CR-171988

CSDL-T-953

COLLISION DETECTION FOR SPACECRAFT PROXIMITY OPERATIONS

by

Robin M. Vaughan

May 1987

Doctor of Philosophy Thesis
Massachusetts Institute of Technology

(NASA-CR-171988) COLLISION DETECTION FOR SPACECRAFT PROXIMITY OPERATIONS Ph.D. Thesis - MIT (Draper (Charles Stark) Lab.) 357 p Avail: NTIS HC A16/MF A01 CSCL 22A

N87-26935

Unclas
G3/17 0090667

The Charles Stark Draper Laboratory, Inc.
555 Technology Square
Cambridge, Massachusetts 02139

I hereby assign my copyright of this thesis to the Charles Stark Draper Laboratory, Inc., Cambridge, Massachusetts.

Robin M. Vaughan

Robin M. Vaughan

Permission is hereby granted by the Charles Stark Draper Laboratory, Inc. to the Massachusetts Institute of Technology to reproduce and to distribute copies of this thesis document in whole or in part.

PRECEDING PAGE BLANK NOT FILMED

COLLISION DETECTION FOR SPACECRAFT PROXIMITY OPERATIONS

by

Robin M. Vaughan

Submitted to the Department of Aeronautics and Astronautics in partial fulfillment of the requirements for the degree of Doctor of Philosophy

ABSTRACT

This thesis describes the development of a new collision detection algorithm to be used when two spacecraft are operating in the same vicinity. The two spacecraft are modelled as unions of convex polyhedra, where the polyhedron resulting from the union may be either convex or nonconvex. The relative motion of the two spacecraft is assumed to be such that one vehicle is moving with constant linear and angular velocity with respect to the other. The algorithm determines if a collision is possible and, if so, predicts the time when the collision will take place.

The theoretical basis for the new collision detection algorithm is the C-function formulation of the configuration space approach recently introduced by researchers in robotics. Three different types of C-functions are defined that model the contacts between the vertices, edges, and faces of the polyhedra representing the two spacecraft. These C-functions are used to formulate three "collision" conditions. The first of these conditions limits the points representing potential collisions to the zeros of the C-functions. The new algorithm is fundamentally a search for the smallest zero of any C-function that satisfies the second and third collision conditions.

The C-functions are shown to be transcendental functions of time for the assumed trajectory of the moving spacecraft. The zeros of these functions cannot be expressed in closed form. Therefore, numerical search procedures are developed to find all of the zeros of a C-function in specified bounded intervals of time. These bounded intervals of time are found by examining the second and third collision conditions.

The capabilities of the new algorithm are demonstrated for several example cases. These include examples of collisions determined by zeros of each of the three different types of C-functions. In addition to predicting the time of first contact of the polyhedra, the algorithm identifies the features of the two polyhedra that are touching at this time.

The new collision detection algorithm is the first such algorithm that is capable of solving the collision detection problem exactly for the case where

the moving object has constant linear and angular velocities. This is a significant improvement on previous collision detection algorithms described in the literature. In particular, the ability to handle constant angular velocity represents a more realistic type of rotational motion than those which have been used in other algorithms.

Thesis supervisor:	Bruce K. Walker
Title:	Associate Professor of Aeronautics and and Astronautics, M.I.T.
Thesis supervisor:	Richard H. Battin
Title:	Adjunct Professor of Aeronautics and Astronautics, M.I.T.
Thesis supervisor:	Wallace E. Vander Velde
Title:	Professor of Aeronautics and Astronautics M.I.T.
Thesis supervisor:	Edward V. Bergmann
Title:	Section Chief, Control Development C. S. Draper Laboratory, Inc.

Acknowledgements

I would like to thank the Charles Stark Draper Laboratory for providing the financial support for my six years of graduate education at M.I.T. The members of the laboratory have provided a stimulating and friendly environment in which to pursue my research. Thanks to the technical staff of divisions 10E and 10C who have shared their expertise over the years. Thanks also to Mario Santerelli and Dave Hauger of the Simulation Laboratory for the use of their computers to program the algorithm described in this thesis and for their patience in answering my questions. Very special thanks to Ed Bergmann, my supervisor in division 10C, for his unfailing enthusiam and optimism. He always believed I would finish this thesis even when I had doubts. His confidence in me and in my work has often inspired me to continue despite many frustrations and setbacks.

Many thanks to my advisor, Dr. Bruce Walker, for all the hours he spent listening to me discuss the problems I encountered in my thesis research and for his suggestions and insights. His careful reading of this document resulted in many improvements. I am happy to be his first "adivsee" to complete the PhD. I hope that there will be many other students who will benefit from his advice as I have.

Thanks also to the other members of my doctoral committee, Dr. Richard Battin and Dr. Wallace Vander Velde. Along with Bruce Walker and Ed Bergamnn, they have suffered through many hours of committee meetings while I tried to explain my results and provided useful criticisms on this document. Their influence on my graduate education has extended far beyond my doctoral research. Dr. Battin was my master's thesis advisor and inspired my interest in astrodymanics through his two excellent courses on this subject. Dr. Vander Velde has helped me to unravel the mysteries of probability theory, stochastic processes, and optimal control in the courses I have taken from him.

My years of residence in Ashdown House have enriched my stay at M.I.T. by giving me the opportunity to meet students from every department. Many of these fellow graduate students deserve my thanks for sharing the trials and tribulations of life at M.I.T. A special "thank you" goes to my former officemate at Draper, (now Dr.) Janice Voss, who patiently listened to my complaints and shared her experiences while finishing her PhD. My thanks also go to Brian Oki, Glen Kissel, Mark Smith, Eugene Gath,

Monty McGovern, and Bob Holt for the years of shared on-campus dining, Ashdown coffee hours, LSC movies, and Saturday night dinner excursions. Bob Holt also deserves credit for helping me to sort out the mathematics used in parts of Chapters 4 and 5. My deepest gratitude goes to Mark Smith for making my last semester at M.I.T. a very special one.

I am also deeply grateful to my family and friends in New Orleans for their continued encouragement and support. They have shared both the joy and the pain I have experienced during the past six years and kept me in touch with the world outside of academia. Special thanks to my parents, Robert W. and Rose Vaughan, and to my brother, Robert E. Vaughan, for always being there when I need them. Thanks also to Mike and Kay Lynch, M.I.T. alumni, who encouraged me to attend M.I.T. and helped me survive being here and to Barbara Parsons for her friendship since we attended high school together so many years ago.

This report was prepared under contract NAS9-17560 with the National Aeronautics and Space Administration.

Publication of this report does not constitute approval by the Charles Stark Draper Laboratory or NASA of the findings or conclusions contained herein. It is published solely for the exchange of ideas.

Contents

1 Introduction **1**
- 1.1 Survey of Related Work 2
- 1.2 Description of Thesis 7

2 The Configuration Space Approach **10**
- 2.1 Configuration Space and Configuration Space Obstacles . . 10
- 2.2 Contacts for Convex Polyhedra 12
  - 2.2.1 The C-functions for Convex Polyhedra 15
- 2.3 The C-functions and Construction of the Configuration Space Obstacle 22
  - 2.3.1 The Configuration Space Obstacle for Fixed Orientation of $\mathcal{A}$ 22
  - 2.3.2 The Six-dimensional Configuration Space Obstacle . 30
  - 2.3.3 The Three Collision Conditions 32
- 2.4 Applicability of C-functions for Convex Polyhedra 35
  - 2.4.1 Applicability Constraint Functions 37
  - 2.4.2 The Two Types of ACFs 43
- 2.5 The Third Collision Condition 50
- 2.6 Type (c) Contacts with Aligned Edges 52
  - 2.6.1 The Type (c) C-function for Aligned Edges 53
  - 2.6.2 Applicability of the Type (c) C-function 54
  - 2.6.3 The Third Collision Condition 57
  - 2.6.4 Detecting Collisions of Aligned Edges 59
- 2.7 Allowable Zeros for C-functions 60
- 2.8 Configuration Space Obstacles for Nonconvex Polyhedra . . 62
- 2.9 A Conceptual Collision Detection Procedure 63

3 Applicability for Constant Angular Velocity **65**
- 3.1 The ACFs for Constant Angular Velocity 66
  - 3.1.1 Quaternions and Quaternion Algebra 66
  - 3.1.2 Expressing the Vectors Representing the Features of $\mathcal{A}$ as Functions of Time 69
  - 3.1.3 Derivation of the Type (i) and (ii) ACFs as Functions of Time 71

3.1.4 The Behavior of an ACF as a Function of Time . . . 73
3.2 A Graphical Interpretation of the ACFs 76
3.2.1 The Type (*i*) ACFs 79
3.2.2 The Type (*ii*) ACFs 82
3.2.3 Constant Type (*i*) and (*ii*) ACFs 83
3.3 The Applicability and Orientation Timelines of a C-function 86
3.3.1 The Applicability Tests 91

4 The C-functions for Constant Linear and Angular Velocities 114
4.1 Derivation of the C-functions as Functions of Time 116
4.1.1 The Type (a) C-functions 116
4.1.2 The Type (b) C-functions 118
4.1.3 The Type (c) C-functions 119
4.1.4 The C-functions for Translation Only 122
4.1.5 The C-functions for Rotation Only 123
4.1.6 The Generalized C-function 124
4.2 Analytic Root Intervals for the Generalized C-function . . . 127
4.3 The Dynamic Spherical Approximation 130
4.4 The Physical Constraints 135
4.4.1 The Type (a) Physical Constraints 137
4.4.2 The Type (b) Physical Constraints 139
4.4.3 The Type (c) Physical Constraints 144
4.5 Summary of Methods for Bounding the Zero Search 149

5 Finding the Zeros of the C-functions 151
5.1 Class *I* 151
5.1.1 Case 1 152
5.1.2 Case 2 153
5.1.3 Cases 3 and 4 153
5.1.4 Case 5 155
5.1.5 Cases 6 and 7 155
5.1.6 Summary of Class *I* 164
5.2 Class *II* 164
5.2.1 Cases 1 and 2 166
5.2.2 Cases 3, 4, & 5 167
5.2.3 Summary of Class *II* 187

6 The Collision Detection Algorithm **189**
6.1 The Inputs 189
6.1.1 The Model for the Shapes of $\mathcal{A}$ and $\mathcal{B}$ 189
6.1.2 The Model for the Motion of $\mathcal{A}$ 193
6.2 The Algorithm 194
6.2.1 The Static and Dynamic Spherical Approximation Procedures 194
6.2.2 The Configuration Space Static Collision Detection Procedure 196
6.2.3 The Configuration Space Dynamic Collision Detection Procedure 197
6.2.4 The Hierarchy of the Algorithm 206
6.3 An Example 209
6.3.1 The Inputs 209
6.3.2 The Algorithm 214
6.4 Limitations of the Algorithm Due to Numerical Inaccuracies 232

7 Examples **234**
7.1 Rectangular Box Examples 236
7.2 Octahedron and Pyramid Examples 241
7.3 Examples of Two Nonconvex Polyhedra 247
7.4 Examples of Unexpected Types of Contact at Collision . . . 257
7.5 Summary of Examples 266
7.6 Comparison of Two Zero Search Methods 269
7.6.1 A Simpler Zero Search Method 269
7.6.2 The Examples 271
7.6.3 Comparison of the Execution Times for the Two Zero Search Methods 287

8 Conclusions and Recommendations **298**
8.1 Conclusions 298
8.2 Recommendations 300
8.2.1 Improving the Collision Detection Algorithm 300
8.2.2 Collision Avoidance Issues 304

References **309**

A The Intervals where the Second Derivative is Zero for Case 5 C-Functions of Class II **312**
A.1 Choosing the values of β_1 and β_2 312
A.2 Choosing the value of α 316
A.3 New Bounds on the Intervals where the Second Derivative is Zero 320
A.3.1 The Cases Where $m_2 \neq 0$ and $m_3 \neq 0$ 325
A.3.2 The Cases where $m_2 = 0$ or $m_3 = 0$ 336

B Two Iterative Methods for Finding the Zeros of a Function **343**
B.1 Linear Interpolation 343
B.2 The Secant Method 345

Chapter 1

Introduction

The next major goal of the U. S. space program is to develop a permanently manned space station. Current plans call for many other spacecraft to be operating in the vicinity of the station. These range from free-flying platforms in neighboring orbits to active maneuvering vehicles such as the OMV, OTV, and the shuttle orbiter.

One of the many areas of concern resulting from this environment is the problem of collision avoidance. This is one of several related issues in rendezvous and proximity operations which have been grouped under the heading of space traffic control [1,2]. The "establishment of a plan for evolving space traffic control development among elements of the space fleet" has been recognized as a design goal for orbital operations in the vicinity of the space station [3]. An important part of this will be a system which provides safety from collision between known participating vehicles [4].

The problem of avoiding a collision between two spacecraft can be divided into three main phases. The first phase is the detection of a potential collision at some time in the future. The second phase is the resolution of the collision predicted by the first phase. This involves selecting a suitable strategy for changing the motion of one or both of the vehicles to prevent the collision. The last phase is the implementation of this strategy, which is then monitored to confirm that the collision has been successfully avoided. All of these phases merit further study, but this thesis will concentrate on the first phase.

The goal of the thesis is to design a reliable algorithm for detection of potential collisions between two spacecraft. This will require models for the shapes of the two vehicles and for their relative motion. The algorithm will determine if a collision is possible and, if so, it will predict when the collision will occur. The time remaining before the potential collision is needed in the second phase to decide on an appropriate response.

Many maneuvers that are routinely used in spacecraft proximity operations require that two spacecraft operate near enough that their shapes and rotational motion must be taken into account to accurately predict any

potential collisions. As will be seen later in this thesis, the relative rotation of the two spacecraft greatly increases the complexity of the functions modelling contacts between them.

A summary of previous work on collision detection and avoidance in related fields is given in section 1.1. This includes a short introduction to the configuration space approach, which will be used throughout the remainder of this thesis. The collision detection algorithm implemented for this thesis is based on this approach. The reasons for choosing this approach and the assumptions used for the new algorithm are listed at the end of section 1.1.

The development of the new algorithm is outlined in section 1.2. The contents of the subsequent chapters of this thesis are also briefly described in this section.

1.1 Survey of Related Work

Aircraft collision prevention has long been of interest and references are available on many different airborne collision avoidance systems. These systems typically model the two aircraft as point masses and consider only their relative translation. These assumptions simplify the equations that are used to compute the projected time of a collision or the miss distance if no collision can occur [5]. The usual allowable distance between two aircraft in flight is large enough so that their shapes and relative rotation can be neglected. These systems are of limited interest for the spacecraft collision detection problem because the shapes and relative rotation of the of two spacecraft cannot be ignored, as mentioned in the previous section.

Two other fields in which related work has been done are mechanical design and robotics. Mechanical designers are concerned with the interaction of the components that make up a given assembly. In the case where the assembly contains no moving parts, relatively simple static tests are performed to determine if the components will fit together without overlap. When the assembly contains moving parts, more complex dynamic testing is needed to determine if any components interfere with each other during the motion. Advances in computer-aided design have motivated recent work in the mathematical formulation of these tests. Some commercial systems offer this type of testing for a specified set of trajectories of the

moving parts. However, these trajectories are usually quite restrictive and may not allow simultaneous rotation and translation [6].

An active area of robotics research in recent years is task planning. Its goal is to develop a high-level system for automatic planning of manipulator movements. It brings together techniques from such diverse disciplines as computational geometry, artificial intelligence, and control systems engineering. An essential part of any task planner is the ability to generate a collision-free trajectory for the manipulator through a workspace containing known obstacles. This has been termed the FINDPATH problem or the "piano movers" problem [7,8]. The trajectory of the manipulator is unknown and is used as a free parameter in the search for a safe path. The motion of the manipulator along its trajectory as a function of time is not considered at this stage of the task planner. This distinguishes FINDPATH from the collision detection problem where the trajectory is assumed known.

FINDPATH can be viewed as a generalization of the collision detection problem that is much more difficult to solve. Nevertheless, many techniques which have been proposed for the solution of the FINDPATH problem are applicable to the collision detection problem.

One of the earliest schemes suggested for solving the FINDPATH problem is the "generate and test" method. An initial trajectory is proposed and tested for potential collisions. If collisions are detected, a new path is generated and tested for collisions. This iteration is continued until a path is found which has no collisions. The collision detection problem must be solved in each iteration step of this scheme. Some of the methods that have been used to solve this part of a "generate and test" strategy are discussed in the following paragraphs.

Perhaps the simplest method for detecting collisions is multiple interference detection [9,10]. The objects are tested for overlap at discrete points along the proposed trajectory. There are several algorithms of varying complexity described in the computational geometry literature for determining if two stationary sets of points in space intersect [11]. This method can be used to predict the time to collision within the accuracy of the time step along the trajectory. The difficulty lies in the choice of the time step size. Smaller time steps require more applications of the interference detection algorithm, but a time step which is too large can result in a potential collision being missed. This has serious consequences for the reliability of the

collision detection algorithm.

