Computer modeling of spatial mechanisms for kinematic analysis and synthesis
by John Michael Lowell

A thesis submitted in partial fullfillment of the requirements for the degree of MASTER OF SCIENCE in Mechanical Engineering
Montana State University

Abstract:
A methodology is presented from which a general purpose computer code for determining the kinematics (position, velocity, and acceleration) of spatial mechanisms may be generated. This methodology permits the equations which describe the mechanism kinematics to be assembled and solved by a program which only needs to read a concise data base describing the mechanism and type of solution desired.

The inherent flexibility in this method permits modification of the mechanism configuration and characteristics by altering the data base describing the mechanism. Changing relatively few parameters in the data base makes possible the synthesis of motion. That is, for a given desired path or configuration of the mechanism, the values of the mechanism parameters (coordinates) can be determined without alterations in computer code. The method presented is well suited for multi-positional kinematic analysis and synthesis of complex spatial mechanisms with multiple degrees of freedom.

Two examples of kinematic analysis using this method are presented. The first is a three degree of freedom closed loop mechanism. The second is the robot arm used on the space shuttle, which is a six-degree-of freedom mechanism. Examples of position, velocity, and acceleration solutions are given for both mechanisms.

STATEMENT OF PERMISSION TO COPY

In presenting this thesis in partial fulfillment of the requirements for an advanced degree at Montana State University, I agree that the Library shall make it freely available for inspection. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by my major professor, or, in his absence, by the Director of Libraries. It is understood that any copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Signature John M. Lowell

Date May 28, 1982

COMPUTER MODELING OF SPATIAL MECHANISMS FOR KINEMATIC ANALYSIS AND SYNTHESIS

by

JOHN MICHAEL LOWELL

A thesis submitted in partial fullfillment
of the requirements for the degree

of

MASTER OF SCIENCE

in

Mechanical Engineering

Approved:

Chairperson, Graduate Committee

Head, Major Department

Graduate Dean

MONTANA STATE UNIVERSITY
Bozeman, Montana

May, 1982

ACKNOWLEDGEMENTS

The author wishes to thank Dr. D. O. Blackketter and Dr. H. W. Townes for their advice and assistance in performing this work.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

Symbol	Description
{AA}	column matrix containing second time derivative of {PA}
$\{AA_I\}$	second time derivative of $\{PA_I\}$ (Equation 2.12)
$[a_I]$	3 X 4 acceleration coefficient array for vector I (Equation 2.11)
c_I	4 X 1 cosine squared constraint coefficient array (Equation 2.5)
c'_I	first time derivative of c_I (Equation 2.9)
c''_I	second time derivative of c_I (Equation 2.13)
$[d_I]$	3 X 4 displacement coefficient array for vector I (Equation 2.2)
[D]	n X m assembled displacement matrix for model (Equation 3.1)
i	number of vectors in model
{IA}	array used to identify known and unknown accelerations
{IP}	array used to identify known and unknown geometry parameters
{IV}	array used to identify known and unknown velocities
j	number of vector paths in the model
[LI]	path information matrix, stores information as to which vector is in which path and its direction
m	number of kinematic equations, m is equal to j times three plus i

Symbol	Description
np	number of parameters defining model, np is equal to i times four
{PA}	column matrix containing all known and unknown geometry parameters
$\{PA_I\}$	1 X 4 parameter array for vector I (Equation 2.3)
{PD}	path displacement matrix, contains X, Y, and Z displacement from beginning to end of each vector path
{VA}	column matrix containing first time derivative of {PA}
$\{VA_I\}$	time derivative of $\{PA_I\}$ (Equation 2.8)
$[v_I]$	3 X 4 velocity coefficient array for vector I (Equation 2.7)

Subscripts

a	denotes parameter in direction angle coordinate system
c	denotes parameter in direction cosine coordinate system
I	corresponding to vector I

ABSTRACT

A methodology is presented from which a general purpose computer code for determining the kinematics (position, velocity, and acceleration) of spatial mechanisms may be generated. This methodology permits the equations which describe the mechanism kinematics to be assembled and solved by a program which only needs to read a concise data base describing the mechanism and type of solution desired.

The inherent flexibility in this method permits modification of the mechanism configuration and characteristics by altering the data base describing the mechanism. Changing relatively few parameters in the data base makes possible the synthesis of motion. That is, for a given desired path or configuration of the mechanism, the values of the mechanism parameters (coordinates) can be determined without alterations in computer code. The method presented is well suited for multi-positional kinematic analysis and synthesis of complex spatial mechanisms with multiple degrees of freedom.

