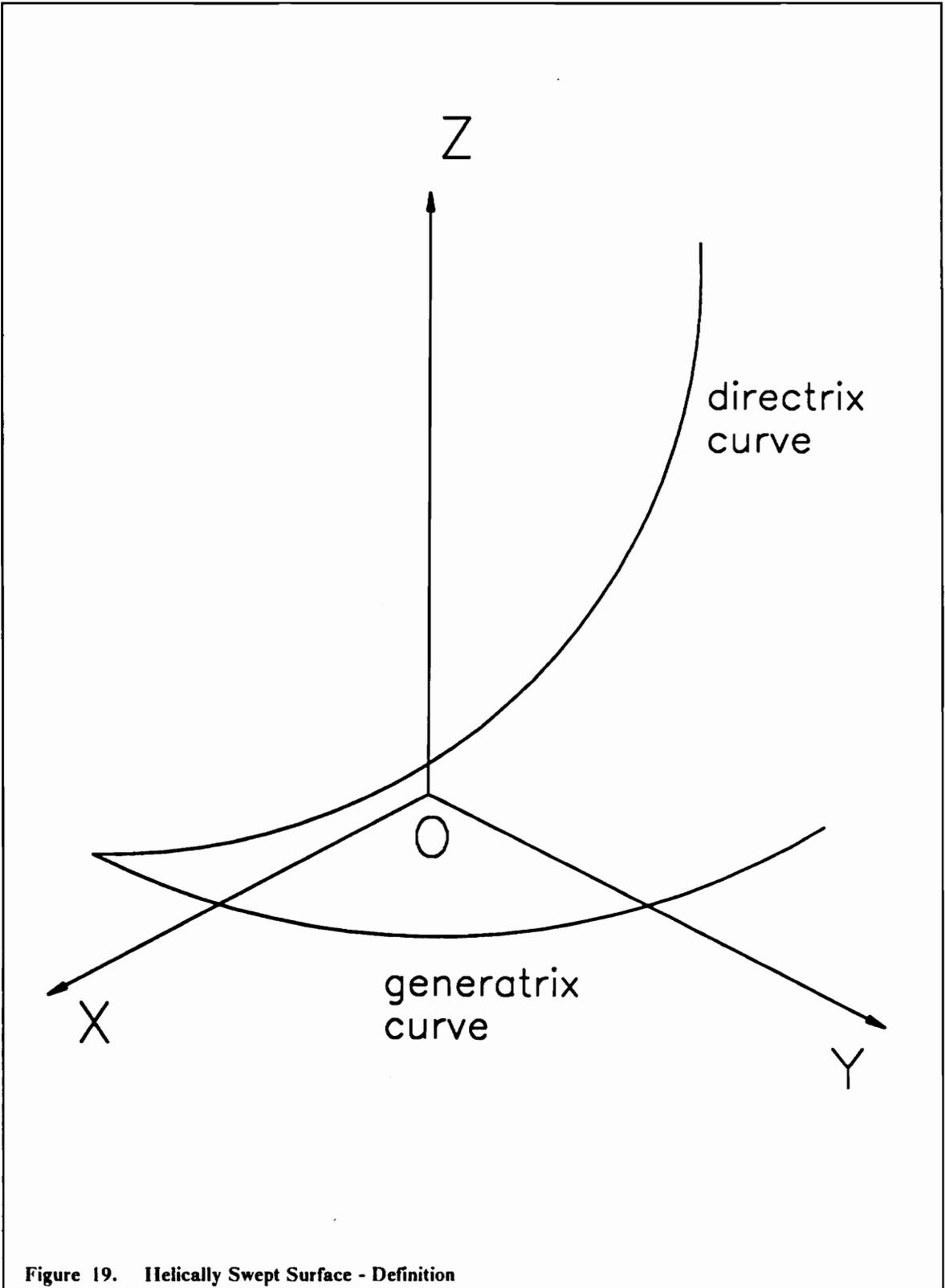


Figure 19. Helically Swept Surface - Definition

If this curve is rotated around the Z axis through an angle ν , ($0 \leq \nu \leq 2\pi$) as well as translated along the Z axis through a distance $(p\nu/2\pi)$ where p is the pitch of the helix then the location of the generic point P of the bi-parametric helically swept surface Σ_1 is given as follows

$$\underline{r}(u, \nu) = \begin{bmatrix} x(u, \nu) \\ y(u, \nu) \\ z(u, \nu) \end{bmatrix} = \begin{bmatrix} \cos \nu & -\sin \nu & 0 & 0 \\ \sin \nu & \cos \nu & 0 & 0 \\ 0 & 0 & 1 & \frac{p\nu}{2\pi} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x(u) \\ y(u) \\ 0 \\ 1 \end{bmatrix} \quad [4.2a]$$

or

$$x(u, \nu) = \cos(\nu) x(u) - \sin(\nu) y(u)$$

$$y(u, \nu) = \sin(\nu) x(u) + \cos(\nu) y(u) \quad [4.2b]$$

$$z(u, \nu) = \frac{p\nu}{2\pi}$$

It is assumed that the directrix curve is a right-handed helix.

There are several different processes that one can use to manufacture a helically swept surface. However, if the accuracy of the generated surface is required to be high then one can employ any one of the following three schemes,

1. Side-mill machining
2. End-mill machining
3. Wheel-mill machining

However, all the three schemes of machining can be represented by the symbolic model shown in Figure 6 in Chapter 2. This is the symbolic three-dimensional algorithm that was developed in Chapter 2. Let σ_1 be a generatrix curve defining a helically swept surface Σ_1 . Let σ_2 be the

generatrix curve of an axisymmetric cutter surface Σ_2 . It is necessary that Σ_1 be a conjugate surface to Σ_2 when bodies or coordinate frames attached to Σ_1 and Σ_2 are under going a relative helical motion.

The three schemes of machining will be discussed separately in the coming sections. Examples of the three schemes will be run using the symbolic MACSYMA program and results obtained will also be presented in these sections.

4.2 *Side-Mill Machining*

A graphical representation of the side-mill machining operation of helically swept extruder screw surfaces is shown in Figure 20 with all the coordinate systems. As can be seen from Figure 20, the horizontal side-mill can be represented as a cylinder. A cylindrical blank is used for the extruder screw.

The coordinate system M_1 is attached to the side-mill and the coordinate system M_2 is attached to the blank. The axis of the side-mill is along the X_1 axis and the axis of the blank is along the Z_2 axis. Thus the axis of the side-mill is perpendicular to the axis of the cylindrical blank. The side-mill rotates about its own axis, which does not affect the relative motion between the side-mill and the blank. So the side-mill will be assumed to be stationary. The location of the side-mill with respect to the fixed coordinate system M_0 can be varied. The three axes of M_1 are parallel to the three axes of M_0 . The blank coordinate system M_2 is coincident with the fixed coordinate system M_0 to begin with, but the blank is given helical motion about the Z_2 axis. Thus there is a rotation γ_2 about Z_2 and a translation along Z_2 .

The input to the symbolic program would be the equation of the surface of the side-mill, the location of the side-mill, the location of the blank and the helical motion parameters of the blank. The output from the symbolic program would be the equation of the condition of contact (this