Another method for collision detection is the swept volume method [10,12]. The idea here is to consider the volume "swept" out by the moving object along its trajectory as a new three-dimensional object. An interference calculation is then performed for this swept volume and the known obstacles to determine if they intersect. Depending on the nature of the motion, the swept volume can be very hard to represent exactly. Rotation of the moving object can result in especially complicated shapes. This problem can be circumvented by either restricting the class of allowable motions or by approximating the actual shape of the swept volume with a simpler shape. Although the latter approach can decrease the accuracy of the method, it usually simplifies the interference calculation. The main disadvantage of this method is that it is very difficult to obtain the first time of collision. The problem is reduced to a purely geometric level where the time parameter is not directly taken into account.

A more sophisticated approach is to view the objects as sets of points in four-dimensional space-time [10,13]. Once these sets are found for the given trajectory, an interference detection algorithm is applied in four dimensions. Theoretically, this method can be made to detect any collisions that will occur and can predict the time of first contact. However, it is difficult to represent the four-dimensional sets in any practical implementation. The mathematics quickly becomes quite complex even for simple shapes and motions.

The efficiency of the "generate and test" method for finding a collision-free path is dependent on the means for generating each new trajectory. Only localized information is available from the results of the previous collision detection process and the best choice of the next trajectory is not obvious. An alternative to this method which can potentially eliminate this difficulty is the configuration space approach [7,12,14]. This provides a global view of the problem by working directly with the constraints on the motion of the moving objects due to the presence of the obstacles.

The *configuration* of an object is a set of parameters that determine its position and orientation in space. The set of all possible configurations is the six-dimensional *configuration space*. The motion of the three-dimensional object in the original three-dimensional space is collapsed to the motion of a single point in the configuration space. The presence of the obstacles implies that there are configurations of the moving object

which are "forbidden." In these configurations, the moving object is either just touching the surface of an obstacle or overlaps an obstacle. The set of forbidden configurations generated by a particular obstacle is called a *configuration space obstacle.* The set of configurations outside of all the configuration space obstacles is called *free space.* Once the configuration space obstacles are known, a search algorithm is employed to generate a path through free space. All possible safe configurations are included in this space; no restrictions need be placed on the trajectory of the moving object while it remains in free space.

The configuration space obstacles are usually complex six-dimensional solids even for simple object shapes such as convex polyhedra. For this reason, less complicated approximations to these regions have been used. The obstacle itself can be enclosed in a series of six-dimensional convex polyhedra. Another approximation scheme involves slice projections and cross-sections of the configuration space obstacles for certain ranges of some subset of the six coordinates of the configuration space [7,14]. The shape of a particular robot has also been exploited to simplify these obstacle shapes [15]. The approximations are conservative in that they contain part of the free space as well as the entire configuration space obstacle. This is not of great concern in the robotics context of generating a safe path, but it can be a disadvantage to a collision detection algorithm. A potential collision could be predicted to occur sooner than it actually occurs or a collision could be predicted which does not occur causing a "false collision alarm" to be issued.

Another way to model a configuration space obstacle is to find the functions which generate its boundary. Such a function is called a *C-function* and the surface in configuration space where its value is zero is called a *C-surface.* Some part of each infinite C-surface forms part of the finite boundary of the configuration space obstacle. The obstacle can thus be viewed as the intersection of all the closed half-spaces where an appropriate set of C-functions have nonpositive or nonnegative values.

The C-functions were first defined for convex polyhedral objects in reference [14]. They were subsequently used in the development of the first published implementation of a system for the solution of the complete six degree-of-freedom FINDPATH problem [16]. These functions served two purposes in this system. First, they were used in heuristic algorithms for generating trajectory directions by moving along "levels" where the C-

functions maintained constant values. The function values gave a rough idea of the distance of the trajectory from the surface of the configuration space obstacle. Second, these functions were used to detect when the trajectory intersected the boundary of the configuration space obstacle. The value of an appropriate C-function passing through zero indicated that the trajectory had intersected the obstacle. A collision-free path was generated in pieces by first determining how far the trajectory could extend along the suggested direction before a C-surface was encountered and then using this information to suggest a new direction for the next piece of the trajectory such that it moved away from the configuration space obstacle.

In the system described above, collision detection was performed by moving in discrete increments along the suggested trajectory and checking the values of all appropriate C-functions at each increment. This strategy is similar to multiple interference detection and also involves a choice of step size. The accuracy of the computed collision points is limited to within this specified resolution. The primary justification for this approach is that, for most types of motion, the C-functions are transcendental and no explicit formula exists for obtaining their roots.

An alternate collision detection scheme has recently been described which exploits an advantageous limitation on the motion to simplify these functions to polynomials in time [17]. This is accomplished by considering only those trajectories that can be written as polynomial functions of time in the configuration space (not the original three-dimensional space). For the collision detection algorithm presented in reference [17], it is assumed that the trajectory of the configuration of the moving object follows a straight line in the configuration space. A straight line path through configuration space corresponds to translation with constant linear velocity and rotation about a fixed axis. However, the angular velocity is not constant. In fact, it decreases as time increases so that the object essentially stops rotating after a long period of time. A straight line path in configuration space results in cubic equations for the C-functions. The roots of these cubic C-functions can be found analytically and thus the collision points can be computed exactly.

The assumption of higher degree polynomial paths for the trajectory of the moving object in configuration space gives polynomials of degree five or greater for the C-functions. The roots of these functions cannot be found analytically, but standard iterative methods are available. In

general, the assumption of such a path in the configuration space yields a restricted class of motions in three-dimensional space, especially in the rotational component. Very high degree polynomials could perhaps be used to approximate more natural classes of motion, but the increased complexity of the methods used to find the roots of these polynomials would outweigh the benefit of having such a "simplified" form for the C-functions.

In spite of the difficulties described above, the C-function formulation of the configuration space approach is attractive for two reasons. First, it has the potential to provide exact values for the time of first contact. Second, it is guaranteed to detect a collision, if one will occur, within the accuracy of the computer's calculations. However, polynomial paths in the configuration space do not represent realistic models for the relative motion of two spacecraft. Therefore, a different trajectory must be chosen for the moving spacecraft.

Any useful assumption for the relative motion of the two spacecraft should combine simultaneous translation and rotation for the moving vehicle. The simplest trajectory that satisfies this requirement is characterized by constant linear and angular velocities. For this thesis, one of the spacecraft will be assumed to be moving with constant linear and angular velocity with respect to the other craft. The major difference between the algorithm developed based on this assumption and the one described in reference [17] is the rotational motion of the moving object. The assumption of a constant angular speed as well as a fixed axis of rotation represents a more natural type of motion for the moving spacecraft.

An overview of the development leading to the new collision detection algorithm is given in the following section.

1.2 Description of Thesis

The configuration space approach is discussed in detail in Chapter 2. The division of configuration space into configuration space obstacles and free space is more fully described for the case where the moving object and obstacle representing the two spacecraft are modelled as unions of convex polyhedra. The configuration space obstacles are characterized in terms of the C-functions introduced in references [7,14,16,17]. Three different types of C-functions are defined that model the contacts between different

features of the polyhedra representing the two spacecraft. The concepts of applicability and orientation of a C-function are introduced to explain the relationship between the C-functions and the configuration space obstacles. An explanation of these concepts is postponed until Chapter 2.

Chapter 2 also presents the three fundamental "collision" conditions. These conditions determine when the configuration of the moving polyhedron lies *on the boundary of* a configuration space obstacle. For these configurations, the surfaces of the two polyhedra are in contact but their interiors are disjoint. The first collision condition limits the points representing possible contacts between the two polyhedra to the zeros of the C-functions. The second collision condition further restricts the potential contacts to the zeros of a C-function that occur where the C-function is applicable. The exact nature of the third collision condition will not be given here, but any zero that occurs when a C-function is applicable must also satisfy this condition to represent a true collision between the two polyhedra.

The three collision conditions are used to outline a preliminary collision detection procedure at the end of Chapter 2. This procedure solves the collision detection problem by first finding all the zeros of the C-functions. It then checks these zeros to find the ones that represent points where the second and third collision conditions are satisfied. The first of these zeros must represent the first contact between the two polyhedra.

The presentation in Chapter 2 makes no assumption regarding the trajectory of the moving polyhedron. The three subsequent chapters apply the general theory of Chapter 2 to the particular case where the moving polyhedron has constant linear and angular velocities. Chapters 3, 4, and 5 discuss the implications of this assumption for each of the three collision conditions. The nature of the C-functions for this trajectory is such that it is more difficult to find the zeros of these functions than it is to determine the regions where the second and third collision conditions can be satisfied. Therefore, the second and third collision conditions are examined before the first collision condition.

The second collision condition is considered in Chapter 3. This chapter describes how to determine the applicability of the three types of C-functions when the moving polyhedron is rotating with constant angular velocity. The regions where a C-function is applicable can be determined exactly for this type of rotational motion.

The form of the C-functions for constant linear and angular velocities is examined in the first section of Chapter 4. These functions are shown to be transcendental functions of time. The remainder of Chapter 4 is concerned with the third collision condition. Approximate conditions are derived that determine when a C-function can satisfy the third condition if the moving polyhedron is translating with constant linear velocity.

Finally, the first collision condition is considered in Chapter 5. This chapter classifies each C-function into one of twelve cases and describes how to find the zeros of the C-function, if any, for each case. Certain details of the zero search procedure for the most general of these cases are presented in Appendices A and B.

The ideas developed in Chapters 3, 4, and 5 are combined to produce the new collision detection algorithm in Chapter 6. In contrast to the preliminary procedure described in Chapter 2, the new algorithm first finds the regions where the second and third collision conditions can be satisfied and then searches for the zeros of the C-functions in these regions. This algorithm is a hierarchy of tests representing more complicated, but more accurate, levels of modelling the interaction of the moving polyhedron with the stationary polyhedron.

The first section of Chapter 6 specifies the information required as input by the algorithm. The next sections describe four major procedures used by the algorithm. Then, the placement of these four procedures in the overall hierarchy of the algorithm is discussed. An example is included to illustrate a typical path through the hierarchy.

The results of applying the new algorithm to several example cases are presented in Chapter 7. Three different sets of moving and stationary polyhedra are considered. One example of each of the three types of contact at collision is given for each set of polyhedra. Chapter 7 also includes a brief discussion of the computation time required by the algorithm. A comparison is made between the time required by the new algorithm and a similar algorithm employing a simpler search procedure for the zeros of the C-functions.

The final chapter, Chapter 8, contains the conclusions and some recommendations for future work.

Chapter 2

The Configuration Space Approach

This chapter presents an overview of the configuration space approach as a means of modelling the interaction of the shapes of objects moving in space. The discussion is primarily concerned with those aspects of the theory that result from the assumption that the objects are represented by polyhedra. No specific assumptions will be made concerning the relative motion of the two objects. The implications of the assumption that one of the objects is moving with constant linear and angular velocities relative to the other object will be discussed in subsequent chapters.

The majority of the material presented in this chapter is based on reference [16]. Some changes have been made in the tests for applicability of the type (c) C-functions. The concept of allowable zeros is original.

2.1 Configuration Space and Configuration Space Obstacles

Consider a rigid object, denoted $\mathcal{A}$, which is free to rotate and translate in three-dimensional space. Let the motion of $\mathcal{A}$ be measured with respect to a reference coordinate system $\{X, Y, Z\}$. The unit vectors along the reference coordinate axes form an orthogonal basis for the three-dimensional space of motion of $\mathcal{A}$. Also define a body coordinate system $\{X_a, Y_a, Z_a\}$ which is rigidly attached to $\mathcal{A}$ and having its origin at the center of rotation of $\mathcal{A}$. The center of rotation of $\mathcal{A}$ will be denoted C^a.

Both the position of C^a in the reference coordinate system and the orientation of the body coordinate axes of $\mathcal{A}$ with respect to the reference coordinate axes must be known to completely describe the spatial arrangement of the points in the rigid body $\mathcal{A}$. The position of $\mathcal{A}$ will be represented by the vector $\mathbf{x}$ pointing from the origin of the reference coordinate system to C^a. Three common representations for the orientation of $\mathcal{A}$ are Euler angles, transformation matrices, and quaternions. A specific choice among these representations will not be made at this point. Instead, $\vec{\Theta}$ will be used as a generic symbol denoting the orientation of $\mathcal{A}$. An indication that some

quantity depends on $\vec{\Theta}$ means that the value of that quantity as expressed in reference coordinates changes with the orientation of $\mathcal{A}$.

A given set of values for $\{\mathbf{x}, \vec{\Theta}\}$ is called a *configuration* of $\mathcal{A}$. The space of all possible values of $\mathbf{x}$ and $\vec{\Theta}$ is the six-dimensional *configuration space*, denoted $C_{space}(\mathcal{A})$. $C_{space}(\mathcal{A})$ is the Cartesian product of the translation space and the rotation space of $\mathcal{A}$. The translation space is the space of all possible positions of C^a, that is all possible values of the vector $\mathbf{x}$. This is simply the familiar three-dimensional Euclidean space $\Re^3$. The rotation space is the space of all possible orientations of $\mathcal{A}$ expressed as possible values of $\vec{\Theta}$ once a particular representation is chosen. This space is the three-dimensional special orthogonal group denoted $SO(3)$. $C_{space}(\mathcal{A})$ can thus be written as $\Re^3 \times SO(3)$. [18]

Now consider another rigid object, denoted $\mathcal{B}$, in the same three-dimensional space as the motion of $\mathcal{A}$ which has fixed position and orientation relative to the reference coordinates. $\mathcal{B}$ is called an obstacle because it restricts the motion of $\mathcal{A}$. The presence of $\mathcal{B}$ implies that there are forbidden configurations of $\mathcal{A}$ which correspond to contact of the surfaces of $\mathcal{A}$ and $\mathcal{B}$ or overlap of their interiors.

The *configuration space obstacle* for $\mathcal{A}$ due to $\mathcal{B}$ is the set of all forbidden configurations of $\mathcal{A}$ and is denoted $\mathbf{CO}_{\mathcal{B}}^{\mathcal{A}}$. The boundary of $\mathbf{CO}_{\mathcal{B}}^{\mathcal{A}}$ is the set of configurations where $\mathcal{A}$ and $\mathcal{B}$ have points in common only on their surfaces. $\mathcal{A}$ and $\mathcal{B}$ are said to be just touching at these configurations because their interiors are disjoint. For configurations of $\mathcal{A}$ inside $\mathbf{CO}_{\mathcal{B}}^{\mathcal{A}}$, the interiors of $\mathcal{A}$ and $\mathcal{B}$ overlap. The set of configurations of $\mathcal{A}$ outside of $\mathbf{CO}_{\mathcal{B}}^{\mathcal{A}}$ is called *free space*.

The motion of $\mathcal{A}$ as it translates and rotates in the original three-dimensional space is equivalent to the trajectory of a single point in $C_{space}(\mathcal{A})$. The original three-dimensional obstacle $\mathcal{B}$ is correspondingly "grown" to the six-dimensional obstacle $\mathbf{CO}_{\mathcal{B}}^{\mathcal{A}}$ in $C_{space}(\mathcal{A})$. The collision detection problem can thus be viewed as the process of determining if and when a given trajectory of the configuration of $\mathcal{A}$ starting in free space first intersects the boundary of $\mathbf{CO}_{\mathcal{B}}^{\mathcal{A}}$. This will be the first point where the surface of the moving object comes into contact with the surface of the obstacle.

A discussion of the properties of $\mathbf{CO}_{\mathcal{B}}^{\mathcal{A}}$ clearly depends on the choice of the shapes for $\mathcal{A}$ and $\mathcal{B}$. For this thesis, both $\mathcal{A}$ and $\mathcal{B}$ are assumed to be polyhedra. The characterization of $\mathbf{CO}_{\mathcal{B}}^{\mathcal{A}}$ presented in next sections also

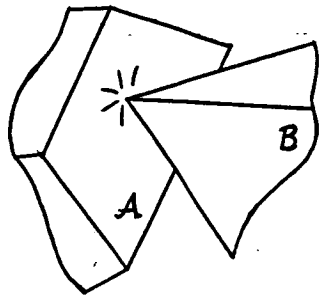
requires that $\mathcal{A}$ and $\mathcal{B}$ be convex. Section 2.2 discusses the possible types of contacts between the features of the two polyhedra and introduces the C-functions that model these contacts. A characterization of $\mathbf{CO}_{\mathcal{B}}^{\mathcal{A}}$ in terms of the C-functions is given in section 2.3. An extension of this characterization to include the case where either $\mathcal{A}$ or $\mathcal{B}$ is a nonconvex polyhedron will be discussed in section 2.8.