Two examples of kinematic analysis using this method are presented. The first is a three degree of freedom closed loop mechanism. The second is the robot arm used on the space shuttle, which is a six-degree-of freedom mechanism. Examples of position, velocity, and acceleration solutions are given for both mechanisms.

CHAPTER I

INTRODUCTION

Rapid advances in robotics and automated equipment for industrial, underwater, and space applications have placed demands on engineers to design and analyze increasingly sophisticated three-dimensional mechanisms. While the use of vector mechanics to generate the equations necessary for the kinematic analysis of three-dimensional mechanisms is a well known procedure, the derivation and solution of the resulting equations is a lengthy process which typically has required a problem specific computer code for solution.

This thesis does not enlarge on the body of knowledge related to mechanism theory, but describes a methodology by which a three-dimensional, rigid body mechanism can be described with a relatively concise data base. This data base is used as input to a computer code which can assemble and solve the equations which determine the position, velocity, and acceleration of each parameter used to represent the mechanism model. The computer code is not problem dependent and can be used for either analysis or synthesis by changing the data base. Synthesis as used in this thesis refers to determining the configuration of the mechanism required to obtain a given position, velocity, or acceleration of some member or point in the mechanism. This computer code frees the designer from the necessity of developing programs which are only useful for a single mechanism and difficult to modify.

In application the method presented resembles the finite element technique [1] in the way mechanisms are modeled and the governing equations assembled. Matrix equations for the position, velocity, and acceleration relations of an arbitrary vector are derived using vector mechanics. The overall mechanism equations are assembled from these blocks of equations using information from a data base which describes the mechanism configuration and type of solution desired.

Kinematic problems are typically formulated as problems in vector analysis. However, a variety of solution techniques can be applied to the resulting vector equations. Some of the most commonly used solution techniques include graphical solutions, vector algebra, and complex number methods. While these techniques are methods of vector analysis, each method of solution has inherent advantages and disadvantages.

Historically, graphical techniques [2,3] have been the predominating technique for solution of the kinematics of planar mechanisms. Since solution by graphical methods requires drawing of the position, velocity and acceleration polygons, its accuracy is limited. Also, graphical methods become very tedious when multiple position solutions are desired. Graphical methods are difficult to use on spatial mechanisms.

Solutions by algebraic methods, whether based on complex numbers [2,3] or vector algebra [2,3], have several advantages over graphical techniques. Accuracy is limited only by the accuracy of the problem data and the numerical evaluation of the solution. Once the problem has been put in an algebraic form, a kinematic solution can be obtained at different positions of the mechanism by solving sets of simultaneous equations. However, an algebraic solution usually requires tedious mathematical manipulations to obtain a solution. Algebraic methods are also applicable to spatial mechanisms [3], although the algebra is more complicated than for planar mechanisms.

Chace, at the University of Michigan, was one of the first to use vector notation to obtain closed form solutions to three-dimensional vector equations [4]. The Chace approach consists of defining solutions to a vector tetrahedron equation in spherical coordinates according to the unknowns in the equations. The resulting nine possible solutions are reduced to explicit closed-forms and can be quickly evaluated. Chace used this technique to develop the ADAMS (Automatic Dynamic Analysis of Mechanical Systems) computer code, one of the more widely used industrial programs for analysis of three-dimensional mechanisms.

Garcia de Jalon, Serna, and Viadero [5] have offered a new technique using matrix methods and Lagrangian coordinates to solve the kinematics of spatial mechanisms. In this technique the members of a mechanism are represented by three or more points which are the Lagrangian coordinates for the member. Geometric constraint equations requiring constant distance between points, constant area of planar surfaces, and constant volume of solids are specified. Time derivatives of these equations are used for kinematic analysis. However, this method requires a previously known position which limits it's use for multi-position analysis.

The purpose of this thesis is to present a method using classical vector mechanics from which a general purpose computer code may be developed to perform kinematic analysis and synthesis of spatial mechanisms.

CHAPTER II

DEVELOPMENT OF VECTOR EQUATIONS

An arbitrary vector in a three-dimensional Cartesian coordinate system will be defined in terms of four parameters. One parameter is the length of the vector and the other three parameters define its direction. Direction can be defined using either direction angles or cosines. The three direction parameters are not independent as any one can be calculated given the other two. Equations defining the position, velocity, and acceleration relationships for an arbitrary vector will be developed in terms of these four parameters and their derivatives. One or more vectors can be used to define a member or locate points of interest in a mechanism.