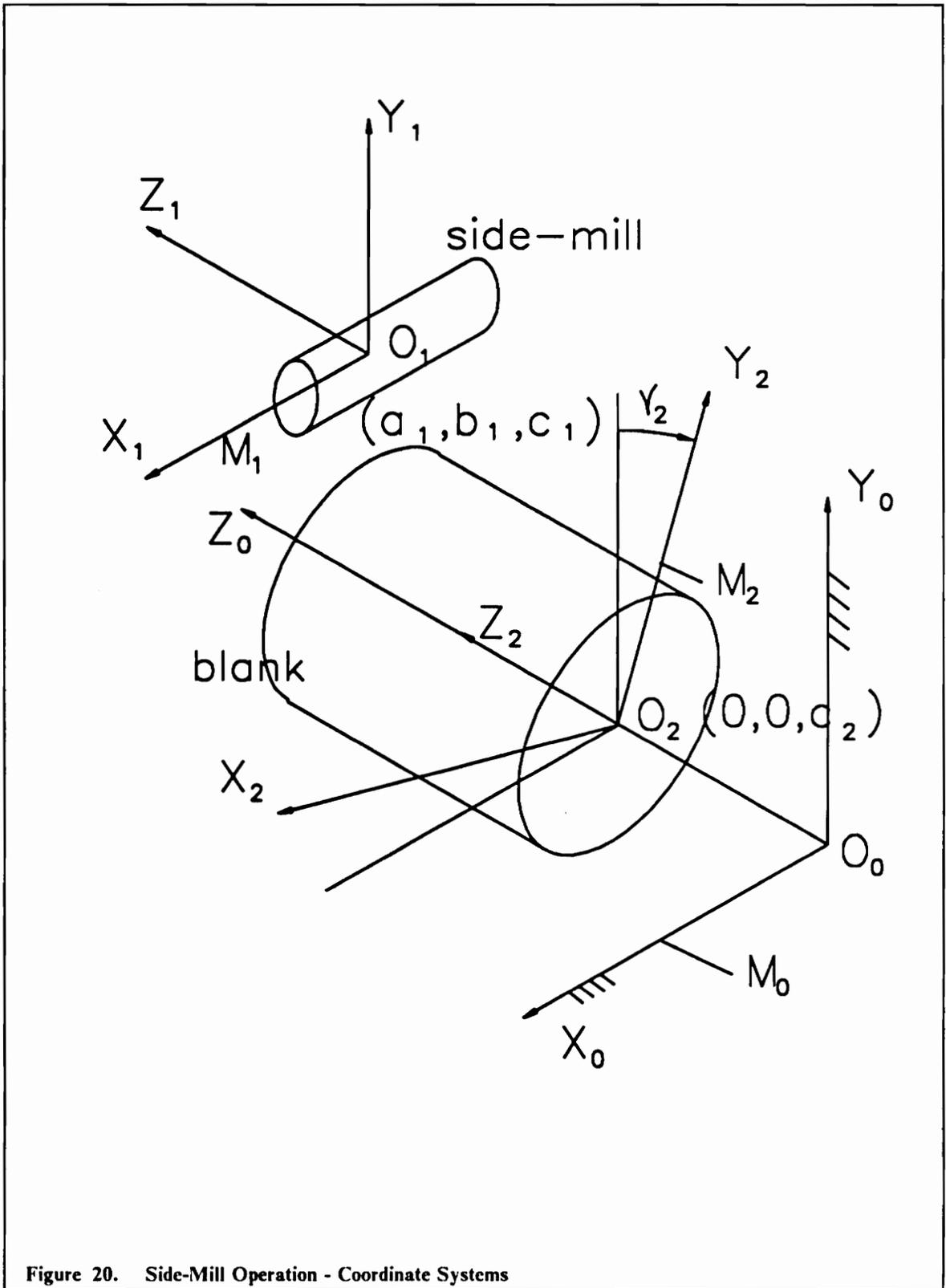


Figure 20. Side-Mill Operation - Coordinate Systems

relates the three parameters u , v and t) and the equation of the surface of the helically swept extruder surface in terms of the three parameters u , v and t .

The equation of the surface of the side-mill is

$$x_1 = v \quad [4.3]$$

$$y_1 = r \cos(u)$$

$$z_1 = r \sin(u)$$

where $0 \leq u \leq 2\pi$ and $v_{\min} \leq v \leq v_{\max}$ and r is the radius of the side-mill, which was selected as 12.5 for the example. The location of the side-mill is given by

$$a_1 = 0 \quad [4.4]$$

$$b_1 = 50$$

$$c_1 = 0$$

The location and motion of the blank is described by

$$a_2 = 0 \quad [4.5]$$

$$b_2 = 0$$

$$c_2 = \frac{l\gamma_2}{2\pi}$$

where the equation for c_2 describes the helical motion of the blank. l is the lead of the desired screw, and γ_2 is the angular rotation of the blank which is described by equation [4.6].

$$\gamma_2 = \omega_2 t \quad [4.6]$$

where ω_2 is the uniform angular velocity of rotation of the blank which is selected as 10 for this example. The lead of the screw was selected as 60 units for the purpose of the example.

The equation for the condition of contact as derived by the symbolic program for the above input is

$$v = \frac{-30 \tan(u)}{\pi} \quad [4.7]$$

Equation [4.7] is the simplified version of the output from MACSYMA.

The equation of the swept surface from MACSYMA is

$$\begin{aligned} x_2 &= \cos(10t)v + 12.5 \sin(10t) \cos(u) + 50 \sin(10t) \\ y_2 &= -\sin(10t)v + 12.5 \cos(10t) \cos(u) + 50 \cos(10t) \\ z_2 &= 12.5 \sin(u) - \frac{300t}{\pi} \end{aligned} \quad [4.8]$$

Substituting for v from Equation [4.7] into Equation [4.8], a unique bi-parametric equation for the helically swept surface is obtained.

In engineering practice the equation of the helically swept surface is of little use. The generatrix curve of this surface is of more use. This curve can be obtained by intersecting a plane perpendicular to the Z_2 axis with the helically swept surface Σ_2 . This curve is the σ_2 curve. One of the generatrix curves for the helically swept screw surface can be obtained by setting $z_2 = 0$ in Equation [4.8]. This results in a relation in terms of u and t given below

$$t = \frac{12.5 \pi \sin(u)}{300} \quad [4.9]$$

Substituting for t from Equation [4.9] into Equation [4.8], the equation of the generatrix curve σ_2 is obtained. This curve is illustrated in Figure 21 superimposed on the original blank shape (circle),

the shaded area in the figure is the removed material as a result of the side-mill machining operation. Figure 21 also shows the generatrix curve for the side-mill surface (σ_1), which is a circle of radius r ($= 12.5$ in this particular example).

4.3 End-Mill Machining

Figure 22 shows the end-mill machining operation of helically swept surfaces. The surface of the end-mill that is in contact with the blank is the hemispherical portion. Thus it can be represented by the equation of a sphere.

The coordinate system M_1 is attached to the end-mill and the coordinate system M_2 is attached to the cylindrical blank. The blank and its properties are similar to the case of the side-mill already described in the previous section. The axis of the end-mill is along the Y_1 axis. Thus the axis of the end-mill is perpendicular to the axis of the blank. The end-mill rotates about its own axis, which does not affect the relative motion between the end-mill and the blank. So the end-mill will be assumed to be stationary. The location of the end-mill with respect to the fixed coordinate system M_0 can be varied. The three axes of M_1 are parallel to the three axes of M_0 . The blank is given a similar helical motion as in the side-mill case.

The input to the symbolic program is described here. The equation of the surface of the end-mill is

$$\begin{aligned}x_1 &= r \cos(v) \cos(u) \\y_1 &= -r \sin(v) \\z_1 &= r \cos(v) \sin(u)\end{aligned}\tag{4.10}$$