2.2 Contacts for Convex Polyhedra

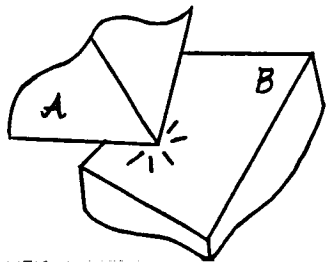
If $\mathcal{A}$ and $\mathcal{B}$ are both convex polyhedra, the boundary of $\mathbf{CO}_{\mathcal{B}}^{\mathcal{A}}$ is determined by three types of contact between the vertices, edges, and faces of $\mathcal{A}$ and $\mathcal{B}$. A type (a) contact occurs when a face of $\mathcal{A}$ touches a vertex of $\mathcal{B}$, a type (b) contact occurs when a vertex of $\mathcal{A}$ touches a face of $\mathcal{B}$, and a type (c) contact occurs when an edge of $\mathcal{A}$ touches an edge of $\mathcal{B}$. Consider a configuration of $\mathcal{A}$ where it is just touching $\mathcal{B}$ such that the intersection of their surfaces is a single point. If the only contact at this configuration is a single type (a) contact, then one vertex of $\mathcal{B}$ must be touching the interior of the polygon representing one of the faces of $\mathcal{A}$. Analogously, if the only contact at this configuration is a single type (b) contact, then a vertex of $\mathcal{A}$ must be touching the interior of the polygon representing one of the faces of $\mathcal{B}$. If there is a single type (c) contact and no type (a) or (b) contacts at this configuration, then the intersection point of the surfaces of $\mathcal{A}$ and $\mathcal{B}$ must be the point of intersection between an edge of $\mathcal{A}$ and an edge $\mathcal{B}$ that is not an endpoint of one of the two edges. The three configurations in Figure 2.1 illustrate single type (a), (b), and (c) contacts in the original three-dimensional space of the motion of $\mathcal{A}$.

Multiple type (a), (b), or (c) contacts occur when a pair of features of $\mathcal{A}$ and $\mathcal{B}$ are touching other than the three pairs which define type (a), (b), and (c) contacts. As an example, suppose that a face of $\mathcal{A}$ touches a face of $\mathcal{B}$. Any vertices on the face of $\mathcal{B}$ which are touching the face of $\mathcal{A}$ correspond to type (a) contacts. Any vertices on the face of $\mathcal{A}$ which are touching the face of $\mathcal{B}$ correspond to type (b) contacts. Any points where an edge bounding the face of $\mathcal{A}$ intersects an edge bounding the face of $\mathcal{B}$ correspond to type (c) contacts. Figure 2.2 shows the type (a), (b), and (c) contacts for several different cases where a face of $\mathcal{A}$ touches a face of $\mathcal{B}$.

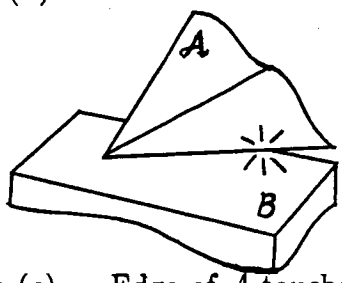
A more subtle case of multiple type (a), (b), and (c) contacts occurs



Type (a) — Vertex of $\mathcal{B}$ touches face of $\mathcal{A}$

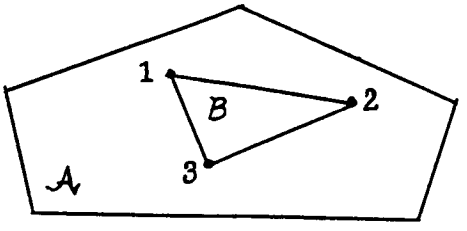


Type (b) — Vertex of $\mathcal{A}$ touches face of $\mathcal{B}$

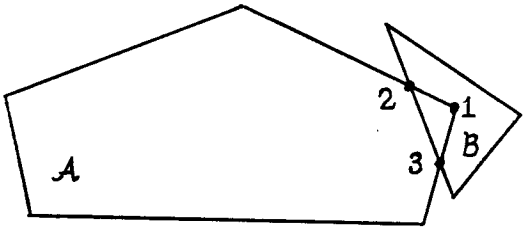


Type (c) — Edge of $\mathcal{A}$ touches edge of $\mathcal{B}$

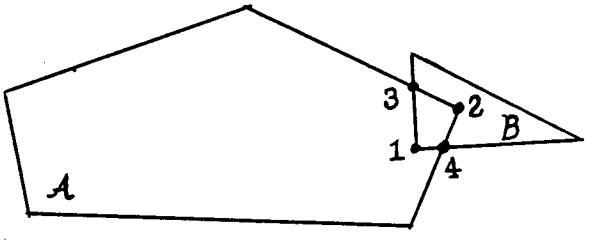
Figure 2.1: The three types of contact for polyhedra



1, 2, 3 – Type (a) contacts



1 – Type (b) contact
2, 3 – Type (c) contacts



1 – Type (a) contact
2 – Type (b) contact
3, 4 – Type (c) contacts

Figure 2.2: Multiple type (a), (b), and (c) contacts for overlapping faces of $\mathcal{A}$ and $\mathcal{B}$

for configurations of $\mathcal{A}$ where one of its edges is aligned with an edge of $\mathcal{B}$. The term "aligned" means that the lines containing the two edges are parallel. Suppose a type (c) contact occurs between two aligned edges of $\mathcal{A}$ and $\mathcal{B}$ such that the only points in common between $\mathcal{A}$ and $\mathcal{B}$ are the points which lie on both of these edges. The two edges can overlap either at a common endpoint, for a partial segment of their length, or one edge can be completely contained in the other. In either of the first two cases, one of the vertices at the endpoints of the edge of $\mathcal{A}$ is touching the edge of $\mathcal{B}$ and one of the vertices at the endpoints of the edge of $\mathcal{B}$ is touching the edge of $\mathcal{A}$. Since there are two faces of each polyhedron which meet at each of the aligned edges, any vertex of $\mathcal{A}$ which touches the edge of $\mathcal{B}$ also touches two of the faces of $\mathcal{B}$. Any vertex of $\mathcal{B}$ which touches the edge of $\mathcal{A}$ is touching two of the faces of $\mathcal{A}$. This means that there are two type (a) and two type (b) contacts for these two configurations. In the third case where one edge completely overlaps the other edge, there are either four type (a) or four type (b) contacts. If the edge of $\mathcal{A}$ is the larger edge, then both vertices at the endpoints of the edge of $\mathcal{B}$ are contained in the edge of $\mathcal{A}$. Therefore, both of these vertices touch the two faces that meet at the edge of $\mathcal{A}$ and there are four type (a) contacts at this configuration. On the other hand, if the edge of $\mathcal{B}$ is the larger edge, then both vertices at the endpoints of the edge of $\mathcal{A}$ touch the two faces that meet at the edge of $\mathcal{B}$ and there are four type (b) contacts at this configuration. For the remainder of this section and for the three sections which follow, it will be assumed that the two edges are not aligned whenever type (c) contacts are discussed. The implications of edge-edge alignment will be discussed in detail in section 2.6.

2.2.1 The C-functions for Convex Polyhedra

Each of the three types of contacts described above can be represented by a different set of functions. These functions are called *C-functions* because they model the constraints on the configuration of $\mathcal{A}$ due to the presence of $\mathcal{B}$. There is one such C-function for each pair of features of $\mathcal{A}$ and $\mathcal{B}$ corresponding to type (a), (b), or (c) contacts.

The C-functions are formulated using vectors to represent the the three features of the polyhedra representing $\mathcal{A}$ and $\mathcal{B}$. The vectors representing the faces, edges, and vertices of $\mathcal{A}$ and $\mathcal{B}$ are defined below.

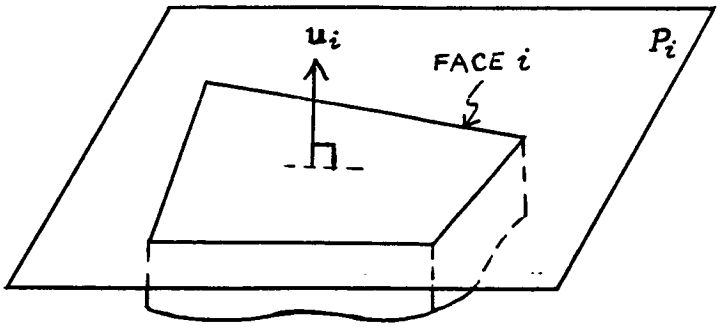
Let the open positive half-space of a plane containing a face of a convex polyhedron be the half-space which *does not* contain any points on or inside of the polyhedron. Then the closed negative half-space is the half-space which consists of the plane itself and the open-half space which *does* contain points on or inside of the polyhedron. Then, the unit, outward-pointing normal vector for a face of a convex polyhedron is defined as that unit vector which is perpendicular to the plane containing that face and which points into the positive half-space of that plane.

An edge vector is defined to be a vector that points along some edge of the polyhedron from one of the vertices which is an endpoint of this edge to the other vertex which is an endpoint. The vertex where an edge vector begins is referred to as its *tail* vertex while the vertex where it terminates is referred to as its *head* vertex. Clearly, there are two possible edge vectors for each edge depending on the choice of which of the two endpoints is the tail vertex and which is the head vertex. These two edge vectors point in opposite directions. For the remainder of this thesis, when the direction of the edge vector is relevant to the discussion its tail vertex will be specified. When the direction of an edge vector is not relevant, the choice of the tail vertex is arbitrary and it will not be explicitly identified.

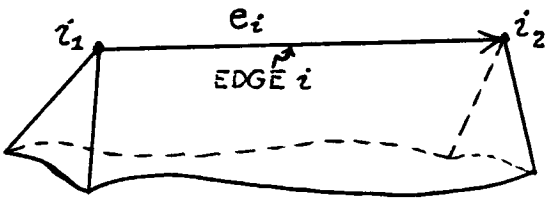
The vector representing a vertex of the polyhedron for the moving object $\mathcal{A}$ is the vector which points from C^a to that vertex. For $\mathcal{B}$, the vector representing a given vertex points from the origin of reference coordinates to that vertex.

These vectors are illustrated in Figure 2.3. The vectors representing features of $\mathcal{A}$ are fixed in the body coordinate system of $\mathcal{A}$, but depend on the orientation of $\mathcal{A}$ when expressed in reference coordinates. The vectors representing features of $\mathcal{B}$ are fixed in reference coordinates. All of these vectors are expressed in reference coordinates when defining the C-functions. The dependence of vectors representing features of $\mathcal{A}$ on $\vec{\Theta}$ indicates that these vectors have been transformed from the body coordinates of $\mathcal{A}$ to reference coordinates.

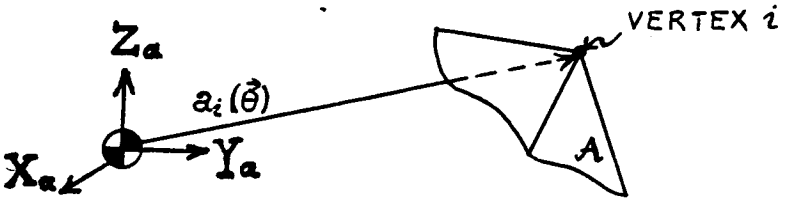
Type (a) C-functions A type (a) C-function models the contact between a face of $\mathcal{A}$ and a vertex of $\mathcal{B}$. Let $\mathbf{u}_i^a\left(\vec{\Theta}\right)$ be the unit, outward-pointing normal vector to the ith face of $\mathcal{A}$ and $\mathbf{a}_i\left(\vec{\Theta}\right)$ be the vector representing any one of the vertices of $\mathcal{A}$ which lies on its ith face . Let $\mathbf{b}_j$ be



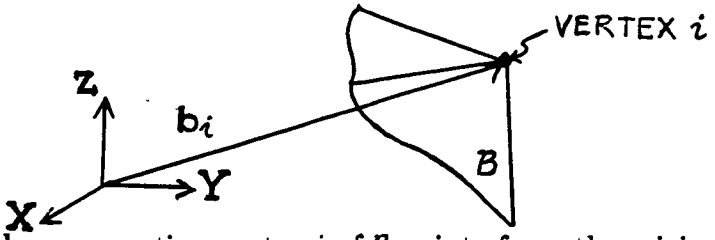
The unit, outward-pointing normal vector $\mathbf{u}_i$ for face i of a convex polyhedron is perpendicular to the plane P_i which contains face i and points into the half-space of P_i which *does not* contain the polyhedron.



An edge vector $\mathbf{e}_i$ for edge i of a polyhedron points along the edge from its tail vertex i_1 to its head vertex i_2.



The vector $\mathbf{a}_i\left(\vec{\Theta}\right)$ representing vertex i of $\mathcal{A}$ points from C^a, the center of rotation of $\mathcal{A}$, to vertex i.



The vector $\mathbf{b}_i$ representing vertex i of $\mathcal{B}$ points from the origin of reference coordinates to vertex i.

Figure 2.3: The vectors representing vertices, edges, and faces of $\mathcal{A}$ and $\mathcal{B}$

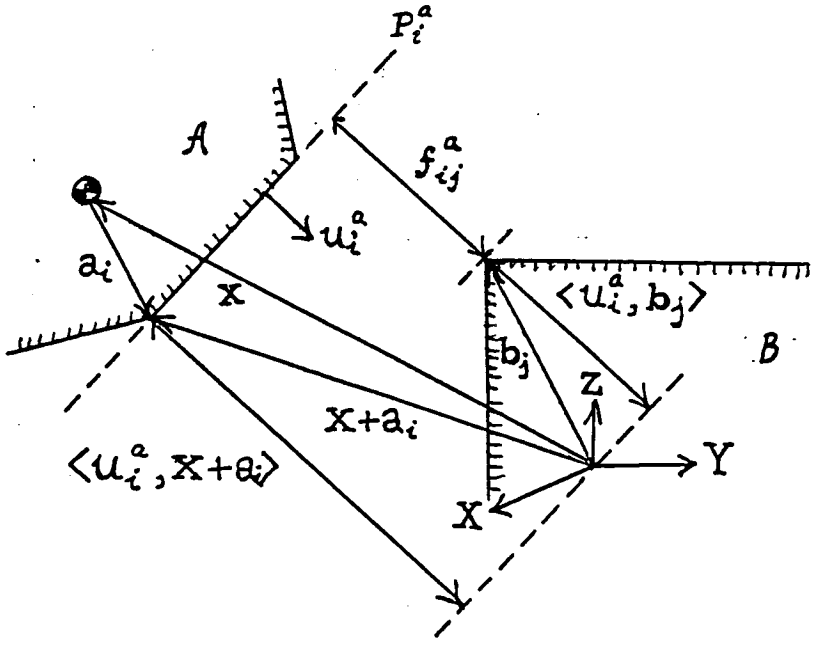


Figure 2.4: The type (a) C-function $f_{i,j}^a$

the vector representing the jth vertex of $\mathcal{B}$. Then the type (a) C-function $f_{i,j}^a$ for face i of $\mathcal{A}$ and vertex j of $\mathcal{B}$ is defined as

$$f_{i,j}^a\left(\mathbf{x}, \vec{\Theta}\right) = \left\langle \mathbf{u}_i^a\left(\vec{\Theta}\right), \mathbf{b}_j \right\rangle - \left\langle \mathbf{u}_i^a\left(\vec{\Theta}\right), \mathbf{x} + \mathbf{a}_i\left(\vec{\Theta}\right) \right\rangle$$

The first inner product is the magnitude of the component of $\mathbf{b}_j$ along $\mathbf{u}_i^a\left(\vec{\Theta}\right)$. The term $\mathbf{x} + \mathbf{a}_i\left(\vec{\Theta}\right)$ is the position of vertex i of $\mathcal{A}$ in the reference coordinate system. Recall that this vertex was chosen to lie on face i of $\mathcal{A}$. Therefore, the second inner product is the distance to the plane containing face i of $\mathcal{A}$ from the origin of the reference coordinate system measured along the normal to this plane. This distance is the same for all points on face i of $\mathcal{A}$. The C-function $f_{i,j}^a$ then measures the distance of vertex j of $\mathcal{B}$ from the plane P_i^a containing face i of $\mathcal{A}$ along its outward-pointing normal vector. Its magnitude is equal to the minimum distance from the vertex of $\mathcal{B}$ to this plane.

Figure 2.4 shows the construction of $f_{i,j}^a$ for a configuration of $\mathcal{A}$ where it has a positive value. The type (a) C-function $f_{i,j}^a$ is positive whenever vertex j of $\mathcal{B}$ lies in the open positive half-space of P_i^a and it is negative whenever vertex j of $\mathcal{B}$ lies in the open negative half-space of P_i^a. When the value of $f_{i,j}^a$ equals zero, vertex j of $\mathcal{B}$ lies in P_i^a.