With vectors defined in this manner, one has the option of developing the kinematic equations in terms of either the direction angles or the direction cosines. These two coordinate systems yield different forms of the vector path equations. Use of direction cosine coordinates to define vector direction results in nonlinear algebraic equations while use of direction angle coordinates to define vector direction results in nonlinear transcendental equations. Use of the direction angles also causes convergence problems when an iterative method is used to solve the nonlinear equations for the geometry of the mechanism. The time derivatives of the cosine squared constraint equations expressed in direction angle coordinates are identically satisfied when a vector is parallel to one of the axes. This can result in singular matrices for the velocity and acceleration relations at certain positions of a mechanism. Using direction cosines as parameters results in equations which take less computer time to construct, show better convergence characteristics when using an iterative technique to solve them, and their derivatives do not yield a singular set of equations when a vector is parallel to one of the axes. However, use of cosine variable parameters can result in problems when attempting to convert known and unknown angular velocities and accelerations into and out of the cosine variable coordinates from more commonly used units in other coordinate systems. Due to the advantages of using cosine variables, all equations were developed using the direction cosines.

For convenience and ease of programming, the parameters defining the arbitrary vector I shown in Figure 1 are stored in a parameter array {PA}, and subscripted in the following manner:

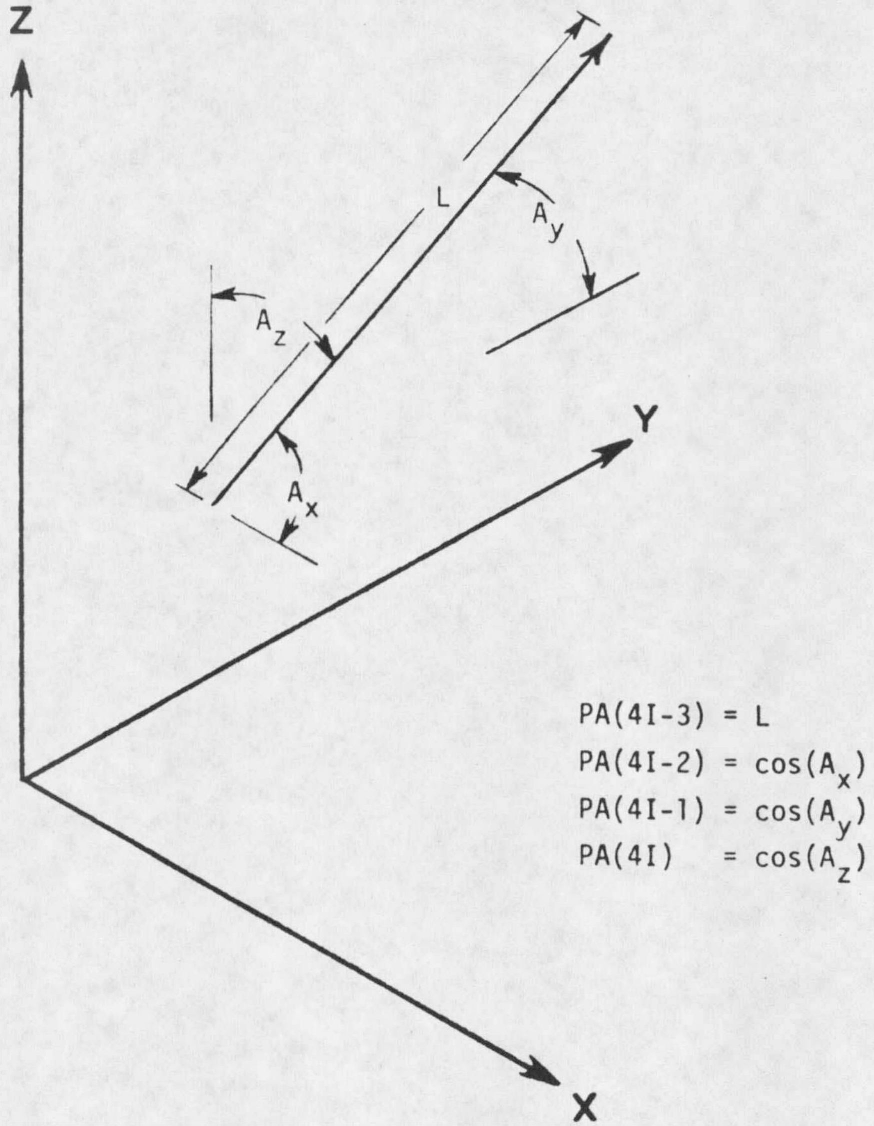

FIGURE 1 Schematic of Vector I