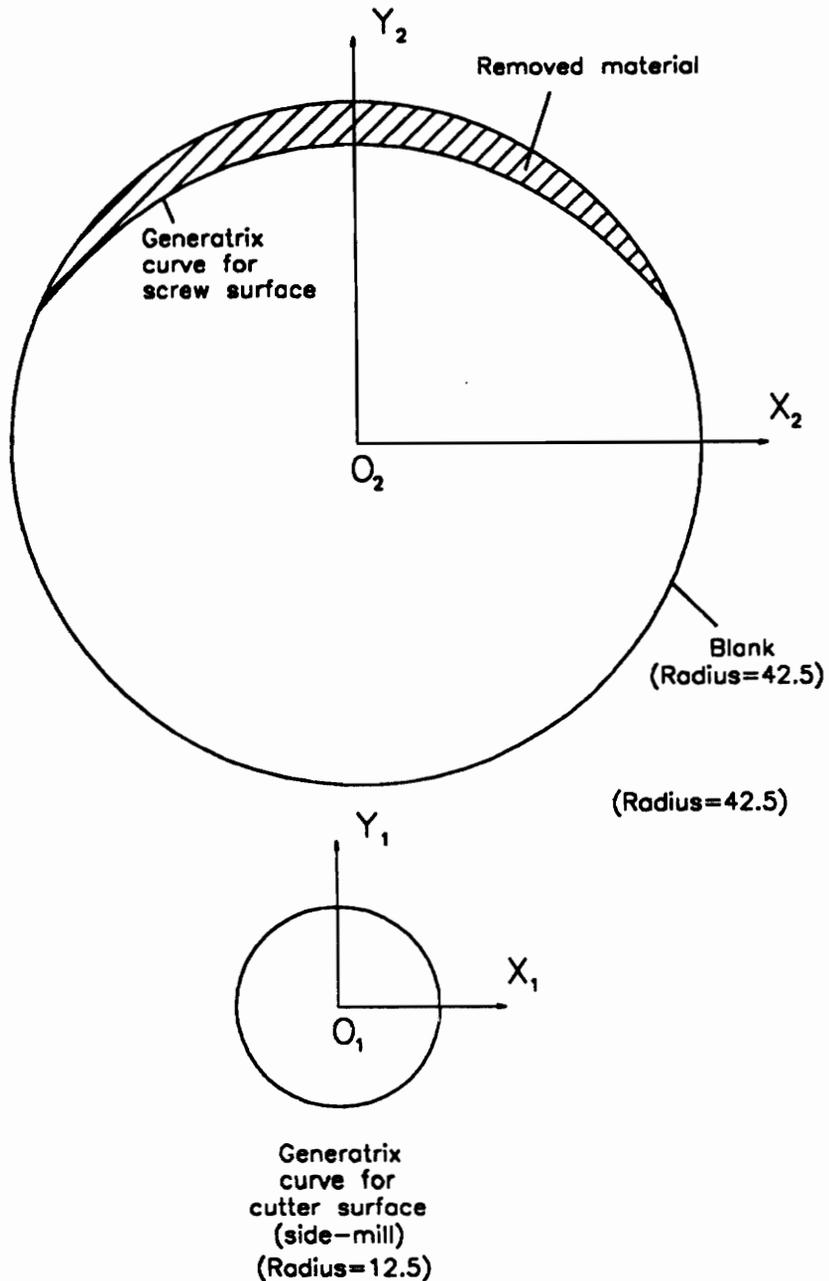


Figure 21. Side-Mill Operation - Generatrix Curves

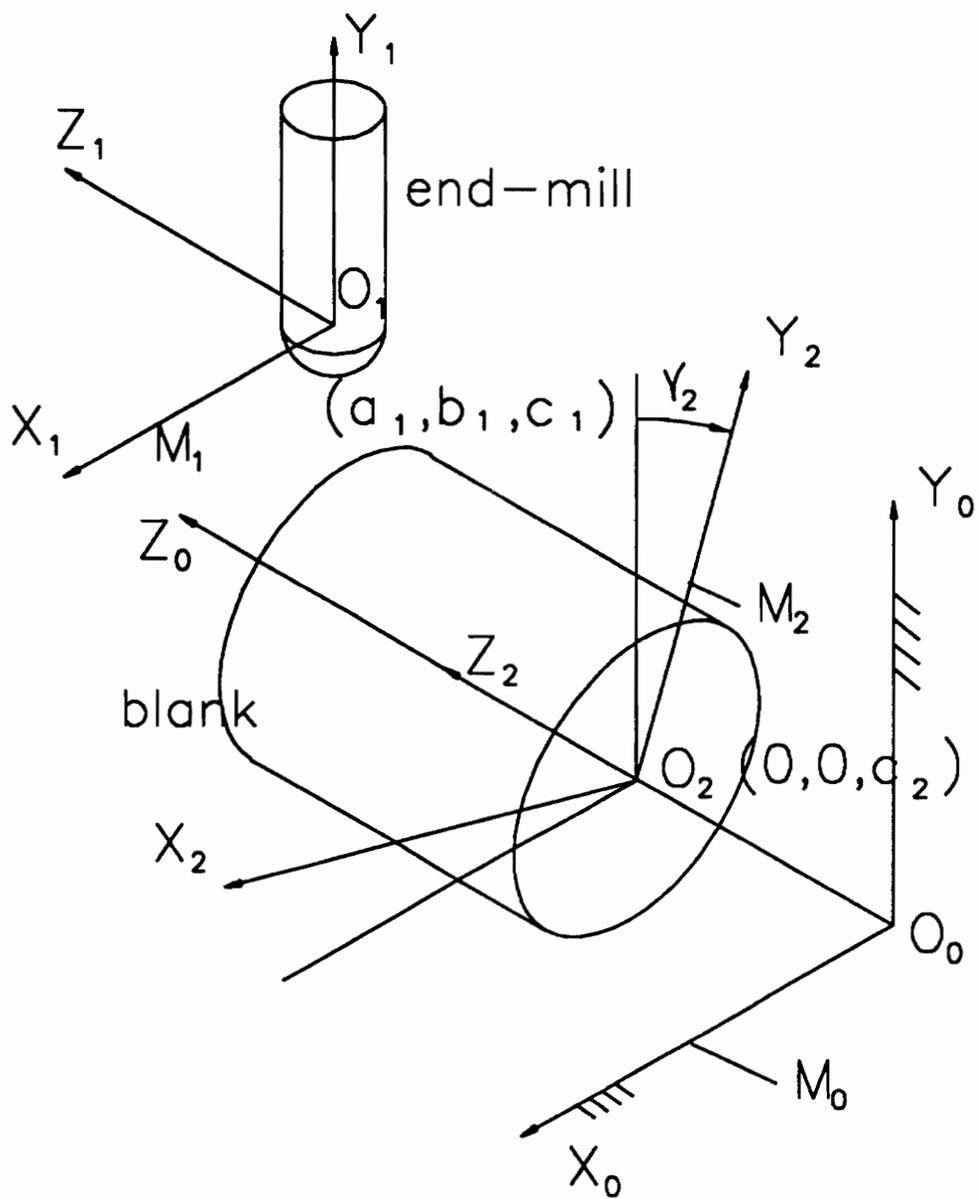


Figure 22. End-Mill Operation - Coordinate Systems

where $0 \leq u \leq 2\pi$ and $0 \leq v \leq \pi$ and r is the radius of the end-mill, which was selected as 12.5 for this example. The location of the end-mill is given by

$$\begin{aligned} a_1 &= 0 \\ b_1 &= 50 \\ c_1 &= 0 \end{aligned} \tag{4.11}$$

The location and motion of the blank is described by

$$\begin{aligned} a_2 &= 0 \\ b_2 &= 0 \\ c_2 &= \frac{l\gamma_2}{2\pi} \end{aligned} \tag{4.12}$$

where the equation for c_2 describes the helical motion of the blank. l is the lead of the desired screw, and γ_2 is the angular rotation of the blank which is described by equation [4.13].

$$\gamma_2 = \omega_2 t \tag{4.13}$$

where ω_2 is the uniform angular velocity of rotation of the blank which is selected as 10 for this example. The lead of the screw was selected as 60 units for the purpose of the example.

The equation for the condition of contact as derived by the symbolic program for the above input is

$$u = 1.3821 \tag{4.14}$$

Equation [4.14] is the simplified version of the output from MACSYMA.

The equation of the swept surface from MACSYMA is

$$\begin{aligned}
 x_2 &= -12.5 \sin(10t) \sin(v) + 12.5 \cos(10t) \cos(u) \cos(v) + 50 \sin(10t) \\
 y_2 &= -12.5 \cos(10t) \sin(v) - 12.5 \sin(10t) \cos(u) \cos(v) + 50 \cos(10t) \quad [4.15]
 \end{aligned}$$

$$z_2 = 12.5 \sin(u) \cos(v) - \frac{300t}{\pi}$$

Sustituting for u from Equation [4.14] into Equation [4.15], a unique bi-parametric equation for the helically swept surface is obtained.

The generatrix curve of the helically swept surface is obtained by setting $z_2 = 0$ in Equation [4.15]. This results in the following relation between u , v and t

$$t = \frac{12.5 \pi \sin(u) \cos(v)}{300} \quad [4.16]$$

Sustituting for t from Equation [4.16] into Equation [4.15], the equation of the generatrix curve σ_2 is obtained. This curve is illustrated in Figure 23 superimposed on the original blank shape (circle), the shaded area in the figure is the removed material as a result of the end-mill machining operation. Figure 23 also shows the generatrix curve for the end-mill surface (σ_1), which is a semi-circle of radius r ($= 12.5$ in this particular example).

4.4 Wheel-Mill Machining

A wheel-mill is a milling cutter shaped as a grinding wheel. In this section the generation of a helically swept screw surface by a wheel-mill is discussed. The wheel-mill is modeled as a semi-circle swept around the mill axis. The cutter portion of the mill is the semi-circular annulus of the wheel-mill. Thus the generatrix curve of the wheel-mill is a semi-circle. The wheel-mill with all the coordinate systems is shown in Figure 24.