Type (b) C-functions A type (b) C-function models contact between a vertex of $\mathcal{A}$ and a face of $\mathcal{B}$. Let $\mathbf{a}_i\left(\vec{\Theta}\right)$ be the vector representing the ith vertex of $\mathcal{A}$. Let $\mathbf{u}_j^b$ be the unit, outward-pointing normal vector to the jth face of $\mathcal{B}$ and $\mathbf{b}_j$ be the vector representing any one of the vertices of $\mathcal{B}$ which lies on its jth face. Then the type (b) C-function $f_{i,j}^b$ for vertex i of $\mathcal{A}$ and face j of $\mathcal{B}$ is defined as

$$f_{i,j}^b\left(\mathbf{x}, \vec{\Theta}\right) = \left\langle \mathbf{u}_j^b, \mathbf{x} + \mathbf{a}_i\left(\vec{\Theta}\right)\right\rangle - \left\langle \mathbf{u}_j^b, \mathbf{b}_j\right\rangle$$

The first inner product is the magnitude of the component of the position vector of vertex i of $\mathcal{A}$, $\mathbf{x} + \mathbf{a}_i\left(\vec{\Theta}\right)$, along $\mathbf{u}_j^b$. The second inner product is the distance of the plane P_j^b containing face j of $\mathcal{B}$ from the origin of the reference coordinate system measured along $\mathbf{u}_j^b$. It is constant because $\mathcal{B}$ is stationary with respect to this reference frame. This distance is the same for all points in face j of $\mathcal{B}$. The C-function $f_{i,j}^b$ measures the distance of vertex i of $\mathcal{A}$ from the plane P_j^b along its outward-pointing normal vector. Its magnitude is the minimum distance between the vertex of $\mathcal{A}$ and this plane.

Figure 2.5 shows the construction of $f_{i,j}^b$ for a configuration of $\mathcal{A}$ where it has a positive value. The type (b) C-function $f_{i,j}^b$ is positive whenever vertex i of $\mathcal{A}$ lies in the open positive half-space of P_j^b and it is negative whenever vertex i of $\mathcal{A}$ lies in the open negative half-space of P_j^b. When the value of $f_{i,j}^b$ equals zero, vertex i of $\mathcal{A}$ lies in P_j^b.

Type (c) C-functions A type (c) C-function models contact between an edge of $\mathcal{A}$ and an edge of $\mathcal{B}$. Let $\mathbf{e}_i^a\left(\vec{\Theta}\right)$ be one of the edge vectors for the ith edge of $\mathcal{A}$ and $\mathbf{a}_{i_1}\left(\vec{\Theta}\right)$ be the vector representing the tail vertex of $\mathbf{e}_i^a\left(\vec{\Theta}\right)$. Let $\mathbf{e}_j^b$ be one of the edge vectors for the jth edge of $\mathcal{B}$ and $\mathbf{b}_{j_1}$ be the vector representing the tail vertex of $\mathbf{e}_j^b$. Then the type (c) C-function $f_{i,j}^c$ for edge i of $\mathcal{A}$ and edge j of $\mathcal{B}$ is defined as

$$f_{i,j}^c\left(\mathbf{x}, \vec{\Theta}\right) = \left\langle \mathbf{e}_i^a\left(\vec{\Theta}\right) \times \mathbf{e}_j^b, \mathbf{x} + \mathbf{a}_{i_1}\left(\vec{\Theta}\right)\right\rangle - \left\langle \mathbf{e}_i^a\left(\vec{\Theta}\right) \times \mathbf{e}_j^b, \mathbf{b}_{j_1}\right\rangle$$

Either of the two edge vectors for edge i of $\mathcal{A}$ and edge j of $\mathcal{B}$ can be used in the above definition.

The cross product $\mathbf{e}_i^a\left(\vec{\Theta}\right) \times \mathbf{e}_j^b$, which appears twice in the definition of $f_{i,j}^c$, is the normal vector to the plane P_{ij}^c which contains edge j of $\mathcal{B}$ and

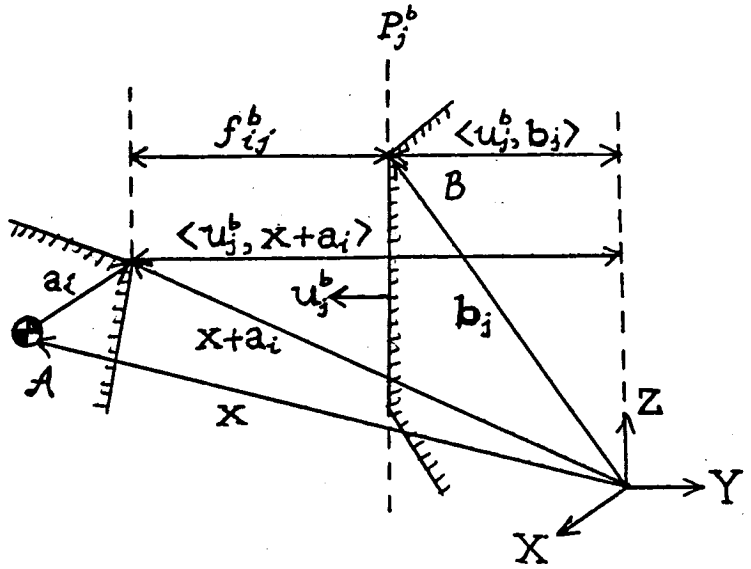


Figure 2.5: The type (b) C-function $f_{i,j}^b$

is parallel to edge i of $\mathcal{A}$. This plane is unique except when the two edges are aligned, as will be discussed in section 2.6. The first inner product is the magnitude of the component of the position vector of the tail vertex of edge i of $\mathcal{A}$, $\mathbf{x} + \mathbf{a}_i\left(\vec{\Theta}\right)$, along this normal vector. This term measures the distance of edge i of $\mathcal{A}$ from the origin of the reference coordinate system along the normal vector to P_{ij}^c. Since P_{ij}^c is parallel to edge i of $\mathcal{A}$, this distance is the same for all points along edge i of $\mathcal{A}$. The second inner product is the distance of the plane P_{ij}^c from the origin of the reference coordinate system measured along the normal vector to this plane. This distance is the same for all points along the jth edge of $\mathcal{B}$ because the jth edge lies in P_{ij}^c. By taking the difference of the two inner products, the C-function $f_{i,j}^c$ measures the distance of edge i of $\mathcal{A}$ from P_{ij}^c along its normal vector. The magnitude of $f_{i,j}^c$ is equal to the minimum distance between edge i of $\mathcal{A}$ and P_{ij}^c, which is equivalent to the minimum distance between the lines containing edge i of $\mathcal{A}$ and edge j of $\mathcal{B}$.

Figure 2.6 shows the construction of $f_{i,j}^c$ for a configuration of $\mathcal{A}$ where it has a positive value. The type (c) C-function $f_{i,j}^c$ is positive whenever edge i of $\mathcal{A}$ lies in the open positive half-space of P_{ij}^c and it is negative whenever the vertex of $\mathcal{A}$ lies in the open negative half-space of P_{ij}^c. When

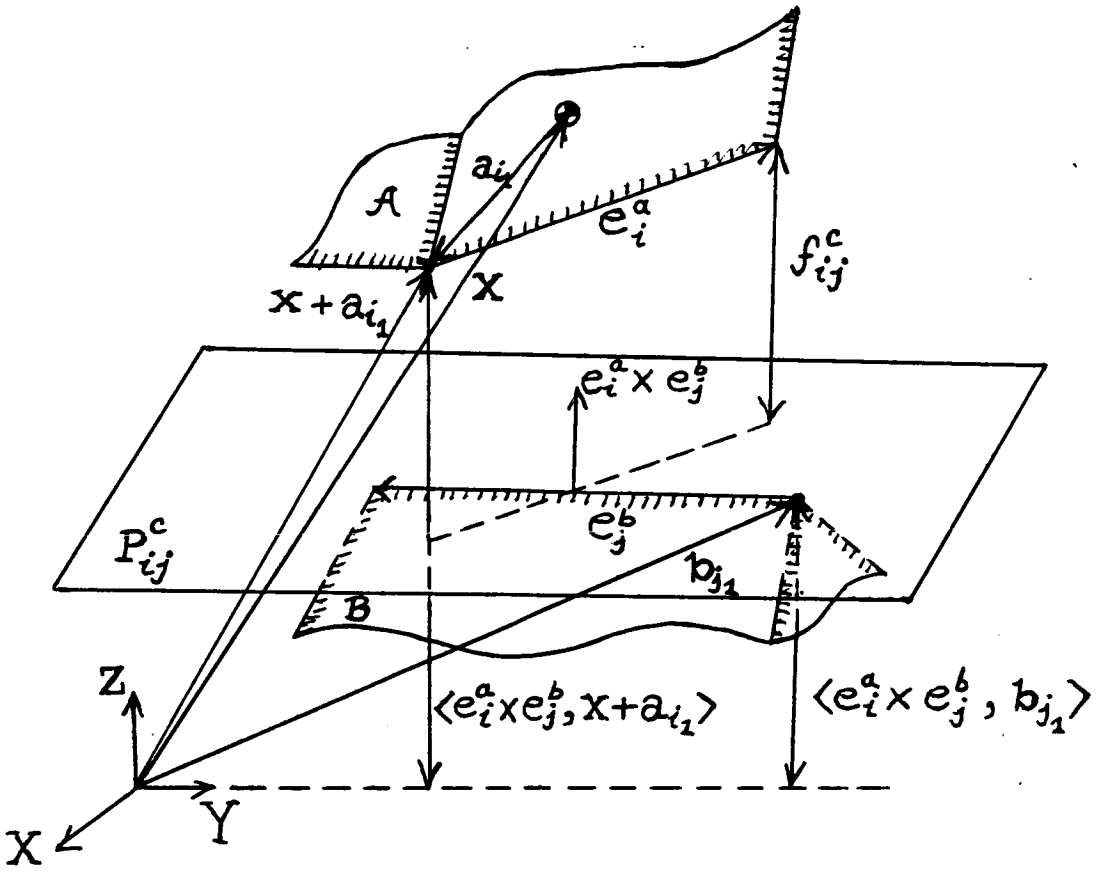


Figure 2.6: The type (c) C-function $f_{i,j}^c$

the value of $f^c_{i,j}$ equals zero, edge i of $\mathcal{A}$ lies in the plane P^c_{ij}. In other words, edge i of $\mathcal{A}$ and edge j of $\mathcal{B}$ are coplanar when $f^c_{i,j}$ is zero.

2.3 The C-functions and Construction of the Configuration Space Obstacle

This section considers the relationship of the C-functions and the configuration space obstacle. The discussion will first be restricted to the case where $\mathcal{A}$ is not rotating so that the orientation of $\mathcal{A}$ is fixed. The projection of $\mathbf{CO}^{\mathcal{A}}_{\mathcal{B}}$ in $C_{space}(\mathcal{A})$ for a fixed orientation of $\mathcal{A}$ is described. This projection is an object in the three-dimensional translation space of $\mathcal{A}$ and will be denoted $\mathbf{CO}^{\mathcal{A}}_{\mathcal{B}}(\mathbf{x})$. Next, the C-surfaces defined by the C-functions are introduced and the concepts of applicability and orientation of a C-function are defined in order to describe the construction of $\mathbf{CO}^{\mathcal{A}}_{\mathcal{B}}(\mathbf{x})$ using these C-surfaces. This construction leads to the formulation of conditions that determine whether a particular value of $\mathbf{x}$ is in free space, on the boundary of $\mathbf{CO}^{\mathcal{A}}_{\mathcal{B}}(\mathbf{x})$, or inside of $\mathbf{CO}^{\mathcal{A}}_{\mathcal{B}}(\mathbf{x})$. Then, the concepts of C-surfaces and applicability and orientation of a C-function are generalized to include the case where the values of both $\mathbf{x}$ and $\vec{\Theta}$ are changing. The construction of the full six-dimensional configuration space obstacle $\mathbf{CO}^{\mathcal{A}}_{\mathcal{B}}$ from the C-surfaces is described. Finally, three "collision" conditions are given that determine when a particular configuration of $\mathcal{A}$, $\{\mathbf{x}, \vec{\Theta}\}$, lies on the boundary of $\mathbf{CO}^{\mathcal{A}}_{\mathcal{B}}$.

2.3.1 The Configuration Space Obstacle for Fixed Orientation of $\mathcal{A}$

As a simple example of the process by which the three types of contacts produce surfaces of $\mathbf{CO}^{\mathcal{A}}_{\mathcal{B}}$ in $C_{space}(\mathcal{A})$, consider the projection of $\mathbf{CO}^{\mathcal{A}}_{\mathcal{B}}$ for a fixed orientation of $\mathcal{A}$, $\mathbf{CO}^{\mathcal{A}}_{\mathcal{B}}(\mathbf{x})$. The part of $\mathbf{CO}^{\mathcal{A}}_{\mathcal{B}}(\mathbf{x})$ which is due to a particular type (a) contact is formed by sliding a face of $\mathcal{A}$ along a vertex of $\mathcal{B}$ with $\mathcal{A}$ in the appropriate fixed orientation. The face of $\mathcal{A}$ must be chosen such that the interiors of $\mathcal{A}$ and $\mathcal{B}$ remain disjoint as the face slides along the vertex of $\mathcal{B}$. The boundary of the part of $\mathbf{CO}^{\mathcal{A}}_{\mathcal{B}}(\mathbf{x})$ due to this type (a) contact is generated as the edges bounding the face of $\mathcal{A}$ slide along the vertex of $\mathcal{B}$. C^a remains within a polygon which is simply the face of $\mathcal{A}$ rotated to the given fixed orientation of $\mathcal{A}$ and displaced by an

appropriate distance to maintain contact between the face and the vertex of $\mathcal{B}$.

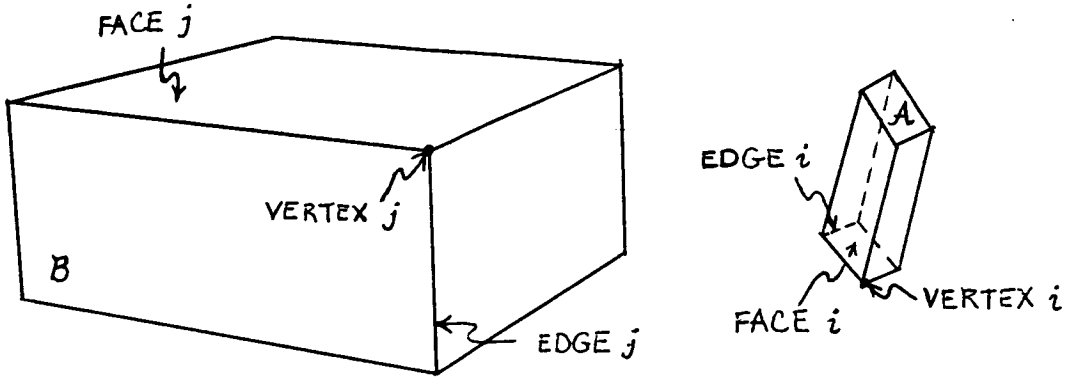
Similarly, the part of $\mathbf{CO}_{\mathcal{B}}^{\mathcal{A}}(\mathbf{x})$ which is due to a particular type (b) contact is formed by sliding a vertex of $\mathcal{A}$ along a face of $\mathcal{B}$ with $\mathcal{A}$ in the appropriate fixed orientation. In this case, the vertex of $\mathcal{A}$ must be chosen such that the interiors of $\mathcal{A}$ and $\mathcal{B}$ remain disjoint as the vertex slides along the face of $\mathcal{B}$. The boundary of the part of $\mathbf{CO}_{\mathcal{B}}^{\mathcal{A}}(\mathbf{x})$ due to this type (b) contact is generated as the vertex of $\mathcal{A}$ slides along the edges bounding the face of $\mathcal{B}$. In this case, C^a remains within a polygon which is simply the face of $\mathcal{B}$ displaced by an appropriate distance to maintain contact between the face and the vertex of $\mathcal{A}$.

The part of $\mathbf{CO}_{\mathcal{B}}^{\mathcal{A}}(\mathbf{x})$ which is due to a particular type (c) contact is formed by sliding an edge of $\mathcal{A}$ along an edge of $\mathcal{B}$ with $\mathcal{A}$ in the appropriate fixed orientation. The edge of $\mathcal{A}$ must be chosen so that the interiors of $\mathcal{A}$ and $\mathcal{B}$ remain disjoint as it slides along the edge of $\mathcal{B}$. Two parallel edges are formed as part of the boundary of $\mathbf{CO}_{\mathcal{B}}^{\mathcal{A}}(\mathbf{x})$ by sliding the edge of $\mathcal{A}$ along the two vertices which are the endpoints of the edge of $\mathcal{B}$. The other two parallel edges which complete the boundary of this part of $\mathbf{CO}_{\mathcal{B}}^{\mathcal{A}}(\mathbf{x})$ are formed by sliding the vertices which are the endpoints of the edge of $\mathcal{A}$ along the edge of $\mathcal{B}$. C^a remains within the parallelogram formed by these two sets of parallel edges. This parallelogram lies in a plane which is parallel to the plane which contains the edges of $\mathcal{A}$ and $\mathcal{B}$. One set of edges of the parallelogram comes from displacing the edge of $\mathcal{A}$ by the two appropriate distances which maintain contact between the edge and the endpoints of the edge of $\mathcal{B}$. The other set of edges comes from displacing the edge of $\mathcal{B}$ by the two appropriate distances which maintain contact between the edge and the endpoints of the edge of $\mathcal{A}$.

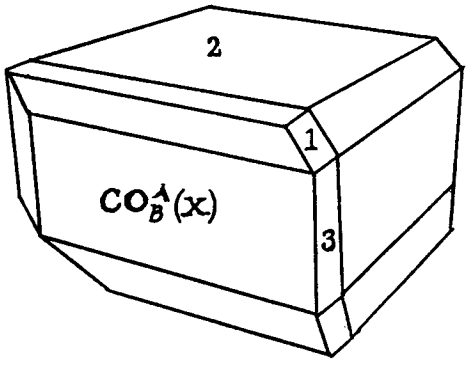
Thus, for a fixed orientation of $\mathcal{A}$, the projection of $\mathbf{CO}_{\mathcal{B}}^{\mathcal{A}}$ onto the three-dimensional translation space is a convex polyhedron in the translation space of $\mathcal{A}$. It is made up of displaced faces of $\mathcal{A}$ and $\mathcal{B}$ due to type (a) and (b) contacts which are connected by parallelograms due to type (c) contacts.

Figure 2.7 shows the complete convex polyhedron $\mathbf{CO}_{\mathcal{B}}^{\mathcal{A}}(\mathbf{x})$ for two rectangular boxes. One face due to each of the three types of contact is indicated along with the features of $\mathcal{A}$ and $\mathcal{B}$ which generated it.

Since $\mathbf{CO}_{\mathcal{B}}^{\mathcal{A}}(\mathbf{x})$ is a convex polyhedron, it is easy to test whether a particular value of $\mathbf{x}$ lies outside, on, or inside it. Convexity guarantees that



Moving object $\mathcal{A}$ and obstacle $\mathcal{B}$



1. Type (a) face generated by sliding face i of $\mathcal{A}$ along vertex j of $\mathcal{B}$
2. Type (b) face generated by sliding vertex i of $\mathcal{A}$ along face j of $\mathcal{B}$
3. Type (c) face generated by sliding edge i of $\mathcal{A}$ along edge j of $\mathcal{B}$

Figure 2.7: The convex polyhedron $\mathbf{CO}_{\mathcal{B}}^{\mathcal{A}}(\mathbf{x})$ for two rectangular boxes

the entire polyhedron lies in the closed negative half-space for any one of the planes containing a face of the polyhedron. Also, a point which is either on the surface of the polyhedron or in its interior lies in the closed negative half-spaces of *all* of the planes containing a face of the polyhedron. Thus, a particular value of $\mathbf{x}$ lies on or inside of $\mathbf{CO}_{\mathcal{B}}^{\mathcal{A}}(\mathbf{x})$ if it lies in the closed negative half-space of each of the planes containing a face of $\mathbf{CO}_{\mathcal{B}}^{\mathcal{A}}(\mathbf{x})$. A particular value of $\mathbf{x}$ lies on the boundary of $\mathbf{CO}_{\mathcal{B}}^{\mathcal{A}}(\mathbf{x})$ if it lies in the plane of at least one of the faces of $\mathbf{CO}_{\mathcal{B}}^{\mathcal{A}}(\mathbf{x})$ and in the open negative half-space of the planes of the remaining faces of $\mathbf{CO}_{\mathcal{B}}^{\mathcal{A}}(\mathbf{x})$.

The next four sections describe how to identify the planes containing the faces of $\mathbf{CO}_{\mathcal{B}}^{\mathcal{A}}(\mathbf{x})$ with an appropriate set of C-functions and how to use the values of these C-functions to test the location of $\mathbf{x}$ relative to $\mathbf{CO}_{\mathcal{B}}^{\mathcal{A}}(\mathbf{x})$.

2.3.1.1 C-surfaces

Consider the possible translations of C^a for a fixed orientation of $\mathcal{A}$ that result in constant values for each of the three types of C-functions. The type (a) C-function $f_{i,j}^a$ measures the perpendicular distance of vertex j of $\mathcal{B}$ from the plane P_i^a containing face i of $\mathcal{A}$. Therefore, C^a must translate in a direction parallel to P_i^a if the value of $f_{i,j}^a$ is to remain constant. Similarly, C^a must translate in a direction parallel to the plane P_j^b containing face j of $\mathcal{B}$ if the value of $f_{i,j}^b$ is to remain constant. This is because $f_{i,j}^b$ measures the perpendicular distance between vertex i of $\mathcal{A}$ and P_j^b. When edge i of $\mathcal{A}$ and edge j of $\mathcal{B}$ are not aligned, the type (c) C-function $f_{i,j}^c$ measures the perpendicular distance of edge i of $\mathcal{A}$ from plane P_{ij}^c that contains edge j of $\mathcal{B}$ and is parallel to edge i of $\mathcal{A}$. The value of $f_{i,j}^c$ will remain constant only if C^a translates in a direction parallel to P_{ij}^c.

Now, the normal vectors to planes P_i^a, P_j^b and P_{ij}^c are fixed for a fixed orientation of $\mathcal{A}$. In this case, when C^a translates in a direction parallel to one of these planes, the position vector $\mathbf{x}$ must terminate in a plane that is also parallel to one of these planes. Thus, the surfaces where a C-function has constant value are two-dimensional planes in the three-dimensional translation space of $\mathcal{A}$. The plane containing all the values of $\mathbf{x}$ where the value of a C-function is zero is the *C-surface* defined by that C-function.

Recall the description of the faces of $\mathbf{CO}_{\mathcal{B}}^{\mathcal{A}}(\mathbf{x})$ given above. A type (a) face is generated by translating $\mathcal{A}$ such that one of its faces remains in contact with one of the vertices of $\mathcal{B}$. For this motion, C^a is translating

parallel to the plane containing the face of $\mathcal{A}$ and the type (a) C-function defined by the face of $\mathcal{A}$ and the vertex of $\mathcal{B}$ that are in contact must be zero. Identical arguments show that the value of a type (b) or (c) C-function whose defining features are in contact when a type (b) or (c) face of $\mathbf{CO}_{\mathcal{B}}^{\mathcal{A}}(\mathbf{x})$ is generated must also be zero. Therefore, the faces of $\mathbf{CO}_{\mathcal{B}}^{\mathcal{A}}(\mathbf{x})$ are contained in planes that are C-surfaces for *some* of the C-functions defined by the features of $\mathcal{A}$ and $\mathcal{B}$.

The next sections will describe how to distinguish those C-functions whose C-surfaces generate the faces of $\mathbf{CO}_{\mathcal{B}}^{\mathcal{A}}(\mathbf{x})$ from the set of all C-functions defined by the features of $\mathcal{A}$ and $\mathcal{B}$.

2.3.1.2 Applicability of a C-function

By definition, the boundary of $\mathbf{CO}_{\mathcal{B}}^{\mathcal{A}}(\mathbf{x})$ consists of those values of $\mathbf{x}$ where the surface of $\mathcal{A}$ touches the surface of $\mathcal{B}$ such that their interiors remain disjoint. For any fixed orientation of $\mathcal{A}$, only a small subset of the possible C-functions defined by the features of $\mathcal{A}$ and $\mathcal{B}$ represent contacts where the interiors of $\mathcal{A}$ and $\mathcal{B}$ do not overlap. The C-function defined by two particular features of $\mathcal{A}$ and $\mathcal{B}$ is said to be *applicable* for orientation $\vec{\Theta}$ if a pure translation of $\mathcal{A}$ with fixed orientation $\vec{\Theta}$ can bring the feature of $\mathcal{A}$ into contact with the feature of $\mathcal{B}$ without overlap of their interiors. Because this definition allows arbitrary translations of $\mathcal{A}$, the applicability of a C-function depends only on the orientation of $\mathcal{A}$. It is the contacts of features in the subset of applicable C-functions that generate the faces of $\mathbf{CO}_{\mathcal{B}}^{\mathcal{A}}(\mathbf{x})$.

The applicability of a C-function for a given value of $\vec{\Theta}$ can be tested by assuming that $\mathcal{A}$ has been translated with the fixed orientation $\vec{\Theta}$ until the features defining the C-function are in contact and asking if the interiors of $\mathcal{A}$ and $\mathcal{B}$ are disjoint after this translation. This is equivalent to asking if there exists a plane containing the features defining the C-function which just separates $\mathcal{A}$ and $\mathcal{B}$. This separating plane must divide the three-dimensional space into two closed half-spaces, one of which contains $\mathcal{A}$ and the other of which contains $\mathcal{B}$. The only points where the surfaces of $\mathcal{A}$ and $\mathcal{B}$ can be in contact lie in the separating plane itself. If a separting plane can be found, the C-function is applicable for the given orientation $\vec{\Theta}$. If no separating plane exists when the two features are brought into contact, then the interiors of $\mathcal{A}$ and $\mathcal{B}$ must overlap and the C-function is

not applicable for the given orientation $\vec{\Theta}$.

The plane that separates $\mathcal{A}$ and $\mathcal{B}$ when the features defining an applicable C-function are in contact depends on the type of the C-function. When the type (a) C-function $f^a_{i,j}$ is applicable, $\mathcal{A}$ can be translated so that its ith face touches the jth vertex of $\mathcal{B}$ without overlap of the interiors of $\mathcal{A}$ and $\mathcal{B}$. The plane P^a_i containing face i of $\mathcal{A}$ then separates $\mathcal{A}$ and $\mathcal{B}$. Similarly, $\mathcal{A}$ can be translated so that its ith vertex touches the jth face of $\mathcal{B}$ without overlap of the interiors of $\mathcal{A}$ and $\mathcal{B}$ when the type (b) C-function $f^b_{i,j}$ is applicable. The plane P^b_j containing face j of $\mathcal{B}$ then separates $\mathcal{A}$ and $\mathcal{B}$. When $f^c_{i,j}$ is applicable and edge i of $\mathcal{A}$ is not parallel to edge j of $\mathcal{B}$, $\mathcal{A}$ can be translated such that edge i of $\mathcal{A}$ touches edge j of $\mathcal{B}$ without overlap of the interiors of $\mathcal{A}$ and $\mathcal{B}$. In this case, the plane P^c_{ij} containing edge j of $\mathcal{B}$ and parallel to edge i of $\mathcal{A}$ separates $\mathcal{A}$ and $\mathcal{B}$.

In general, the individual C-functions may take on any value when $\mathcal{A}$ and $\mathcal{B}$ are overlapping. The value of one of these C-functions contains no information about any contact or overlap of $\mathcal{A}$ and $\mathcal{B}$ unless it is known to be applicable for the given orientation of $\mathcal{A}$. For a given orientation of $\mathcal{A}$ where the C-function is applicable, its value will determine in which half-space of the C-surface containing the corresponding face of $\mathbf{CO}^{\mathcal{A}}_{\mathcal{B}}(\mathbf{x})$ a particular value of $\mathbf{x}$ lies. The next section specifies which half-space corresponds to positive values of the C-function and which half-space corresponds to negative values of the C-function. This must be known to formulate rules that determine the location of $\mathbf{x}$ relative to $\mathbf{CO}^{\mathcal{A}}_{\mathcal{B}}(\mathbf{x})$ using the values of the C-functions.

2.3.1.3 Orientation of a C-function

When the type (a) C-function $f^a_{i,j}$ is applicable, one of the faces of $\mathbf{CO}^{\mathcal{A}}_{\mathcal{B}}(\mathbf{x})$ is a displacement of face i of $\mathcal{A}$ and lies in the C-surface defined by $f^a_{i,j}$. This C-surface is a plane parallel to the plane P^a_i containing face i of $\mathcal{A}$. The outward-pointing normal vector for face i of $\mathcal{A}$ is the inward-pointing normal vector for the corresponding face of $\mathbf{CO}^{\mathcal{A}}_{\mathcal{B}}(\mathbf{x})$.

When the type (b) C-function $f^b_{i,j}$ is applicable, one of the faces of $\mathbf{CO}^{\mathcal{A}}_{\mathcal{B}}(\mathbf{x})$ is a displacement of face j of $\mathcal{B}$ and lies in the C-surface for $f^b_{i,j}$. This C-surface is a plane parallel to the plane P^b_j containing face j of $\mathcal{B}$. The outward-pointing normal vector for face j of $\mathcal{B}$ is also the outward-pointing normal for the corresponding face of $\mathbf{CO}^{\mathcal{A}}_{\mathcal{B}}(\mathbf{x})$.

Because of the order of the inner products in the definitions of $f^a_{i,j}$ and $f^b_{i,j}$, a positive value for either of these C-functions indicates that the position $\mathbf{x}$ lies in the open positive half-space of the corresponding C-surface. In other words, a positive value indicates that $\mathbf{x}$ lies in the half-space of the C-surface that does not contain $\mathbf{CO}^{\mathcal{A}}_{\mathcal{B}}(\mathbf{x})$. A nonpositive value indicates that $\mathbf{x}$ lies in the closed negative half-space of the corresponding C-surface, which is the half-space that does contain $\mathbf{CO}^{\mathcal{A}}_{\mathcal{B}}(\mathbf{x})$. If any of the applicable type (a) or (b) C-functions have a positive value at $\mathbf{x}$, then $\mathbf{x}$ lies outside of $\mathbf{CO}^{\mathcal{A}}_{\mathcal{B}}(\mathbf{x})$. This is a direct consequence of the fact that the outward-pointing normal vectors to the faces of $\mathcal{A}$ and $\mathcal{B}$ were used in the definition of the type (a) and (b) C-functions.

When the type (c) C-function $f^c_{i,j}$ is applicable, one of the faces of $\mathbf{CO}^{\mathcal{A}}_{\mathcal{B}}(\mathbf{x})$ is a parallelogram that lies in the C-surface defined by $f^c_{i,j}$. This C-surface is a plane parallel to the plane P^c_{ij}. However, the normal vector to plane P^c_{ij} used in the definition of $f^c_{i,j}$ was arbitrarily chosen to be $\mathbf{e}^a_i\left(\vec{\Theta}\right) \times \mathbf{e}^b_j$ and it may point into either of the half-spaces defined by P^c_{ij}. In this case, $\mathbf{e}^a_i\left(\vec{\Theta}\right) \times \mathbf{e}^b_j$ may be either the outward-pointing or the inward-pointing normal vector for the corresponding face of $\mathbf{CO}^{\mathcal{A}}_{\mathcal{B}}(\mathbf{x})$. The direction of $\mathbf{e}^a_i\left(\vec{\Theta}\right) \times \mathbf{e}^b_j$ must be determined before the value of $f^c_{i,j}$ can be used to distinguish whether $\mathbf{x}$ lies outside or inside of $\mathbf{CO}^{\mathcal{A}}_{\mathcal{B}}(\mathbf{x})$. This is called *orienting* the type (c) C-function $f^c_{i,j}$.

For an orientation of $\mathcal{A}$ where $f^c_{i,j}$ is applicable and where $\mathbf{e}^a_i\left(\vec{\Theta}\right) \times \mathbf{e}^b_j$ points into the half-space of P^c_{ij} which *does not* contain $\mathcal{B}$, $\mathbf{e}^a_i\left(\vec{\Theta}\right) \times \mathbf{e}^b_j$ is the outward-pointing normal to the corresponding face of $\mathbf{CO}^{\mathcal{A}}_{\mathcal{B}}(\mathbf{x})$. A *positive* value of $f^c_{i,j}$ indicates that $\mathbf{x}$ lies in the half-space of the C-surface for $f^c_{i,j}$ that does not contain $\mathbf{CO}^{\mathcal{A}}_{\mathcal{B}}(\mathbf{x})$ and thus $\mathbf{x}$ lies outside of $\mathbf{CO}^{\mathcal{A}}_{\mathcal{B}}(\mathbf{x})$. The type (c) C-function is said to have *positive orientation* for this value of $\vec{\Theta}$. Figure 2.6 shows an applicable type (c) C-function with positive orientation. Note that all type (a) and (b) C-functions can be said to have positive orientation for any value of $\vec{\Theta}$ where they are applicable.