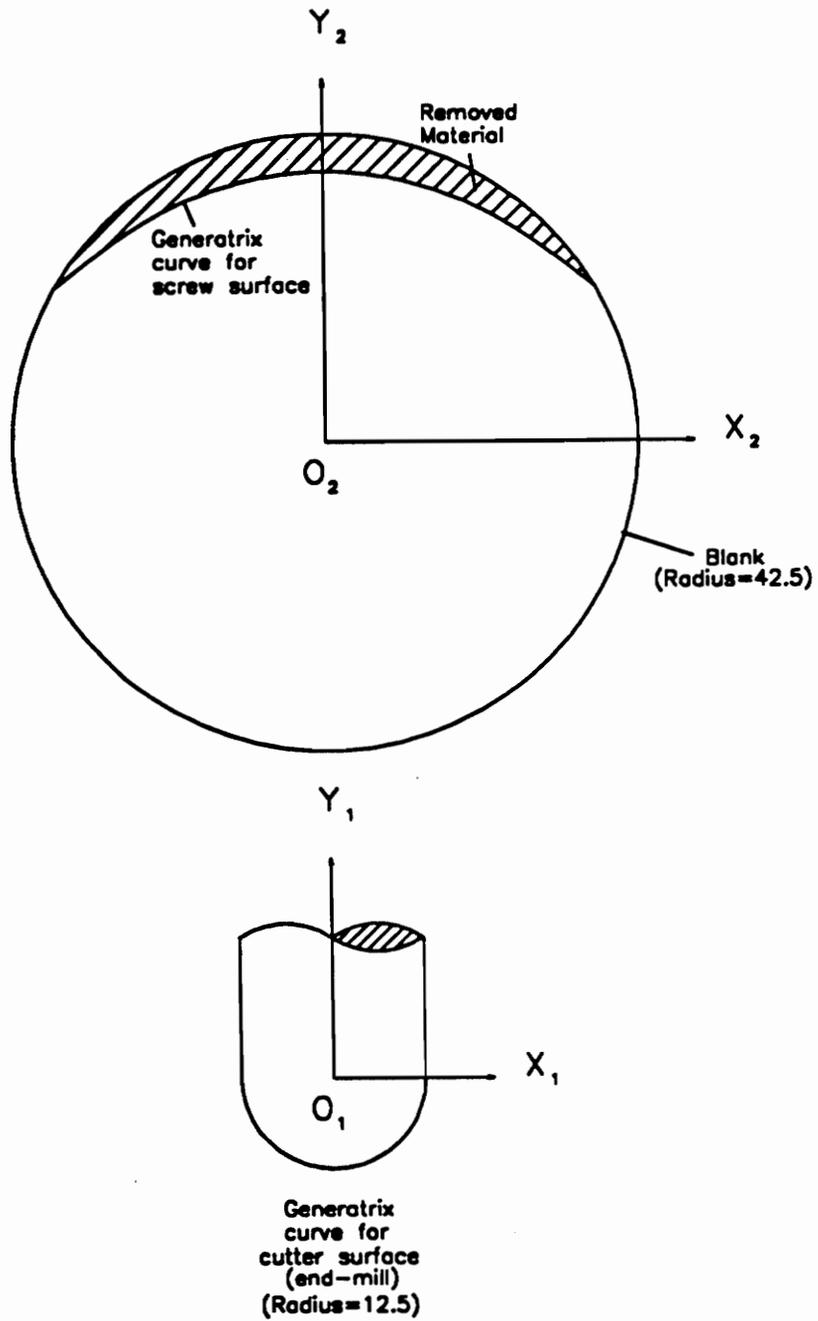


Figure 23. End-Mill Operation - Generatrix Curves

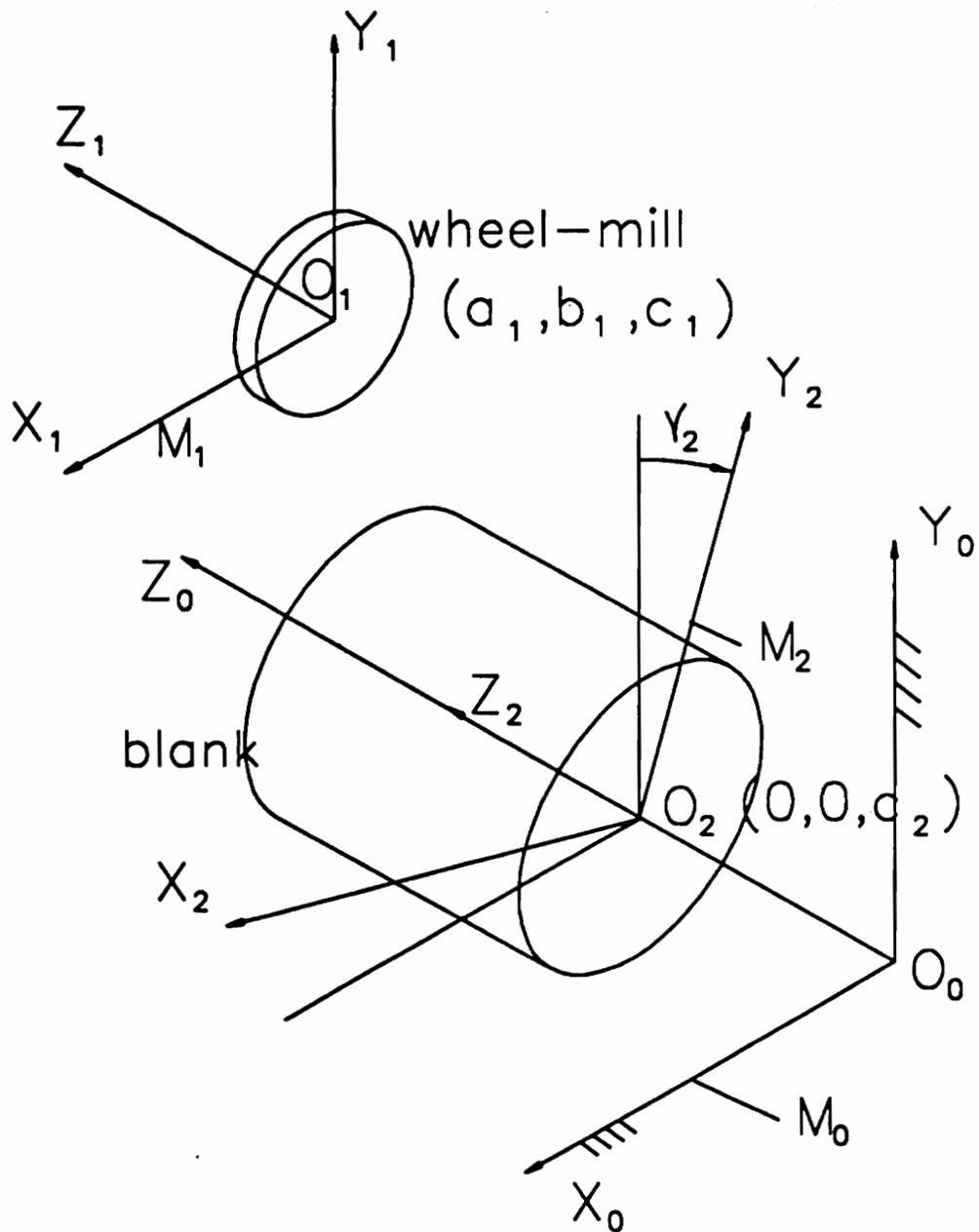


Figure 24. Wheel-Mill Operation - Coordinate Systems

The coordinate system M_1 is attached to the wheel-mill and the coordinate system M_2 is attached to the blank. The properties of the blank are similar to the previous two cases. The wheel-mill may be positioned in such a way that its axis is perpendicular to the resulting helix. For this reason another coordinate system is introduced in this case-study. This will be called the cutter coordinate system (M_c). The axis of the cutter is along the Z_c axis. The M_c coordinate system can rotate in the X_1Z_1 plane about the Y_1 axis. The angle M_c makes with the M_1 coordinate system is β measured in the positive sense from the X_1 axis. The axes of M_1 are parallel to the axes of M_0 as in the previous examples. The wheel-mill rotates about its own axis, which does not affect the relative motion between the wheel-mill and the blank. So the wheel-mill will be assumed to be stationary. The location of the wheel-mill in the fixed coordinate system can be varied. The blank is given a similar helical motion as in the previous two cases.

The surface of the cutter can be represented as

$${}^c\mathbf{P} = \begin{bmatrix} (t + r \sin(v)) \cos(u) \\ (t + r \sin(v)) \sin(u) \\ r \cos(v) \end{bmatrix} \quad [4.17]$$

where t is the radius of the flat portion of the cutter, r is the radius of the semi-circular generatrix curve of the cutter, $0 \leq u \leq 2\pi$ and $0 \leq v \leq \pi$. r is 12.5 and t is 20 for this example.

The surface of the cutter in the M_1 system is

$${}^1\mathbf{P} = [{}^1_c\mathbf{T}] {}^c\mathbf{P} \quad [4.18]$$

where $[{}^1_c\mathbf{T}]$ is

$$[{}^1_c\mathbf{T}] = \begin{bmatrix} \cos \beta & 0 & \sin \beta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \beta & 0 & \cos \beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad [4.19]$$

Equation [4.18] will be the input to the symbolic program. The location of the M_1 coordinate system is given by

$$\begin{aligned} a_1 &= 0 \\ b_1 &= 70 \\ c_1 &= 0 \end{aligned} \quad [4.20]$$

The location and motion of the blank is described by

$$\begin{aligned} a_2 &= 0 \\ b_2 &= 0 \\ c_2 &= \frac{l\gamma_2}{2\pi} \end{aligned} \quad [4.21]$$

where the equation for c_2 describes the helical motion of the blank. l is the lead of the desired screw, and γ_2 is the angular rotation of the blank which is described by equation [4.22].

$$\gamma_2 = \omega_2 t \quad [4.22]$$

where ω_2 is the uniform angular velocity of rotation of the blank which is selected as 10 for this example. The lead of the screw was selected as 60 units for the purpose of the example.

The equation for the condition of contact as derived by the symbolic program for the above input is

$$u = \frac{1875 \cos(v) \sin(v) + 3000 \cos(v)}{4375 \sin^3(v) + 7000 \sin(v)} + \pi \quad [4.23]$$

Equation [4.23] is the simplified version of the output from MACSYMA.

The equation of the swept surface from MACSYMA is

$$\begin{aligned}
x_2 &= (12.5 \sin(10t) \sin(u) + 12.5 \cos(10t) \cos(u)) \sin(v) + 20 \sin(10t) \sin(u) \\
&\quad + 20 \cos(10t) \cos(u) + 70 \sin(10t) \\
y_2 &= (12.5 \cos(10t) \sin(u) - 12.5 \sin(10t) \cos(u)) \sin(v) + 20 \cos(10t) \sin(u) \\
&\quad - 20 \sin(10t) \cos(u) + 70 \cos(10t)
\end{aligned} \tag{4.24}$$

$$z_2 = 12.5 \cos(v) - \frac{300t}{\pi}$$

Sustituting for u from Equation [4.23] into Equation [4.24], a unique bi-parametric equation for the helically swept surface is obtained.

The generatrix curve of the helically swept surface is obtained by setting $z_2 = 0$ in Equation [4.24]. This results in the following relation between u , v and t

$$t = \frac{12.5 \pi \cos(v)}{300} \tag{4.25}$$

Sustituting for t from Equation [4.25] into Equation [4.24], the equation of the generatrix curve σ_2 is obtained. This curve is illustrated in Figure 25 superimposed on the original blank shape (circle), the shaded area in the figure is the removed material as a result of the wheel-mill machining operation. Figure 25 also shows the generatrix curve for the wheel-mill surface (σ_1), which is a semi-circle of radius r ($= 12.5$ in this particular example).