For an orientation of $\mathcal{A}$ where $f^c_{i,j}$ is applicable and where $\mathbf{e}^a_i\left(\vec{\Theta}\right) \times \mathbf{e}^b_j$ points into the half-space of P^c_{ij} which *does* contain $\mathcal{B}$, $\mathbf{e}^a_i\left(\vec{\Theta}\right) \times \mathbf{e}^b_j$ is the inward-pointing normal to the corresponding face of $\mathbf{CO}^{\mathcal{A}}_{\mathcal{B}}(\mathbf{x})$. A positive value of $f^c_{i,j}$ indicates that $\mathbf{x}$ lies in the half-space of the C-surface for $f^c_{i,j}$

that does contain $\mathbf{CO}_{\mathcal{B}}^{\mathcal{A}}(\mathbf{x})$. A *negative* value of $f_{i,j}^{c}$ indicates that $\mathbf{x}$ lies in the half-space of the C-surface for $f_{i,j}^{c}$ that does not contain $\mathbf{CO}_{\mathcal{B}}^{\mathcal{A}}(\mathbf{x})$. In this case, a negative value of $f_{i,j}^{c}$ indicates that $\mathbf{x}$ lies outside of $\mathbf{CO}_{\mathcal{B}}^{\mathcal{A}}(\mathbf{x})$.The type (c) C-function is said to have *negative orientation* for this value of $\vec{\Theta}$.

2.3.1.4 Constructing $\mathbf{CO}_{\mathcal{B}}^{\mathcal{A}}(\mathbf{x})$ from C-surfaces

The three preceding sections have shown that $\mathbf{CO}_{\mathcal{B}}^{\mathcal{A}}(\mathbf{x})$ can be constructed from the C-surfaces of the set of applicable C-functions at orientation $\vec{\Theta}$. The C-surfaces are planes that divide the three-dimensional translation space into two half-spaces. The closed positive half-space of a C-surface is the half-space where the value of the C-function is greater than or equal to zero and the closed negative half-space of a C-surface is the half-space where the value of the C-function is less than or equal to zero. These closed half-spaces both include the C-surface itself. $\mathbf{CO}_{\mathcal{B}}^{\mathcal{A}}(\mathbf{x})$ is the intersection of the closed negative half-spaces of the C-surfaces defined by applicable C-functions with positive orientation at $\vec{\Theta}$ and the closed positive half-spaces of the C-surfaces defined by applicable C-functions with negative orientation at $\vec{\Theta}$.

The following procedure can be used to determine if any position $\mathbf{x}$ lies on or inside of $\mathbf{CO}_{\mathcal{B}}^{\mathcal{A}}(\mathbf{x})$ for a given value of $\vec{\Theta}$. First, the subset of all applicable C-functions is found from the larger set of all possible C-functions. Then $\mathbf{x}$ will lie on or inside of the convex polyhedron $\mathbf{CO}_{\mathcal{B}}^{\mathcal{A}}(\mathbf{x})$ if

- The value of each applicable type (a) C-function is nonpositive.
- The value of each applicable type (b) C-function is nonpositive.
- The value of each applicable type (c) C-function with positive orientation is nonpositive.
- The value of each applicable type (c) C-function with negative orientation is nonnegative.

Furthermore, at least one of the applicable C-functions must be zero for $\mathbf{x}$ to lie on the surface of $\mathbf{CO}_{\mathcal{B}}^{\mathcal{A}}(\mathbf{x})$. If this is the case, then $\mathcal{A}$ and $\mathcal{B}$ are just touching at $\{\mathbf{x}, \vec{\Theta}\}$. If all of the applicable type (a) and (b) C-functions and the applicable type (c) C-functions with positive orientation

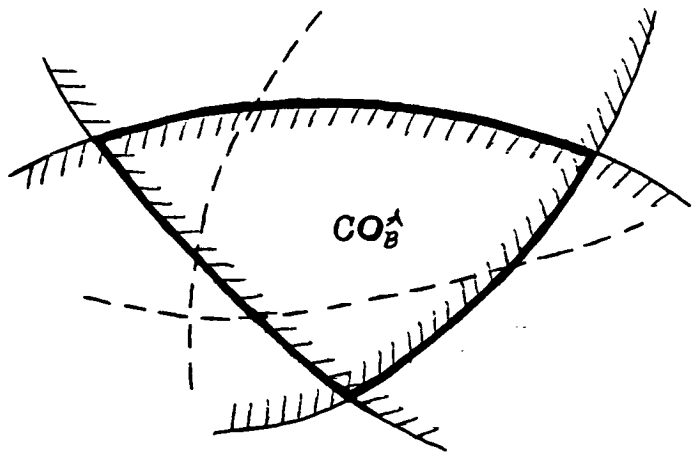
are strictly negative and all of the applicable type (c) C-functions with negative orientation are strictly positive, then the interiors of $\mathcal{A}$ and $\mathcal{B}$ overlap at $\{\mathbf{x}, \vec{\Theta}\}$. If any of the applicable type (a) and (b) C-functions and the applicable type (c) C-functions with positive orientation are positive or any of the applicable type (c) C-functions with negative orientation are negative, then there is no contact of $\mathcal{A}$ and $\mathcal{B}$ at $\{\mathbf{x}, \vec{\Theta}\}$.

2.3.2 The Six-dimensional Configuration Space Obstacle

The discussion thus far has been restricted to describing the structure of $\mathbf{CO}_{\mathcal{B}}^{\mathcal{A}}$ for some fixed orientation of $\mathcal{A}$. The concept of defining $\mathbf{CO}_{\mathcal{B}}^{\mathcal{A}}(\mathbf{x})$ as the intersection of half-spaces determined by planes in the three-dimensional translation space of $\mathcal{A}$ can be extended to characterize the full six-dimensional obstacle $\mathbf{CO}_{\mathcal{B}}^{\mathcal{A}}$. This yields a definition for the boundary of $\mathbf{CO}_{\mathcal{B}}^{\mathcal{A}}$ for the full range of possible motions of $\mathcal{A}$ where both the position and orientation of $\mathcal{A}$ are allowed to vary with time. If $\vec{\Theta}$ is fixed, the values of the C-functions depend only on the translation parameters $\mathbf{x}$, but they are more properly considered to be functions of both the rotation and the translation parameters. The domain of each C-function is the entire $C_{space}(\mathcal{A})$.

When $\vec{\Theta}$ is allowed to vary, there are motions of $\mathcal{A}$ that keep the value of a given C-function constant that involve both rotation and translation. In this case, the points where the value of a C-function is constant define a five-dimensional surface in $C_{space}(\mathcal{A})$. The five-dimensional surface where the value of the C-function is zero is the true C-surface defined by that function. The planes that were referred to as C-surfaces of the C-functions when $\vec{\Theta}$ is fixed are actually projections of the five-dimensional C-surface onto the three-dimensional translation space of $\mathcal{A}$. Each C-surface divides $C_{space}(\mathcal{A})$ into two open half-spaces, one where the value of the C-function is positive and one where the value of the C-function is negative.

As was the case for the construction of $\mathbf{CO}_{\mathcal{B}}^{\mathcal{A}}(\mathbf{x})$, both the applicability and orientation of a C-function must be considered in order to determine which part of its C-surface lies on the boundary of $\mathbf{CO}_{\mathcal{B}}^{\mathcal{A}}$. There can be many values of $\vec{\Theta}$ where the C-function is applicable. The set of all orientations where a C-function is applicable is called the *applicability region*



- - - - C-surface for a C-function which is not applicable
——— C-surface for an applicable C-function
▬▬▬ Boundary of $\mathbf{CO}_B^A$

Shading indicates the open half-space where an applicable C-function with positive orientation is negative or the open half-space where an applicable C-function with negative orientation is positive.

Figure 2.8: Construction of $\mathbf{CO}_B^A$ from C-surfaces

of the C-function. The applicability regions divide the C-functions into those that represent possible contacts of the surfaces of $\mathcal{A}$ and $\mathcal{B}$ and those that represent contacts where the interiors of $\mathcal{A}$ and $\mathcal{B}$ overlap. A positive or negative orientation can be associated with each value of $\vec{\Theta}$ in the applicability region of a C-function.

The configuration space obstacle, $\mathbf{CO}_B^A$, is the intersection of the closed half-spaces where each applicable C-function with positive orientation is nonpositive and each applicable C-function with negative orientation is nonnegative. On the boundary of $\mathbf{CO}_B^A$, at least one applicable C-function must be zero. Therefore, this boundary is composed of "patches" of various C-surfaces. The patch of any particular C-surface which lies on the boundary of $\mathbf{CO}_B^A$ is found by first discarding that part of the surface where the C-function is not applicable. Of the remaining piece, only the part that lies in the closed negative half-spaces of all the other applicable C-functions with positive orientation and the closed positive half-spaces of all the other applicable C-functions with negative orientation is retained. See Figure 2.8.

2.3.3 The Three Collision Conditions

The method of constructing $\mathbf{CO}_{\mathcal{B}}^{\mathcal{A}}$ from the C-surfaces leads to three conditions that must be satisfied for a configuration $\{\mathbf{x}, \vec{\Theta}\}$ to lie on its *boundary*:

1. At least one C-function must be zero for $\{\mathbf{x}, \vec{\Theta}\}$.

2. At least one of the C-functions whose value is zero must be applicable for $\vec{\Theta}$.

3. Any applicable C-function having positive orientation for $\vec{\Theta}$ must be nonpositive and any applicable C-function having negative orientation for $\vec{\Theta}$ must be nonnegative for $\{\mathbf{x}, \vec{\Theta}\}$.

When these conditions are satisfied, $\mathcal{A}$ and $\mathcal{B}$ are touching without overlap. These conditions will be referred to as the "collision" conditions. It is interesting to note that the configuration must lie on or inside of $\mathbf{CO}_{\mathcal{B}}^{\mathcal{A}}$ whenever the third collision condition is satisfied. Conversely, the configuration must lie in free space if the third collision condition is not satisfied. The first two conditions must be added to the third condition in order to distinguish configurations *on the boundary of* $\mathbf{CO}_{\mathcal{B}}^{\mathcal{A}}$ from configurations inside of $\mathbf{CO}_{\mathcal{B}}^{\mathcal{A}}$.

The first collision condition insures that the configuration of $\mathcal{A}$ lies on at least one C-surface. If the configuration does not lie on any C-surfaces, then it cannot lie on the boundary of $\mathbf{CO}_{\mathcal{B}}^{\mathcal{A}}$. There will typically be many points along the trajectory of $\mathcal{A}$ where its configuration $\{\mathbf{x}, \vec{\Theta}\}$ passes through a C-surface. However, a more detailed discussion of how to find these points where the value of a C-function passes through zero cannot be given unless a specific assumption is made for the trajectory of $\mathcal{A}$. Since this chapter is concerned with those aspects of the theory that apply for any trajectory of $\mathcal{A}$, no further consideration will be given to the first collision condition. When convenient, it will be assumed that configurations where this condition is satisfied can be found by some unspecified means.

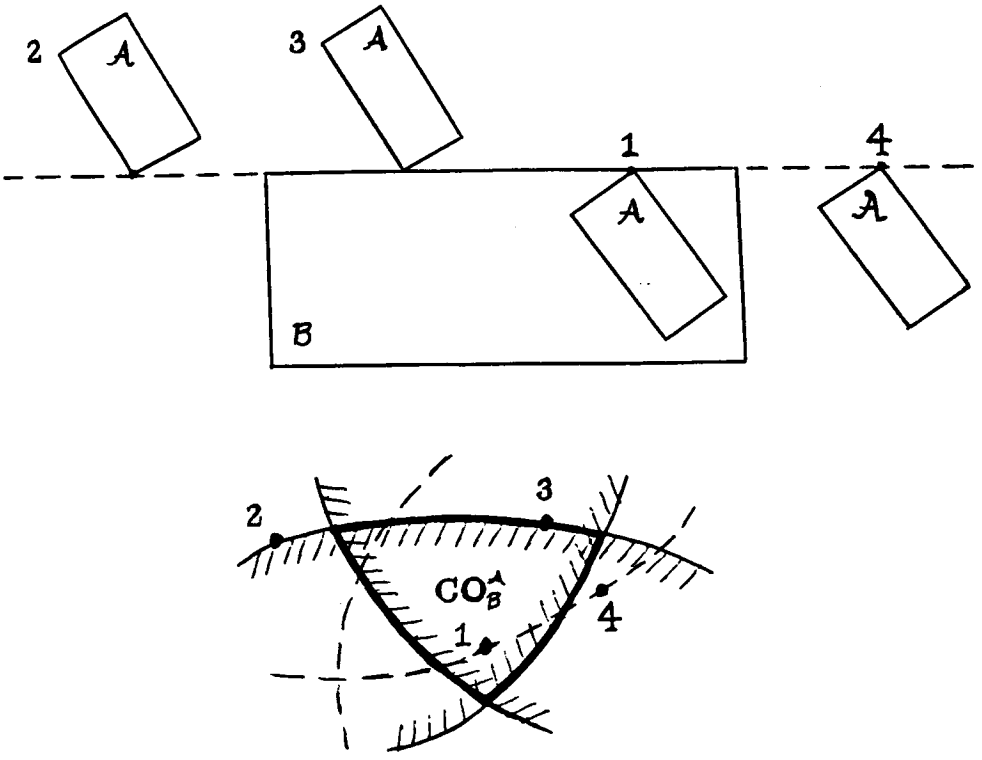
The second collision condition requiring that at least one zero-valued C-function be applicable is a restriction on the orientation parameters $\vec{\Theta}$. The value of a particular C-function being zero implies that the features of $\mathcal{A}$ and $\mathcal{B}$ which define it may be in contact, but no conclusion can be drawn about the relative locations of the other features of $\mathcal{A}$ and $\mathcal{B}$. In particular, it is not known whether or not the interiors of the objects will overlap when the two features of $\mathcal{A}$ and $\mathcal{B}$ come into contact. The condition that a

zero-valued C-function be applicable is used to insure that the interiors of $\mathcal{A}$ and $\mathcal{B}$ do not overlap if the two features defining the C-function are in contact.

The third collision condition is primarily a constraint on the translation parameters $\mathbf{x}$. It arises from the fact that infinite planes are used to define the C-functions while only finite parts of these planes can represent actual contact between the relevant features of $\mathcal{A}$ and $\mathcal{B}$. Whenever the two features of $\mathcal{A}$ and $\mathcal{B}$ defining a particular C-function are in contact, the value of the C-function must be zero. However, the converse is not true. There are configurations where the value of a C-function can be zero and the two features which define it are not in contact. The third collision condition is required to insure that the relevant features for at least one of the applicable C-functions having zero value at some configuration are actually in contact for that configuration.

The collision conditions are both necessary and sufficient for determining when a configuration lies on the boundary of $\mathbf{CO}_{\mathcal{B}}^{\mathcal{A}}$. This can be explained in terms of the contacts between $\mathcal{A}$ and $\mathcal{B}$. Consider, for example, a type (b) contact where some vertex of $\mathcal{A}$ is just touching a face of $\mathcal{B}$. Then this vertex lies in the plane containing the face of $\mathcal{B}$ and, by definition, the corresponding type (b) C-function is zero. Depending on the orientation of $\mathcal{A}$, the interiors of $\mathcal{A}$ and $\mathcal{B}$ may or may not overlap when this contact takes place. Since the boundary of $\mathbf{CO}_{\mathcal{B}}^{\mathcal{A}}$ is by defintion a set of configurations where the interiors of $\mathcal{A}$ and $\mathcal{B}$ *do not* overlap, the fact that the C-function is zero is not sufficient to guarantee that the configuration lies on this boundary. Thus, the condition of applicability must be included. Suppose now that the C-function defined by the vertex of $\mathcal{A}$ and the face of $\mathcal{B}$ has a zero value at some configuration where it is applicable. The definition of the C-function implies that its value is zero whenever the vertex of $\mathcal{A}$ is in the *plane* containing the face of $\mathcal{B}$. The actual face of $\mathcal{B}$ occupies only a finite part of this plane. Therefore, the C-function can be zero when the vertex of $\mathcal{A}$ is located in a part of this plane outside the boundaries of the face of $\mathcal{B}$. The third condition guarantees that the vertex of $\mathcal{A}$ lies within the boundaries of the face of $\mathcal{B}$ at the configuration of interest.

Figure 2.9 shows how the second and third conditions eliminate configurations that are not on the boundary of $\mathbf{CO}_{\mathcal{B}}^{\mathcal{A}}$ for type (b) contacts. Similar arguments to the above can be made for type (a) and (c) contacts.



Four points where a type (b) C-function is zero:

1 & 4 The C-function is not applicable. This implies contact of a vertex of $\mathcal{A}$ with a plane containing a face of $\mathcal{B}$ such that $\mathcal{A}$ and $\mathcal{B}$ would overlap.

2 The C-function is applicable, but the other applicable C-functions with positive orientation are not all nonpositive or the other applicable C-functions with negative orientation are not all nonnegative. This implies contact of a vertex of $\mathcal{A}$ with a plane containing a face of $\mathcal{B}$ but not with the face itself.

3 The C-function is applicable, all the other applicable C-functions with positive orientation are nonpositive, and all the other applicable C-functions with negative orientation are nonnegative. This point lies on the boundary of $\mathbf{CO}_{\mathcal{B}}^{\mathcal{A}}$. This implies a collision of $\mathcal{A}$ and $\mathcal{B}$.

Figure 2.9: Illustration of conditions 2 and 3 for a configuration to lie on the boundary of $\mathbf{CO}_{\mathcal{B}}^{\mathcal{A}}$

Unlike the first collision condition, further details regarding the configurations of $\mathcal{A}$ where the second and third collision conditions are staisfied can be given without assuming a specific trajectory for $\mathcal{A}$. The next section, section 2.4, derives a set of tests that determine whether each of the three types of C-functions is applicable for a given value of $\vec{\Theta}$. Section 2.5 discusses another means for testing the third collision condition that does not require computation of the values of the other applicable C-functions given a configuration where at least one applicable C-function is zero.

2.4 Applicability of C-functions for Convex Polyhedra

The convexity of the polyhedra $\mathcal{A}$ and $\mathcal{B}$ can be exploited to derive simple tests for the applicability of each type of C-function. Since $\mathcal{A}$ is convex, it lies entirely in the closed negative half-space defined by the plane P_i^a containing its ith face. Suppose that $\mathcal{A}$ is translated with fixed orientation $\vec{\Theta}$ until its ith face is touching the jth vertex of $\mathcal{B}$. Then, by definition, the type (a) C-function $f_{i,j}^a$ will be applicable at $\vec{\Theta}$ if $\mathcal{B}$ lies entirely in the closed positive half-space of P_i^a. In other words, $f_{i,j}^a$ will be applicable if P_i^a is the unique separating plane for $\mathcal{A}$ and $\mathcal{B}$ when face i of $\mathcal{A}$ is touching vertex j of $\mathcal{B}$ with $\mathcal{A}$ at orientation $\vec{\Theta}$. Since $\mathcal{B}$ is also convex, it will lie entirely in the closed positive half-space of P_i^a if all of its edges which meet at vertex j are in this half-space. Note that this includes orientations where one or more of these edges lies in P_i^a. It is not possible for all of the edges that meet at a vertex of a polyhedron to be coplanar, so at least one of the edges which meet at vertex j of $\mathcal{B}$ will not lie in P_i^a. Convexity guarantees that the edges of $\mathcal{B}$ meeting at its jth vertex are the only features of $\mathcal{B}$ that need be considered to determine the applicability of $f_{i,j}^a$.

Applicability of the type (b) C-function $f_{i,j}^b$ can be tested in a similar manner. Since $\mathcal{B}$ is convex, it lies entirely in the closed negative half-space defined by the plane P_j^b containing its jth face. Suppose that $\mathcal{A}$ is translated with fixed orientation $\vec{\Theta}$ until its ith vertex is touching the jth face of $\mathcal{B}$. Then, by definition, the type (b) C-function $f_{i,j}^b$ will be applicable at $\vec{\Theta}$ if $\mathcal{A}$ lies entirely in the closed positive half-space of P_j^b. In other words, $f_{i,j}^b$ will be applicable if P_j^b is the unique separating plane for $\mathcal{A}$ and $\mathcal{B}$ when

vertex i of $\mathcal{A}$ is touching face j of $\mathcal{B}$ with $\mathcal{A}$ at orientation $\vec{\Theta}$. Since $\mathcal{A}$ is also convex, it will lie entirely in the closed positive half-space of P_j^b if all of its edges that meet at vertex i are in this half-space. Orientations where some of these edges lie in P_j^b are included in this criterion, but it is not possible for all of them to lie in this plane simultaneously. Convexity guarantees that the edges of $\mathcal{A}$ meeting at its ith vertex are the only features of $\mathcal{A}$ that need be considered to determine the applicability of $f_{i,j}^b$.

If edge i of $\mathcal{A}$ is not parallel to edge j of $\mathcal{B}$ for orientation $\vec{\Theta}$, there is a unique plane P_{ij}^c that contains edge j of $\mathcal{B}$ and is parallel to edge i of $\mathcal{A}$. This plane has normal vector $\mathbf{e}_i^a\left(\vec{\Theta}\right) \times \mathbf{e}_j^b$. Now suppose that $\mathcal{A}$ is translated with fixed orientation $\vec{\Theta}$ until its ith edge is touching the jth edge of $\mathcal{B}$. Then, edge i of $\mathcal{A}$ will also lie in P_{ij}^c. By definition, the type (c) C-function $f_{i,j}^c$ will be applicable at $\vec{\Theta}$ if $\mathcal{B}$ lies entirely in one of the closed half-spaces of P_{ij}^c and $\mathcal{A}$ lies entirely in the other closed half-space of P_{ij}^c. In other words, $f_{i,j}^c$ will be applicable if P_{ij}^c is the unique separating plane for $\mathcal{A}$ and $\mathcal{B}$ when edge i of $\mathcal{A}$ is touching edge j of $\mathcal{B}$ with $\mathcal{A}$ at orientation $\vec{\Theta}$. Since $\mathcal{B}$ is convex, it will lie entirely in one closed half-space of P_{ij}^c if the two faces that meet at its jth edge lie in that half-space. An identical argument holds for the two faces of $\mathcal{A}$ that meet at its ith edge. This criterion includes orientations where one of these faces of $\mathcal{A}$ or $\mathcal{B}$ lies in P_{ij}^c. The two faces of a convex polyhedron that meet at one edge are never coplanar. Therefore, if one of the two faces of either object lies in P_{ij}^c for some orientation of $\mathcal{A}$, the other face will not lie in P_{ij}^c. Convexity guarantees that these faces are the only features of $\mathcal{A}$ and $\mathcal{B}$ that need be considered to determine the applicability of $f_{i,j}^c$.

However, it is not sufficient to determine that both $\mathcal{A}$ and $\mathcal{B}$ lie entirely in one of the closed half-spaces of P_{ij}^c. It must also be true that the half-space of P_{ij}^c containing $\mathcal{A}$ does not contain $\mathcal{B}$ and vice versa. That is, $\mathcal{A}$ and $\mathcal{B}$ must lie in opposite half-spaces if $f_{i,j}^c$ is to be applicable. There are two possibilities: (i) $\mathcal{A}$ lies in the closed positive half-space of P_{ij}^c and $\mathcal{B}$ lies in the closed negative half-space of P_{ij}^c, or (ii) $\mathcal{A}$ lies in the closed negative half-space of P_{ij}^c and $\mathcal{B}$ lies in the closed positive half-space of P_{ij}^c. In the first case, the normal vector $\mathbf{e}_i^a\left(\vec{\Theta}\right) \times \mathbf{e}_j^b$ points into the half-space of P_{ij}^c which does not contain $\mathcal{B}$. This is simply the definition of positive orientation for $f_{i,j}^c$. In the second case, the normal vector $\mathbf{e}_i^a\left(\vec{\Theta}\right) \times \mathbf{e}_j^b$ points

into the half-space of P^c_{ij} which does contain $\mathcal{B}$, which is the definition of negative orientation of $f^c_{i,j}$ Thus, the test for applicability of the type (c) C-function can also determine the orientation of the C-function.

The applicability of $f^c_{i,j}$ can also be tested by looking at the location of edges of $\mathcal{A}$ and $\mathcal{B}$ relative to the plane P^c_{ij}. Let the two faces of $\mathcal{A}$ meeting at its ith edge be called faces $1a$ and $2a$, respectively. Then face $1a$ or $2a$ will lie entirely in one half-space of P^c_{ij} if any one of the points in that face, other than a point lying along edge i, lies in that half-space. In particular, this is true for all of the points along an edge of $\mathcal{A}$ which bounds face $1a$ or $2a$ other than edge i. The two edges which meet at the tail vertex of edge i of $\mathcal{A}$ and bound faces $1a$ and $2a$, respectively, are arbitrarily chosen for this test. Let these edges be denoted as edges $1a$ and $2a$. If the two faces of $\mathcal{B}$ meeting at its jth edge are called faces $1b$ and $2b$, then they will lie entirely in one of the half-spaces of P^c_{ij} if any one of their edges other than edge j lies in that half-space of P^c_{ij}. Similar to $\mathcal{A}$, the two edges of $\mathcal{B}$ which meet at the tail vertex of edge j and lie in faces $1b$ and $2b$, respectively, are chosen to implement this test and denoted edges $1b$ and $2b$. In terms of these pairs of edges of $\mathcal{A}$ and $\mathcal{B}$, the applicability test for the type (c) C-function $f^c_{i,j}$ becomes:

- Edges $1a$ and $2a$ of $\mathcal{A}$ must lie in the *same* half-space of P^c_{ij}.
- Edges $1b$ and $2b$ of $\mathcal{B}$ must lie in the *same* half-space of P^c_{ij}.
- Edges $1a$ and $2a$ of $\mathcal{A}$ must lie in the *opposite* half-space of P^c_{ij} from edges $1b$ and $2b$.

If all three of these tests are satisfied, then $f^c_{i,j}$ is applicable and its orientation is positive if edges $1a$ and $2a$ lie in the positive half-space of P^c_{ij} and negative if edges $1a$ and $2a$ lie in the negative half-space of P^c_{ij}.

2.4.1 Applicability Constraint Functions

The applicability tests can be implemented by checking the values of a group of functions of $\vec{\Theta}$ defined for each C-function. These functions are called *applicability constraint functions* or ACFs. One ACF is defined for each of the features of $\mathcal{A}$ or $\mathcal{B}$ specified in the applicability tests for the different types of C-functions. Only the sign of a particular ACF is required in the test for applicability because this will determine which open half-space of

the potential separating plane contains the relevant feature of $\mathcal{A}$ or $\mathcal{B}$. $\mathcal{A}$ or $\mathcal{B}$ can be shown to lie entirely in one of the closed half-spaces of the potential separating plane by examining the signs of all the ACFs for a particular C-function.

The type (a) C-function $f^a_{i,j}$ is applicable if the edges of $\mathcal{B}$ that meet at its jth vertex all lie in the closed positive half-space of the plane P^a_i containing the ith face of $\mathcal{A}$. The half-space of P^a_i containing one of these edges of $\mathcal{B}$ can be identified by checking the sign of the component of one of the two edge vectors for this edge along the unit, outward-pointing normal vector for face i of $\mathcal{A}$. Hence, the ACFs for $f^a_{i,j}$ are taken to be the inner products of edge vectors for each of the edges of $\mathcal{B}$ that meet at vertex j with the unit, outward-pointing normal vector for face i of $\mathcal{A}$.

Let $\mathbf{e}^b_k$ be the edge vector along one of the edges of $\mathcal{B}$ meeting at vertex j that has its tail vertex at vertex j. The unit, outward-pointing normal vector for face i of $\mathcal{A}$ is denoted $\mathbf{u}^a_i\left(\vec{\Theta}\right)$ as before. The ACF for $f^a_{i,j}$ for edge k of $\mathcal{B}$ is then defined as

$$g^a_k\left(\vec{\Theta}\right) = \left\langle \mathbf{e}^b_k, \mathbf{u}^a_i\left(\vec{\Theta}\right)\right\rangle$$

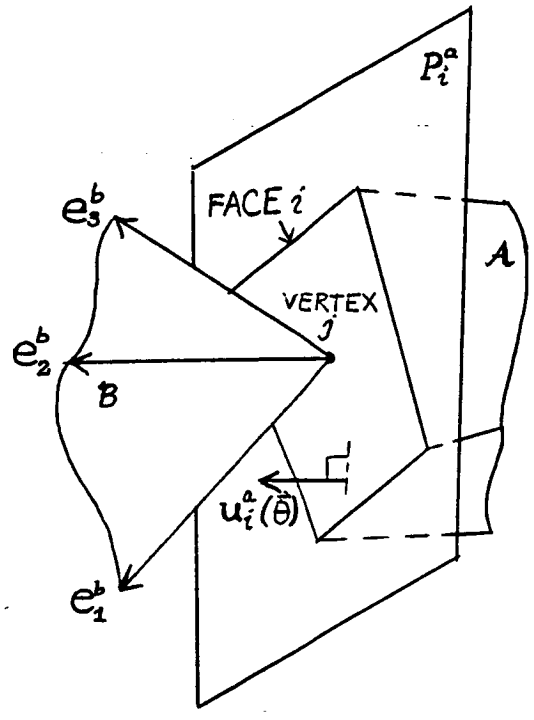
Edge k of $\mathcal{B}$ lies in the open positive half-space of P^a_i if $g^a_k\left(\vec{\Theta}\right)$ is positive and it lies in the open negative half-space of P^a_i if $g^a_k\left(\vec{\Theta}\right)$ is negative. This is because $\mathbf{e}^b_k$ is chosen to point away from vertex j of $\mathcal{B}$ and $\mathbf{u}^a_i\left(\vec{\Theta}\right)$ is defined as the outward-pointing normal vector for face i of $\mathcal{A}$. The ACF $g^a_k\left(\vec{\Theta}\right)$ will be equal to zero if edge k of $\mathcal{B}$ lies in the plane P^a_i.

One ACF $g^a_k\left(\vec{\Theta}\right)$ is defined for each of the edges of $\mathcal{B}$ which meet at vertex j. There will be at least three ACFs for any type (a) C-function since there must be three or more edges meeting at any vertex of $\mathcal{B}$. At least one of the ACFs $\left\{g^a_k\left(\vec{\Theta}\right)\right\}$ will be nonzero for any value of $\vec{\Theta}$.

The direction of the edge vectors as indicated by the common choice for their tail vertex determines that nonnegative values of all the ACFs indicate that all of the edges of $\mathcal{B}$ that meet at its jth vertex lie in the closed positive half-space of P^a_i. Therefore, the type (a) C-function $f^a_{i,j}$ is applicable at orientation $\vec{\Theta}$ if

$$g^a_k\left(\vec{\Theta}\right) \geq 0 \quad \forall \quad k \in \{\text{edges of } \mathcal{B} \text{ meeting at its } j\text{th vertex}\}$$

This is illustrated in Figure 2.10.



P_i^a separates $\mathcal{A}$ and $\mathcal{B}$
Edges 1, 2, and 3 meet at vertex j of $\mathcal{B}$

$$g_1^a\left(\vec{\Theta}\right) \geq 0 \quad g_2^a\left(\vec{\Theta}\right) \geq 0 \quad g_3^a\left(\vec{\Theta}\right) \geq 0$$

$f_{i,j}^a$ is applicable at this orientation of $\mathcal{A}$

Figure 2.10: Applicability tests for the type (a) C-function $f_{i,j}^a$

Analogously, the type (b) C-function $f^b_{i,j}$ is applicable if the edges of $\mathcal{A}$ that meet at its ith vertex all lie in the closed positive half-space of the plane P^b_j containing the jth face of $\mathcal{B}$. In this case, the sign of the component of one of the two edge vectors for an edge of $\mathcal{A}$ that meets at its ith vertex along the unit, outward-pointing normal vector for face j of $\mathcal{B}$ identifies the half-space of P^b_j that contains this edge of $\mathcal{A}$. The ACFs for $f^b_{i,j}$ are the inner products of the unit, outward-pointing normal vector for face j of $\mathcal{B}$ and edge vectors along each of the edges that meet at vertex i of $\mathcal{A}$.

Let $\mathbf{e}^a_k\left(\vec{\Theta}\right)$ be the edge vector along one of the edges of $\mathcal{A}$ meeting at vertex i that has its tail vertex at vertex i. Also let $\mathbf{u}^b_j$ denote the unit, outward-pointing normal vector for face j of $\mathcal{B}$ as before. Then, the ACF for $f^b_{i,j}$ for this edge of $\mathcal{A}$ is defined as

$$g^b_k\left(\vec{\Theta}\right) = \left\langle \mathbf{e}^a_k\left(\vec{\Theta}\right), \mathbf{u}^b_j \right\rangle$$

The choice that $\mathbf{e}^a_k\left(\vec{\Theta}\right)$ point away from vertex i of $\mathcal{A}$ and the definition of $\mathbf{u}^b_j$ as the outward-pointing normal vector for face j of $\mathcal{B}$ imply that edge k of $\mathcal{A}$ lies in the open positive half-space of P^b_j if $g^b_k\left(\vec{\Theta}\right)$ is positive and that it lies in the open negative half-space of P^b_j if $g^b_k\left(\vec{\Theta}\right)$ is negative. When edge k of $\mathcal{A}$ lies in the plane P^b_j, the ACF $g^b_k\left(\vec{\Theta}\right)$ is zero.