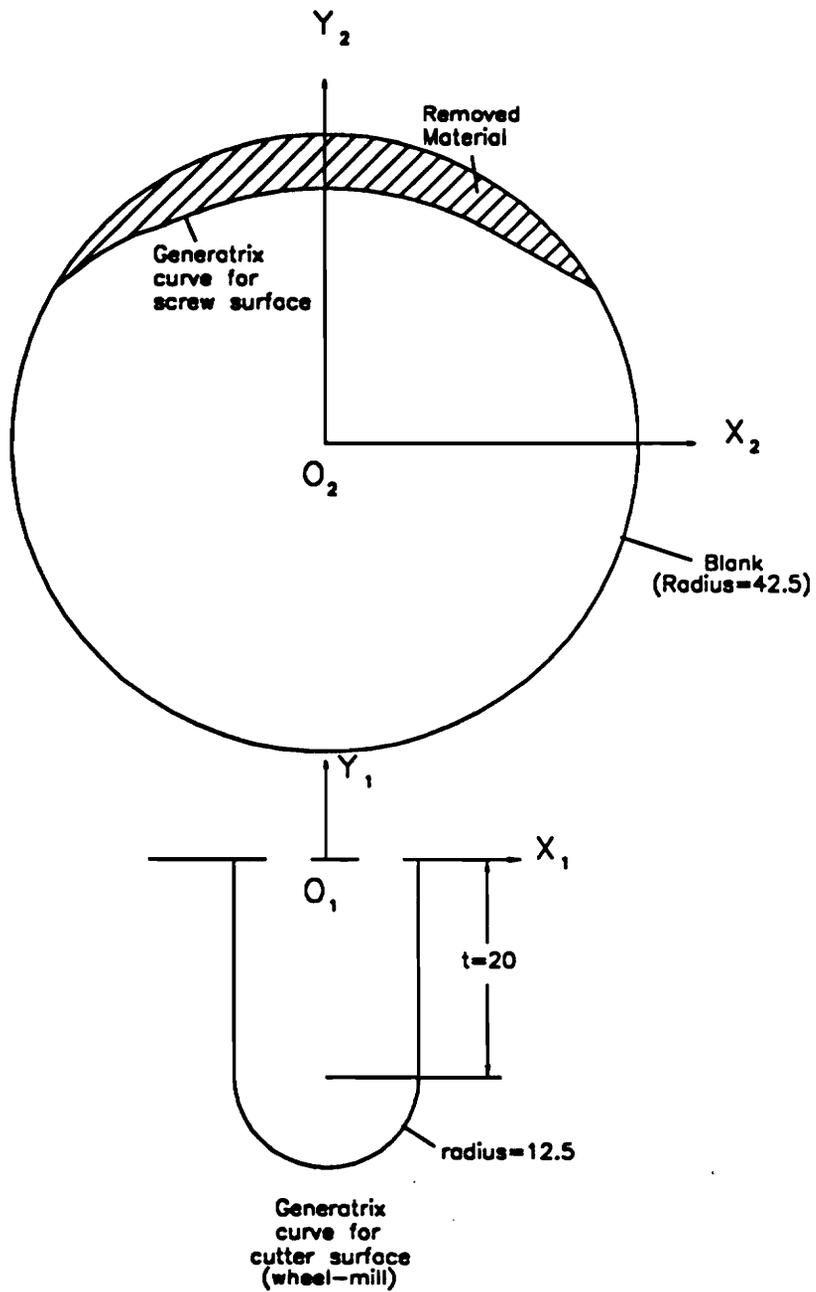


Figure 25. Wheel-Mill Operation - Generatrix Curves

Chapter 5

Conclusions

5.1 Summary

The concept of conjugate geometry has been discussed in Chapter 1. The theory of envelopes is introduced and using the definition of an envelope and examples in envelopes, it has been shown that the theory of envelopes is related to conjugate geometry. Two-dimensional and three-dimensional cases of conjugate curves/surfaces are introduced and a few applications of conjugate curves/surfaces in design and manufacturing have been cited.

Among the applications of conjugate curves/surfaces it has been discussed how the manufacture of helically swept extruder screw surfaces can be accomplished by using conjugate geometry. In the literature survey it has been shown how developments in symbolic and algebraic manipulation systems have led to the availability of powerful tools that can perform tedious manipulations with equations, matrices, etc. Another advantage of using symbolic manipulation software is in the fact that they can perform simplification. Symbolic systems work with exact quantities as opposed to

approximations like Fortran or Pascal do. By reviewing the capabilities of symbolic manipulation systems it was found to be a natural tool in solving conjugate geometry problems. Literature on cam mechanisms and other applications of conjugate geometry have been reviewed. The importance of computational geometry and its applications in the manufacturing industry have been discussed in the literature survey.

The development of a symbolic algorithm for use in MACSYMA was a natural outcome of knowledge of its capabilities. In Chapter 2 it has been shown how the two-dimensional general algorithm for conjugate geometry applications was developed. Also a case-specific three-dimensional symbolic algorithm was developed. The cases of interest were the manufacture of helically swept surfaces by milling operations. The two algorithms are also presented in this chapter.

The applications of conjugate geometry in design and manufacturing areas and the use of the symbolic programs developed in Chapter 2 have been demonstrated in Chapter 3. This has been accomplished by running a few known cam cases, one involute profile case and a manufacturing case using the symbolic algorithm. A typical MACSYMA session has been included in Appendix A. The results of the cases have been illustrated graphically as well as presented in equation form.

Chapter 4 deals with the manufacturing aspects of helically swept surfaces. A helically swept surface has been defined mathematically, and three schemes of manufacturing helically swept surfaces have been presented. The three schemes of manufacture have been discussed separately and in depth. Examples of the three methods of manufacture of helically swept surfaces have been run in MACSYMA using the three-dimensional symbolic program developed in Chapter 2. The results of the examples have been presented in both graphical form and in the form of equations. The importance of the generatrix curve of the the helically swept surface has been stressed and graphical illustrations of the generatrix curve for each manufacturing scheme have been presented. For ease of understanding, the generatrix curve has been superimposed on a cross-section of the original cylindrical blank.

5.2 Recommendations

The two aspects that need discussion here is the use of symbolic manipulation software to solve the conjugate geometry problems and the use of conjugate geometry to analyze design and manufacturing situations.

Conjugate geometry theory is a practical tool that can be used in any engineering situation where there is interaction between two curves/surfaces and the shape one curve/surface, along with some relative motion between the two curves/surfaces, dictates the shape of the other curve/surface. Thus the same principles can be applied in each situation. These situations can be both in the design and manufacturing areas. So the method of analysis does not depend on the situation. Also, conjugate geometry principles are relatively easy to apply and understand. One of the applications of conjugate geometry is in the cases of planar and spacial cams. The advantage in using conjugate geometry is that a generalized model can be developed of which all the planar and spacial cams are special cases. Thus in the case where a number of different examples have to be studied of which a general model can be developed, conjugate geometry is a very good application. Also, when one example has to tested under different circumstances; like finding the effect of varying one parameter while keeping the others constant; conjugate geometry is a perfect tool.

Performing an analysis involving conjugate geometry manually is typically very cumbersome as many matrix and arithmetic operations involving complex expressions are encountered often. In such cases programs in Fortran and Pascal could be written. But depending on the situation different programs may have to be written and also these programs handle only numeric computations. By using symbolic manipulation systems, cumbersome operations can be handled by the software and also a general program to handle different cases can be developed. Thus the program is not situation-dependent. The writing of the program is a one time job for a wide variety of applications. For example, The two-dimensional algorithm for conjugate geometry applications that was developed in Chapter 2 is general and it can handle any conjugate geometry problem in two-

dimensions. Thus the only job that needs to be performed in a specific situation is to identify the input to the symbolic program that represents that particular case. So the use of symbolic manipulation systems for conjugate geometry applications is highly recommended.

5.3 Future Work

The main objective of this research work was to develop a general algorithm for two-dimensional and three-dimensional conjugate geometry applications. To show that these algorithms worked a few case-studies were carried out. But if desired the algorithm for cam-follower cases can be further developed to incorporate conditions to avoid sharp edges and cusps. Also the pressure angle and curvature could be calculated. The program could be extended to make it a foolproof cam design system.

Bibliography

1. Battle, J. A., Foix, S. C., and Sanz, C. V., 1984, "On the Design of Milling Cutters or Grinding Wheels for Twist Drill Manufacture. A CAD Approach", *Proceedings of IMTDR*, pp. 315-320.
2. Boltyanskii, V. C., 1964, *"Envelopes"*, The Macmillan Company, New York, pp. 40-76.
3. Booy, M. L., 1978, "Geometry of Fully Wiped Twin-Screw Equipment", *Polymer Engineering and Science*, Vol. 18, No. 12, p. 973.
4. Chakraborty, J., and Dhande, S. G., 1977, *"Kinematics and Geometry of Planar and Spatial Cam Mechanisms"*, Wiley Eastern Limited, New Delhi, India.
5. Chen, C. H., 1990, "Application of Complementary Conjugate Surface in Surface Generation", *MI90 Conference*, Vol. 4, Atlanta, GA, pp. 21-25.
6. Dyson, A., 1969, *"Kinematics and Geometry of Gears in Three-Dimensions"*, Clarendon Press, Oxford.
7. Farin, G., 1988, *"Curves and Surfaces for Computer-Aided Design - A Practical Guide"*, Academic Press, Inc., San Diego, CA.

8. Faux, I. D., and Pratt, M. J., 1983, "*Computational Geometry for Design and Manufacture*", Ellis Horwood Publishers, Chichester.
9. Forrest, A. R., 1971, "Computational Geometry", *Proc. Roy. Soc. Lond.*, A321, pp. 187-195.
10. Freudenstein, F., 1973, "Kinematics: Past, Present and Future", *Mechanism and Machine Theory*, Vol. 8, No. 2, pp. 151-160.
11. Green, B. F., 1961, "Computer Languages for Symbol Manipulation", *I.R.E. Transactions*, HFE 2.
12. Groot, J. de., 1970, "*Bibliography in Kinematics, Vol. I, II*", University of Technology, Eindhoven.
13. Hamming, R. W., 1962, "*Numerical Methods for Scientists and Engineers*", McGraw Hill Book Co., New York.
14. Hearn, A. C., 1976, "A New Reduce Model for Algebraic Simplification", *Proceedings of the 1976 ACM Symposium on Symbolic and Algebraic Computation*, Yorktown Heights, New York, pp. 46-52.
15. Janssen, L. P. B. M., 1978, "*Twin Screw Extrusion*", Elsevier Scientific Publishing Company, Amsterdam.
16. Jefferys, W. H., 1971, "Automated Algebraic Manipulation in Celestial Mechanics", *ACM Proceedings of the Symposium on Symbolic and Algebraic Manipulation*, Los Angeles, CA, pp. 328-331.
17. Mabie, H. H., and Reinholtz, C. F., 1987, "*Mechanisms and Dynamics of Machinery*", John Wiley and Sons, New York.

18. Martin, W. A., and Fateman, R. J., 1971, "The MACSYMA System", *ACM Proceedings of the Symposium on Symbolic and Algebraic Manipulation*, Los Angeles, CA, pp. 59-73.
19. Martin, W. A., 1971, "Computer Input/Output of Mathematical Expressions", *ACM Proceedings of the Symposium on Symbolic and Algebraic Manipulation*, Los Angeles, CA, pp. 78-87.
20. Mortenson, M. E., 1985, *Geometri Modeling*, John Wiley and Sons, New York.
21. Moses, J., 1971a, "Algebraic Simplification: A Guide for the Perplexed", *Communications of the ACM*, Vol. 18, No. 8, pp. 527-537.
22. Moses, J., 1971b, "Symbolic Integration: The Stormy Decade", *ACM Proceedings of the Symposium on Symbolic and Algebraic Manipulation*, Los Angeles, CA, pp. 427-435.
23. Moses, J., 1974, "The Evolution of Algebraic Manipulation Algorithms", *Proceedings of the IFIP Conference*, Stockholm.
24. Raphael, B., Bobrow, D. G., Fein, L., and Young, J. W., 1966, "A Brief Survey of Computer Languages for Symbolic and Algebraic Manipulation", *Symbol Manipulation Languages and Techniques - Proceedings of the IFIP Working Conference on Symbol Manipulation Languages*, North-Holland Publishing Company, Amsterdam, pp. 1-54.
25. Raphael, B., 1966, "Aspects and Applications of Symbol Manipulation", *Proceedings of the 21st Natl. Conf. ACM*.
26. Rogers, D. F., and Adams, J. A., 1974, *Mathematical Elements for Computer Graphics*, McGraw Hill, New York.
27. Sandor, G. N., and Erdman, A. G., 1984, *Advanced Mechanism Design: Analysis and Synthesis, Vol. 2*, Prentice-Hall, Inc., Englewood Cliffs, NJ.

28. Shinan, L., and Jinguang, Z., 1990, "Envelop Method on the CAD of Profile Cutters", *Proceedings of the ASEE Conference on Computer Graphics (A New Vision on Engineering)*, Miami, FL, pp. 398-404.
29. Soni, A. H., 1968, "A Survey of Literature on Cam Design", *Mechanical Engineering News*, Vol. 5, No. 3, pp. 52-59.
30. Stevens, M. J., 1985, *Extruder Principles and Operation*, Elsevier Applied Science Publishers, London.
31. Stoutemeyer, D. R., 1976, "Qualitative Analysis of Mathematical Expressions Using Computer Symbolic Mathematics", *Proceedings of the 1976 ACM Symposium on Symbolic and Algebraic Computation*, Yorktown Heights, New York, pp. 97-104.
32. Symbolics, Inc., 1988, *MACSYMA Reference Manual Version 13*, Computer Aided Mathematics Group, Symbolics, Inc., Burlington, MA.
33. Symbolics, Inc., 1989, *MACSYMA*, A Presentation by the Computer Aided Mathematics Group, Symbolics, Inc., Burlington, MA.
34. Tadmor, Z., and Klein, I., 1970, *Engineering Principles of Plasticating Extrusion*, Van Nostrand Reinhold Company, New York.
35. Tobey, R. G., 1971, "Symbolic Mathematical Computation - Introduction and Overview", *ACM Proceedings of the Symposium on Symbolic and Algebraic Manipulation*, Los Angeles, CA, pp. 1-16.
36. Wilkes, M. V., 1964, "Lists and Why They Are Useful", *Proceedings of the 19th Natl. Conf. ACM*.

37. Winston, P. II., 1977, "*Artificial Intelligence*", Addison-Wesley Publishing Company, Reading, MA, pp. 263-286.
38. Wolfram, S., 1985, "Symbolic Mathematical Computation", *Communications of the ACM*, Vol. 28, No. 4, pp. 390-394.

Appendix A. MACSYMA Session

The transcript of a typical session in MACSYMA is presented below. This particular session involves the side-mill machining example discussed in Chapter 4.

```
(C3) /* Declaring dependencies of variables on parameters u, v and t */  
depends([x1,y1,z1],[u,v],[a1,a2,b1,b2,cc1,cc2,gamma2],t);
```

```
(D3) [X1(U, V), Y1(U, V), Z1(U, V), A1(T), A2(T), B1(T), B2(T), CC1(T), CC2(T),  
                                           GAMMA2(T)]
```

```
(C4) /* Defining the radius vector of point P in the cutter coordinate system */  
p1:matrix([x1],[y1],[z1],[1]);
```

```
(D4)      [ X1 ]  
          [   ]  
          [ Y1 ]  
          [   ]  
          [ Z1 ]  
          [   ]  
          [  1 ]
```

```
(C5) /* Defining the transformation matrix from cutter coordinate system to fixed  
coordinate system */  
t01:matrix([1,0,0,a1],[0,1,0,b1],[0,0,1,cc1],[0,0,0,1]);
```

```
(D5)      [ 1  0  0  A1 ]  
          [   ]  
          [ 0  1  0  B1 ]  
          [   ]  
          [ 0  0  1  CC1 ]  
          [   ]  
          [ 0  0  0  1  ]
```

```
(C6) /* Defining the transformation matrix from blank coordinate system to fixed
coordinate system */
t02:matrix([cos(gamma2),-sin(gamma2),0,a2],[sin(gamma2),cos(gamma2),0,b2],
[0,0,1,cc2],[0,0,0,1]);
```

$$(D6) \begin{bmatrix} \cos(\text{GAMMA2}) & -\sin(\text{GAMMA2}) & 0 & A2 \\ \sin(\text{GAMMA2}) & \cos(\text{GAMMA2}) & 0 & B2 \\ 0 & 0 & 1 & CC2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

```
(C7) /* Defining the transformation matrix from fixed coordinate system to blank
coordinate system */
t20:t02^^-1;
```

$$(D7) \text{ Col 1} = \begin{bmatrix} \frac{\cos(\text{GAMMA2})}{\sin^2(\text{GAMMA2}) + \cos^2(\text{GAMMA2})} \\ \frac{\sin(\text{GAMMA2})}{\sin^2(\text{GAMMA2}) + \cos^2(\text{GAMMA2})} \\ 0 \\ 0 \end{bmatrix}$$

$$\text{Col 2} = \begin{bmatrix} \frac{\sin(\text{GAMMA2})}{\sin^2(\text{GAMMA2}) + \cos^2(\text{GAMMA2})} \\ \frac{\cos(\text{GAMMA2})}{\sin^2(\text{GAMMA2}) + \cos^2(\text{GAMMA2})} \\ 0 \\ 0 \end{bmatrix} \quad \text{Col 3} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

$$\text{Col 4} = \begin{bmatrix} \frac{B2 \sin(\text{GAMMA2}) + A2 \cos(\text{GAMMA2})}{\sin^2(\text{GAMMA2}) + \cos^2(\text{GAMMA2})} \\ \frac{A2 \sin(\text{GAMMA2}) - B2 \cos(\text{GAMMA2})}{\sin^2(\text{GAMMA2}) + \cos^2(\text{GAMMA2})} \\ -CC2 \\ 1 \end{bmatrix}$$

```
(C8) /* Finding the radius vector of point P in the fixed coordinate system */
p21:t01.p1;
```

$$(D8) \begin{bmatrix} X1 + A1 \\ Y1 + B1 \\ Z1 + CC1 \\ 1 \end{bmatrix}$$

(C9) /* Finding the radius vector of point P in the blank coordinate system */
 p2:t20.p21;

(D9) MATRIX([
$$\frac{\sin(\text{GAMMA2}) (Y1 + B1)}{\sin^2(\text{GAMMA2}) + \cos^2(\text{GAMMA2})} + \frac{\cos(\text{GAMMA2}) (X1 + A1)}{\sin^2(\text{GAMMA2}) + \cos^2(\text{GAMMA2})}$$

$$- \frac{B2 \sin(\text{GAMMA2}) + A2 \cos(\text{GAMMA2})}{\sin^2(\text{GAMMA2}) + \cos^2(\text{GAMMA2})}], [\frac{\cos(\text{GAMMA2}) (Y1 + B1)}{\sin^2(\text{GAMMA2}) + \cos^2(\text{GAMMA2})}$$

$$- \frac{\sin(\text{GAMMA2}) (X1 + A1)}{\sin^2(\text{GAMMA2}) + \cos^2(\text{GAMMA2})} + \frac{A2 \sin(\text{GAMMA2}) - B2 \cos(\text{GAMMA2})}{\sin^2(\text{GAMMA2}) + \cos^2(\text{GAMMA2})}],$$

 [z1 - CC2 + CC1], [1])

(C10) /* Defining the partial derivatives for finding the normal vector */
 t1:'diff(x1,u);

(D10)
$$\frac{dx1}{du}$$

(C11) t2:'diff(y1,u);

(D11)
$$\frac{dy1}{du}$$

(C12) t3:'diff(z1,u);

(D12)
$$\frac{dz1}{du}$$

(C13) t4:'diff(x1,v);

(D13)
$$\frac{dx1}{dv}$$

(C14) t5:'diff(y1,v);

(D14)
$$\frac{dy1}{dv}$$

(C15) t6:'diff(z1,v);

(D15)
$$\frac{dz1}{dv}$$

```
(C16) /* Defining the normal vector in cutter coordinate system as cross product
of
tangent vectors along the u and v directions */
nlp:matrix([t2*t6-t3*t5],[t3*t4-t1*t6],[t1*t5-t2*t4]);
```

$$(D16) \begin{bmatrix} \frac{dy_1 dz_1}{du dv} - \frac{dy_1 dz_1}{dv du} \\ \frac{dx_1 dz_1}{du dv} - \frac{dx_1 dz_1}{dv du} \\ \frac{dx_1 dy_1}{du dv} - \frac{dx_1 dy_1}{dv du} \end{bmatrix}$$

```
(C17) /* Defining the magnitude of the normal vector */
e:sqrt(nlp[1,1]^2+nlp[2,1]^2+nlp[3,1]^2);
```

$$(D17) \text{SQRT}\left(\left(\frac{dy_1 dz_1}{du dv} - \frac{dy_1 dz_1}{dv du}\right)^2 + \left(\frac{dx_1 dz_1}{du dv} - \frac{dx_1 dz_1}{dv du}\right)^2 + \left(\frac{dx_1 dy_1}{du dv} - \frac{dx_1 dy_1}{dv du}\right)^2\right)$$

```
(C18) /* Defining each element of the unit normal vector */
eel:nlp[1,1]/e;
```

$$(D18) \frac{\frac{dy_1 dz_1}{du dv} - \frac{dy_1 dz_1}{dv du}}{\text{SQRT}\left(\left(\frac{dy_1 dz_1}{du dv} - \frac{dy_1 dz_1}{dv du}\right)^2 + \left(\frac{dx_1 dz_1}{du dv} - \frac{dx_1 dz_1}{dv du}\right)^2 + \left(\frac{dx_1 dy_1}{du dv} - \frac{dx_1 dy_1}{dv du}\right)^2\right)}$$

```
(C19) ee2:nlp[2,1]/e;
```

$$(D19) \frac{\frac{dx_1 dz_1}{du dv} - \frac{dx_1 dz_1}{dv du}}{\text{SQRT}\left(\left(\frac{dy_1 dz_1}{du dv} - \frac{dy_1 dz_1}{dv du}\right)^2 + \left(\frac{dx_1 dz_1}{du dv} - \frac{dx_1 dz_1}{dv du}\right)^2 + \left(\frac{dx_1 dy_1}{du dv} - \frac{dx_1 dy_1}{dv du}\right)^2\right)}$$

```
(C20) ee3:nlp[3,1]/e;
```

$$(D20) \frac{\frac{dx_1 dy_1}{du dv} - \frac{dx_1 dy_1}{dv du}}{\text{SQRT}\left(\left(\frac{dy_1 dz_1}{du dv} - \frac{dy_1 dz_1}{dv du}\right)^2 + \left(\frac{dx_1 dz_1}{du dv} - \frac{dx_1 dz_1}{dv du}\right)^2 + \left(\frac{dx_1 dy_1}{du dv} - \frac{dx_1 dy_1}{dv du}\right)^2\right)}$$

```

(C21) /* Defining the unit normal vector */
nlpl:matrix([ee1],[ee2],[ee3],[0]);

(D21) MATRIX(((--- --- - --- ---)/SQRT((--- --- - --- ---)
              du dv dv du du dv dv du
+ (--- --- - --- ---) + (--- --- - --- ---) )],
              dx1 dz1 dx1 dz1 2 dx1 dy1 dx1 dy1 2
              dv du du dv du dv dv du
              dx1 dz1 dx1 dz1
              dv du du dv
[-----]),
SQRT((--- --- - --- ---) + (--- --- - --- ---) + (--- --- - --- ---)
      du dv dv du dv du du dv du dv du
      dx1 dy1 dx1 dy1
      du dv dv du
[-----]), [0])
SQRT((--- --- - --- ---) + (--- --- - --- ---) + (--- --- - --- ---)
      du dv dv du dv du du dv dv du
      dx1 dy1 dx1 dy1
      du dv dv du
[-----]), [0])

(C22) /* Defining the transformation matrix for unit normal vector from the
       cutter coordinate system to the fixed coordinate system */
trans:matrix([1,0,0,0],[0,1,0,0],[0,0,1,0],[0,0,0,0]);

(D22)
      [ 1 0 0 0 ]
      [ 0 1 0 0 ]
      [ 0 0 1 0 ]
      [ 0 0 0 0 ]

(C23) /* Defining the unit normal vector in the fixed coordinate system */
n0pl:trans.nlpl;

(D23) MATRIX(((--- --- - --- ---)/SQRT((--- --- - --- ---)
              du dv dv du du dv dv du
+ (--- --- - --- ---) + (--- --- - --- ---) )],
              dx1 dz1 dx1 dz1 2 dx1 dy1 dx1 dy1 2
              dv du du dv du dv dv du
              dx1 dz1 dx1 dz1
              dv du du dv
[-----]),
SQRT((--- --- - --- ---) + (--- --- - --- ---) + (--- --- - --- ---)
      du dv dv du dv du du dv du dv du
      dx1 dy1 dx1 dy1
      du dv dv du
[-----]), [0])
SQRT((--- --- - --- ---) + (--- --- - --- ---) + (--- --- - --- ---)
      du dv dv du dv du du dv dv du
      dx1 dy1 dx1 dy1
      du dv dv du
[-----]), [0])

```

```
(C24) /* Defining derivatives w.r.t. t */
```

```
gamma2_dot:'diff(gamma2,t);
```

```
(D24) 
$$\frac{dGAMMA2}{dT}$$

```

```
(C25) a2_dot:'diff(a2,t);
```

```
(D25) 
$$\frac{dA2}{dT}$$

```

```
(C26) b2_dot:'diff(b2,t);
```

```
(D26) 
$$\frac{dB2}{dT}$$

```

```
(C27) c2_dot:'diff(cc2,t);
```

```
(D27) 
$$\frac{dCC2}{dT}$$

```

```
(C28) /* Defining the transformation matrix for finding the velocity vector of b  
lank coordinate system
```

```
w.r.t. fixed coordinate system */
```

```
trans2:matrix([gamma2_dot*(-sin(gamma2)),gamma2_dot*(-cos(gamma2)),0,a2_dot],  
[gamma2_dot*cos(gamma2),  
gamma2_dot*(-sin(gamma2)),0,b2_dot],[0,0,0,c2_dot],[0,0,0,0]);
```

```
(D28) 
$$\begin{bmatrix} -\sin(GAMMA2) \frac{dGAMMA2}{dT} & -\cos(GAMMA2) \frac{dGAMMA2}{dT} & 0 & \frac{dA2}{dT} \\ \cos(GAMMA2) \frac{dGAMMA2}{dT} & -\sin(GAMMA2) \frac{dGAMMA2}{dT} & 0 & \frac{dB2}{dT} \\ 0 & 0 & 0 & \frac{dCC2}{dT} \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

```

```
(C29) /* Defining the velocity vector of the blank in the fixed coordinate system */
v0p2:trans2.p2;
```

$$\begin{aligned}
 (D29) \text{ MATRIX} & \left(\left[-\text{SIN}(\text{GAMMA2}) \frac{d\text{GAMMA2}}{dT} \left(\frac{\text{SIN}(\text{GAMMA2}) (Y1 + B1)}{\text{SIN}^2(\text{GAMMA2}) + \text{COS}^2(\text{GAMMA2})} \right. \right. \right. \\
 & + \frac{\text{COS}(\text{GAMMA2}) (X1 + A1)}{\text{SIN}^2(\text{GAMMA2}) + \text{COS}^2(\text{GAMMA2})} - \frac{B2 \text{SIN}(\text{GAMMA2}) + A2 \text{COS}(\text{GAMMA2})}{\text{SIN}^2(\text{GAMMA2}) + \text{COS}^2(\text{GAMMA2})} \left. \right) \\
 & - \text{COS}(\text{GAMMA2}) \frac{d\text{GAMMA2}}{dT} \left(\frac{\text{COS}(\text{GAMMA2}) (Y1 + B1)}{\text{SIN}^2(\text{GAMMA2}) + \text{COS}^2(\text{GAMMA2})} \right. \\
 & \left. \left. - \frac{\text{SIN}(\text{GAMMA2}) (X1 + A1)}{\text{SIN}^2(\text{GAMMA2}) + \text{COS}^2(\text{GAMMA2})} + \frac{A2 \text{SIN}(\text{GAMMA2}) - B2 \text{COS}(\text{GAMMA2})}{\text{SIN}^2(\text{GAMMA2}) + \text{COS}^2(\text{GAMMA2})} + \frac{dA2}{dT} \right] \right), \\
 & \left[\text{COS}(\text{GAMMA2}) \frac{d\text{GAMMA2}}{dT} \left(\frac{\text{SIN}(\text{GAMMA2}) (Y1 + B1)}{\text{SIN}^2(\text{GAMMA2}) + \text{COS}^2(\text{GAMMA2})} + \frac{\text{COS}(\text{GAMMA2}) (X1 + A1)}{\text{SIN}^2(\text{GAMMA2}) + \text{COS}^2(\text{GAMMA2})} \right. \right. \\
 & \left. \left. - \frac{B2 \text{SIN}(\text{GAMMA2}) + A2 \text{COS}(\text{GAMMA2})}{\text{SIN}^2(\text{GAMMA2}) + \text{COS}^2(\text{GAMMA2})} \right) - \text{SIN}(\text{GAMMA2}) \frac{d\text{GAMMA2}}{dT} \right. \\
 & \left. \left(\frac{\text{COS}(\text{GAMMA2}) (Y1 + B1)}{\text{SIN}^2(\text{GAMMA2}) + \text{COS}^2(\text{GAMMA2})} - \frac{\text{SIN}(\text{GAMMA2}) (X1 + A1)}{\text{SIN}^2(\text{GAMMA2}) + \text{COS}^2(\text{GAMMA2})} \right) \right. \\
 & \left. + \frac{A2 \text{SIN}(\text{GAMMA2}) - B2 \text{COS}(\text{GAMMA2})}{\text{SIN}^2(\text{GAMMA2}) + \text{COS}^2(\text{GAMMA2})} + \frac{dB2}{dT} \right], \left[\frac{dCC2}{dT} \right], [0])
 \end{aligned}$$

```
(C30) /* Defining the relative velocity vector as the velocity vector of the bla
nk
since the velocity of the cutter is zero */
v0p12:v0p2;
```

```
(D30) MATRIX([ - SIN(GAMMA2)  $\frac{dGAMMA2}{dT}$  (  $\frac{SIN(GAMMA2) (Y1 + B1)}{SIN^2(GAMMA2) + COS^2(GAMMA2)}$ 
+  $\frac{COS(GAMMA2) (X1 + A1)}{SIN^2(GAMMA2) + COS^2(GAMMA2)}$  -  $\frac{B2 SIN(GAMMA2) + A2 COS(GAMMA2)}{SIN^2(GAMMA2) + COS^2(GAMMA2)}$ 
- COS(GAMMA2)  $\frac{dGAMMA2}{dT}$  (  $\frac{COS(GAMMA2) (Y1 + B1)}{SIN^2(GAMMA2) + COS^2(GAMMA2)}$ 
-  $\frac{SIN(GAMMA2) (X1 + A1)}{SIN^2(GAMMA2) + COS^2(GAMMA2)}$  +  $\frac{A2 SIN(GAMMA2) - B2 COS(GAMMA2)}{SIN^2(GAMMA2) + COS^2(GAMMA2)}$  ) +  $\frac{dA2}{dT}$  ],
[ COS(GAMMA2)  $\frac{dGAMMA2}{dT}$  (  $\frac{SIN(GAMMA2) (Y1 + B1)}{SIN^2(GAMMA2) + COS^2(GAMMA2)}$  +  $\frac{COS(GAMMA2) (X1 + A1)}{SIN^2(GAMMA2) + COS^2(GAMMA2)}$ 
-  $\frac{B2 SIN(GAMMA2) + A2 COS(GAMMA2)}{SIN^2(GAMMA2) + COS^2(GAMMA2)}$  ) - SIN(GAMMA2)  $\frac{dGAMMA2}{dT}$ 
(  $\frac{COS(GAMMA2) (Y1 + B1)}{SIN^2(GAMMA2) + COS^2(GAMMA2)}$  -  $\frac{SIN(GAMMA2) (X1 + A1)}{SIN^2(GAMMA2) + COS^2(GAMMA2)}$ 
+  $\frac{A2 SIN(GAMMA2) - B2 COS(GAMMA2)}{SIN^2(GAMMA2) + COS^2(GAMMA2)}$  ) +  $\frac{dB2}{dT}$  ], [  $\frac{dCC2}{dT}$  ], [ 0 ] )
```

(C31) /* Defining the condition of contact as the orthogonality of the relative velocity vector and the normal vector */
 constraint:transpose(v0p12).n0p1=0;

$$\begin{aligned}
 (D31) & \left(-\sin(\text{GAMMA2}) \frac{d\text{GAMMA2}}{dT} \left(\frac{\sin(\text{GAMMA2}) (Y1 + B1)}{\sin^2(\text{GAMMA2}) + \cos^2(\text{GAMMA2})} \right. \right. \\
 & + \frac{\cos(\text{GAMMA2}) (X1 + A1)}{\sin^2(\text{GAMMA2}) + \cos^2(\text{GAMMA2})} - \frac{B2 \sin(\text{GAMMA2}) + A2 \cos(\text{GAMMA2})}{\sin^2(\text{GAMMA2}) + \cos^2(\text{GAMMA2})} \left. \right) \\
 & - \cos(\text{GAMMA2}) \frac{d\text{GAMMA2}}{dT} \left(\frac{\cos(\text{GAMMA2}) (Y1 + B1)}{\sin^2(\text{GAMMA2}) + \cos^2(\text{GAMMA2})} \right. \\
 & - \frac{\sin(\text{GAMMA2}) (X1 + A1)}{\sin^2(\text{GAMMA2}) + \cos^2(\text{GAMMA2})} + \frac{A2 \sin(\text{GAMMA2}) - B2 \cos(\text{GAMMA2})}{\sin^2(\text{GAMMA2}) + \cos^2(\text{GAMMA2})} \left. \right) + \frac{dA2}{dT} \\
 & \left(\frac{dY1}{dU} \frac{dZ1}{dV} - \frac{dY1}{dV} \frac{dZ1}{dU} \right) / \text{SQRT} \left(\left(\frac{dY1}{dU} \frac{dZ1}{dV} - \frac{dY1}{dV} \frac{dZ1}{dU} \right)^2 + \left(\frac{dX1}{dV} \frac{dZ1}{dU} - \frac{dX1}{dU} \frac{dZ1}{dV} \right)^2 \right) \\
 & + \left(\frac{dX1}{dU} \frac{dY1}{dV} - \frac{dX1}{dV} \frac{dY1}{dU} \right)^2 + \left(\cos(\text{GAMMA2}) \frac{d\text{GAMMA2}}{dT} \right. \\
 & \left. \left(\frac{\sin(\text{GAMMA2}) (Y1 + B1)}{\sin^2(\text{GAMMA2}) + \cos^2(\text{GAMMA2})} + \frac{\cos(\text{GAMMA2}) (X1 + A1)}{\sin^2(\text{GAMMA2}) + \cos^2(\text{GAMMA2})} \right. \right. \\
 & \left. \left. - \frac{B2 \sin(\text{GAMMA2}) + A2 \cos(\text{GAMMA2})}{\sin^2(\text{GAMMA2}) + \cos^2(\text{GAMMA2})} - \sin(\text{GAMMA2}) \frac{d\text{GAMMA2}}{dT} \right) \right. \\
 & \left. \left(\frac{\cos(\text{GAMMA2}) (Y1 + B1)}{\sin^2(\text{GAMMA2}) + \cos^2(\text{GAMMA2})} - \frac{\sin(\text{GAMMA2}) (X1 + A1)}{\sin^2(\text{GAMMA2}) + \cos^2(\text{GAMMA2})} \right. \right. \\
 & \left. \left. + \frac{A2 \sin(\text{GAMMA2}) - B2 \cos(\text{GAMMA2})}{\sin^2(\text{GAMMA2}) + \cos^2(\text{GAMMA2})} + \frac{dB2}{dT} \left(\frac{dX1}{dV} \frac{dZ1}{dU} - \frac{dX1}{dU} \frac{dZ1}{dV} \right) \right) \right. \\
 & \left. / \text{SQRT} \left(\left(\frac{dY1}{dU} \frac{dZ1}{dV} - \frac{dY1}{dV} \frac{dZ1}{dU} \right)^2 + \left(\frac{dX1}{dV} \frac{dZ1}{dU} - \frac{dX1}{dU} \frac{dZ1}{dV} \right)^2 + \left(\frac{dX1}{dU} \frac{dY1}{dV} - \frac{dX1}{dV} \frac{dY1}{dU} \right)^2 \right) \right. \\
 & \left. + \frac{dCC2}{dT} \left(\frac{dX1}{dU} \frac{dY1}{dV} - \frac{dX1}{dV} \frac{dY1}{dU} \right) \right) = 0
 \end{aligned}$$

(C32) matchdeclare(any,true);

(D32) DONE

(C33) /* Declare trigonometric identity for simplification */
trigsimp(sin(any)^2+cos(any)^2);
TIGGER\$DUA0:[MACSYMA_412.SHARE]trgsmp.fas;l being loaded.

(D33) 1

(C34) /* Evaluate condition of contact for the specified input for the specific side-mill case */

solct:ev(constraint,x1=v,y1=12.5*cos(u),z1=12.5*sin(u),
a1=0,b1=50,cc1=0,a2=0,b2=0,cc2=(300*t)/%pi,gamma2=10*t,diff);

$$\begin{aligned} & \text{(D34) } 12.5 \cos(U) (10 \cos(10 T) \left(\frac{\cos(10 T) V}{\sin^2(10 T) + \cos^2(10 T)} \right. \\ & + \left. \frac{\sin(10 T) (12.5 \cos(U) + 50)}{\sin^2(10 T) + \cos^2(10 T)} \right) - 10 \sin(10 T) \\ & \left(\frac{\cos(10 T) (12.5 \cos(U) + 50)}{\sin^2(10 T) + \cos^2(10 T)} - \frac{\sin(10 T) V}{\sin^2(10 T) + \cos^2(10 T)} \right) \\ & / \text{SQRT}(156.25 \sin^2(U) + 156.25 \cos^2(U)) \\ & + \frac{3750.0 \sin(U)}{\%PI \text{SQRT}(156.25 \sin^2(U) + 156.25 \cos^2(U))} = 0 \end{aligned}$$


```
(C37) /* Simplify the expression for the radius vector of point P */
solp21:ratsimp(solp2);
```

```
RAT replaced 12.5 by 25/2 = 12.5
```

```
RAT replaced 12.5 by 25/2 = 12.5
```

```
RAT replaced 12.5 by 25/2 = 12.5
```

```
(D37) [ 2 COS(10 T) V + 25 SIN(10 T) COS(U) + 100 SIN(10 T) ]
      [ ----- ]
      [          2          2          ]
      [ 2 SIN (10 T) + 2 COS (10 T) ]
      [ ]
      [ 2 SIN(10 T) V - 25 COS(10 T) COS(U) - 100 COS(10 T) ]
      [ - ----- ]
      [          2          2          ]
      [ 2 SIN (10 T) + 2 COS (10 T) ]
      [ ]
      [          25 %PI SIN(U) - 600 T ]
      [ ----- ]
      [          2 %PI ]
      [ ]
      [          1 ]
      [ ]
```

```
(D38)
(C39) CLOSEFILE();
```

```
DONE
```

Vita

Ravinder Srinivas Voruganti was born on November 21, 1965, in Devarkonda, A. P., India. After finishing High School in 1983 he joined the M. N. Regional Engineering College in Allahabad, India, where he received his Bachelor's Degree in Mechanical Engineering. On completion of his BS in 1987, he joined the R&D department of TVS Electronics, Bangalore, India as a Design Engineer. Sensing the urge to continue with higher education, he joined the Virginia Polytechnic Institute and State University in 1989 for MS in Mechanical Engineering.