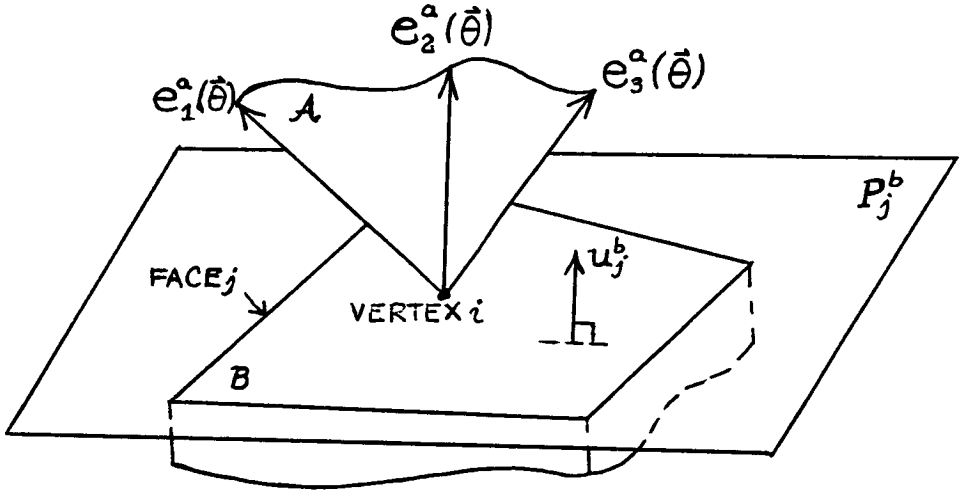
A different ACF $g^b_k\left(\vec{\Theta}\right)$ for $f^b_{i,j}$ is defined for each of the edges of $\mathcal{A}$ that meet at vertex i. Any type (b) C-function will have at least three ACFs because three or more edges meet at any vertex of $\mathcal{A}$. At least one of the ACFs $\left\{g^b_k\left(\vec{\Theta}\right)\right\}$ will be nonzero for any orientation $\vec{\Theta}$.

As was the case for the type (a) C-function, the direction of the edge vectors of $\mathcal{A}$ as indicated by the choice of vertex i as their common tail vertex determines that nonnegative values of all the ACFs indicate that all of these edges of $\mathcal{A}$ lie in the closed positive half-space of P^b_j. Thus, the type (b) C-function $f^b_{i,j}$ is applicable at orientation $\vec{\Theta}$ if

$$g^b_k\left(\vec{\Theta}\right) \geq 0 \quad \forall \quad k \in \{\text{edges of } \mathcal{A} \text{ meeting at its } i\text{th vertex}\}$$

This is illustrated in Figure 2.11.

The situation is somewhat more complicated for a type (c) C-function. To test the applicability of $f^c_{i,j}$ it is first necessary to determine if faces $1a$



P_j^b separates $\mathcal{A}$ and $\mathcal{B}$
Edges 1, 2, and 3 meet at vertex i of $\mathcal{A}$

$$g_1^b\left(\vec{\Theta}\right) \geq 0 \quad g_2^b\left(\vec{\Theta}\right) \geq 0 \quad g_3^b\left(\vec{\Theta}\right) \geq 0$$

$f_{i,j}^b$ is applicable at this orientation of $\mathcal{A}$

Figure 2.11: Applicability tests for the type (b) C-function $f_{i,j}^b$

and $2a$ of $\mathcal{A}$ that meet at its ith edge both lie in the same closed half-space of P_{ij}^c. Let $\mathbf{e}_{1a}^a\left(\vec{\Theta}\right)$ and $\mathbf{e}_{2a}^a\left(\vec{\Theta}\right)$ be the edge vectors along edges $1a$ and $2a$ that meet at the tail vertex of edge i of $\mathcal{A}$, bound faces $1a$ and $2a$, respectvely, and share their tail vertices with edge i of $\mathcal{A}$. Faces $1a$ and $2a$ will both lie in the closed positive half-space of P_{ij}^c if the inner products of $\mathbf{e}_{1a}^a\left(\vec{\Theta}\right)$ and $\mathbf{e}_{2a}^a\left(\vec{\Theta}\right)$ with the normal vector to $\mathbf{e}_i^a\left(\vec{\Theta}\right) \times \mathbf{e}_j^b$ are both nonnegative. These faces will both lie in the closed negative half-space of P_{ij}^c if these inner products are both nonpositive.

The two ACFs for $f_{i,j}^c$ corresponding to $\mathbf{e}_{1a}^a\left(\vec{\Theta}\right)$ and $\mathbf{e}_{2a}^a\left(\vec{\Theta}\right)$ are defined to be the inner products shown below:

$$\begin{aligned} g_{1a}^c\left(\vec{\Theta}\right) &= \left\langle \mathbf{e}_{1a}^a\left(\vec{\Theta}\right), \mathbf{e}_i^a\left(\vec{\Theta}\right) \times \mathbf{e}_j^b \right\rangle \\ g_{2a}^c\left(\vec{\Theta}\right) &= \left\langle \mathbf{e}_{2a}^a\left(\vec{\Theta}\right), \mathbf{e}_i^a\left(\vec{\Theta}\right) \times \mathbf{e}_j^b \right\rangle \end{aligned}$$

If either one of the edges $1a$ or $2a$ lies in P_{ij}^c itself, then either $g_{1a}^c\left(\vec{\Theta}\right)$ or $g_{2a}^c\left(\vec{\Theta}\right)$ will be zero, but both edges cannot lie in P_{ij}^c simultaneously. At least one of the ACFs $g_{1a}^c\left(\vec{\Theta}\right)$ or $g_{2a}^c\left(\vec{\Theta}\right)$ will be nonzero at any orientation $\vec{\Theta}$. The criterion that faces $1a$ and $2a$ of $\mathcal{A}$ both lie in the same closed half-space of P_{ij}^c is

$$\begin{gathered} g_{1a}^c\left(\vec{\Theta}\right) \geq 0 \quad \text{and} \quad g_{2a}^c\left(\vec{\Theta}\right) \geq 0 \\ \text{or} \\ g_{1a}^c\left(\vec{\Theta}\right) \leq 0 \quad \text{and} \quad g_{2a}^c\left(\vec{\Theta}\right) \leq 0 \end{gathered}$$

If this criterion is satisfied, define S_a to be the common sign of $g_{1a}^c\left(\vec{\Theta}\right)$ and $g_{2a}^c\left(\vec{\Theta}\right)$. $S_a = +$ when $\mathcal{A}$ lies entirely in the closed positive half-space of P_{ij}^c. Conversely, $\mathcal{A}$ lies entirely in the closed negative half-space of P_{ij}^c if $S_a = -$.

The next step in testing the applicability of $f_{i,j}^c$ is to determine whether the two faces of $\mathcal{B}$, $1b$ and $2b$, meeting at its jth edge both lie in the same half-space of P_{ij}^c. Following the analysis for the faces of $\mathcal{A}$, let edge vectors $\mathbf{e}_{1b}^b$ and $\mathbf{e}_{2b}^b$ be the vectors along the two edges of $\mathcal{B}$ which meet at the tail vertex of its jth edge, bound faces $1b$ and $2b$, respectively, and have the same tail vertex as the jth edge vector. Then faces $1b$ and $2b$ of $\mathcal{B}$ will both lie entirely in one of the closed half-spaces of P_{ij}^c if the inner products

of $\mathbf{e}^b_{1b}$ and $\mathbf{e}^b_{2b}$ with the normal to P^c_{ij} are either both nonnegative or both nonpositive.

The two ACFs for $f^c_{i,j}$ corresponding to $\mathbf{e}^b_{1b}$ and $\mathbf{e}^b_{2b}$ are defined as

$$g^c_{1b}\left(\vec{\Theta}\right) = \left\langle \mathbf{e}^b_{1b}, \mathbf{e}^a_i\left(\vec{\Theta}\right) \times \mathbf{e}^b_j \right\rangle$$
$$g^c_{2b}\left(\vec{\Theta}\right) = \left\langle \mathbf{e}^b_{2b}, \mathbf{e}^a_i\left(\vec{\Theta}\right) \times \mathbf{e}^b_j \right\rangle$$

Either $g^c_{1b}\left(\vec{\Theta}\right)$ or $g^c_{2b}\left(\vec{\Theta}\right)$ will be zero if one of the edges $1b$ or $2b$ lies in P^c_{ij} itself. However, both edges cannot lie in P^c_{ij} simultaneously. At least one of the ACFs $g^c_{1b}\left(\vec{\Theta}\right)$ or $g^c_{2b}\left(\vec{\Theta}\right)$ will be nonzero at any orientation $\vec{\Theta}$. The criterion that faces $1b$ and $2b$ of $\mathcal{B}$ lie in the same closed half-space of P^c_{ij} is

$$g^c_{1b}\left(\vec{\Theta}\right) \geq 0 \quad \text{and} \quad g^c_{2b}\left(\vec{\Theta}\right) \geq 0$$
$$\text{or}$$
$$g^c_{1b}\left(\vec{\Theta}\right) \leq 0 \quad \text{and} \quad g^c_{2b}\left(\vec{\Theta}\right) \leq 0$$

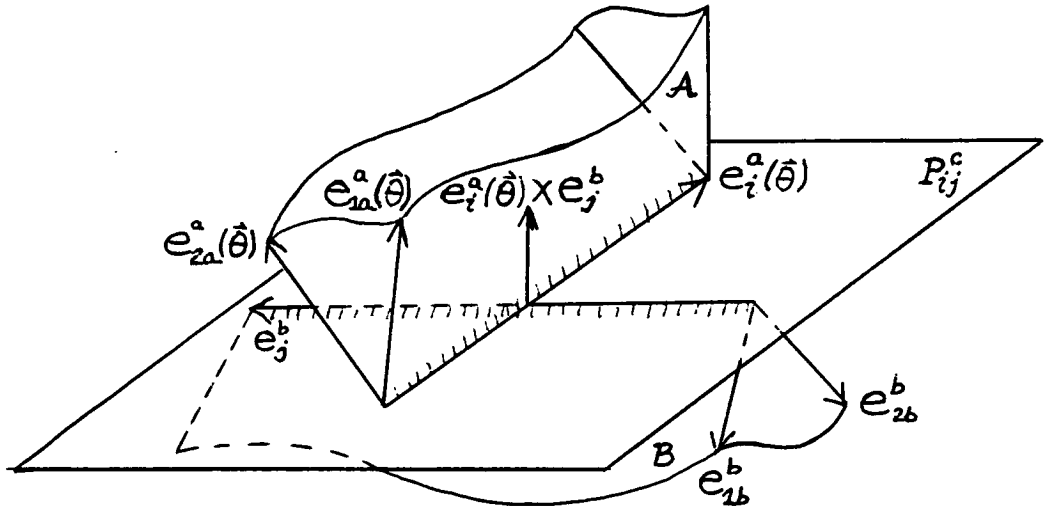
Define S_b to be the common sign of $g^c_{1b}\left(\vec{\Theta}\right)$ or $g^c_{2b}\left(\vec{\Theta}\right)$ when this criterion is satisfied. Then, $S_b = +$ when $\mathcal{B}$ lies entirely in the closed positive half-space of P^c_{ij} and $S_b = -$ when $\mathcal{B}$ lies entirely in the closed negative half-space of P^c_{ij}.

Once it has been determined that $\mathcal{A}$ and $\mathcal{B}$ both lie entirely in one half-space of P^c_{ij}, the final step in testing the applicability of $f^c_{i,j}$ is to determine whether these half-spaces are opposite. This is easily done by comparing S_a and S_b. They must be different for the objects to lie in opposite half-spaces of P^c_{ij}. Thus, the third and final criterion for the applicability of $f^c_{i,j}$ at orientation $\vec{\Theta}$ is $S_a \neq S_b$.

If all three of the applicability tests described above are successful, then the orientation of $f^c_{i,j}$ is easily determined by distinguishing the case where $S_a = +$ and $S_b = -$ from the case where $S_a = -$ and $S_b = +$. In the first case, $f^c_{i,j}$ has positive orientation at $\vec{\Theta}$ while in the second case it has negative orientation at $\vec{\Theta}$. The applicability tests for $f^c_{i,j}$ are illustrated in Figure 2.12.

2.4.2 The Two Types of ACFs

The four ACFs for the type (c) C-function are defined as scalar triple products involving either two edge vectors of $\mathcal{A}$ and one edge vector of $\mathcal{B}$ or



P^c_{ij} separates $\mathcal{A}$ and $\mathcal{B}$

$$\left.\begin{array}{l} g^c_{1a}\left(\vec{\Theta}\right) \geq 0 \\ g^c_{2a}\left(\vec{\Theta}\right) \geq 0 \end{array}\right\} \Rightarrow S_a = + \quad \left.\begin{array}{l} g^c_{1b}\left(\vec{\Theta}\right) \leq 0 \\ g^c_{2b}\left(\vec{\Theta}\right) \leq 0 \end{array}\right\} \Rightarrow S_b = -$$

$f^c_{i,j}$ is applicable and has positive orientation

Figure 2.12: Applicability and orientation tests for the type (c) C-function $f^c_{i,j}$

two edge vectors of $\mathcal{B}$ and one edge vector of $\mathcal{A}$. The scalar triple product identities

$$\langle \mathbf{a}, \mathbf{b} \times \mathbf{c} \rangle = \langle \mathbf{c}, \mathbf{a} \times \mathbf{b} \rangle = \langle \mathbf{b}, \mathbf{c} \times \mathbf{a} \rangle$$

can be used to replace the type (c) ACFs with new functions. These new functions behave identically to the original type (c) ACFs but have the same form as the type (a) and (b) ACFs, as will be shown below.

Each ACF for a type (a) C-function is an inner product of a normal vector to a face of $\mathcal{A}$ and an edge vector of $\mathcal{B}$. The first two type (c) ACFs, $g^c_{1a}\left(\vec{\Theta}\right)$ and $g^c_{2a}\left(\vec{\Theta}\right)$, can be replaced by inner products of this form. Using the scalar triple product identity,

$$\begin{aligned} g^c_{1a}\left(\vec{\Theta}\right) &= \left\langle \mathbf{e}^a_{1a}\left(\vec{\Theta}\right), \mathbf{e}^a_i\left(\vec{\Theta}\right) \times \mathbf{e}^b_j \right\rangle = \left\langle \mathbf{e}^b_j, \mathbf{e}^a_{1a}\left(\vec{\Theta}\right) \times \mathbf{e}^a_i\left(\vec{\Theta}\right) \right\rangle \\ g^c_{2a}\left(\vec{\Theta}\right) &= \left\langle \mathbf{e}^a_{2a}\left(\vec{\Theta}\right), \mathbf{e}^a_i\left(\vec{\Theta}\right) \times \mathbf{e}^b_j \right\rangle = \left\langle \mathbf{e}^b_j, \mathbf{e}^a_{2a}\left(\vec{\Theta}\right) \times \mathbf{e}^a_i\left(\vec{\Theta}\right) \right\rangle \end{aligned}$$

Since both $\mathbf{e}^a_{1a}\left(\vec{\Theta}\right)$ and $\mathbf{e}^a_i\left(\vec{\Theta}\right)$ lie on face $1a$ of $\mathcal{A}$, the cross product vector $\mathbf{e}^a_{1a}\left(\vec{\Theta}\right) \times \mathbf{e}^a_i\left(\vec{\Theta}\right)$ must point either along or opposite to $\mathbf{u}^a_{1a}\left(\vec{\Theta}\right)$, the unit outward-pointing normal vector to face $1a$ of $\mathcal{A}$. Both $\mathbf{e}^a_{2a}\left(\vec{\Theta}\right)$ and $\mathbf{e}^a_i\left(\vec{\Theta}\right)$ lie on face $2a$ of $\mathcal{A}$, so $\mathbf{e}^a_{2a}\left(\vec{\Theta}\right) \times \mathbf{e}^a_i\left(\vec{\Theta}\right)$ must point either along or opposite to $\mathbf{u}^a_{2a}\left(\vec{\Theta}\right)$, the unit outward-pointing normal vector to face $2a$ of $\mathcal{A}$. The two cross products of edges of $\mathcal{A}$ can be written as

$$\begin{aligned} \mathbf{e}^a_{1a}\left(\vec{\Theta}\right) \times \mathbf{e}^a_i\left(\vec{\Theta}\right) &= \pm k_{1a}\, \mathbf{u}^a_{1a}\left(\vec{\Theta}\right) \\ \mathbf{e}^a_{2a}\left(\vec{\Theta}\right) \times \mathbf{e}^a_i\left(\vec{\Theta}\right) &= \mp k_{2a}\, \mathbf{u}^a_{2a}\left(\vec{\Theta}\right) \end{aligned}$$

where k_{1a} and k_{2a} are positive constants.

It is not known a priori which choice of signs should be made in the above equations. However, if one of the cross product vectors points along the corresponding face normal, the other will point opposite to its corresponding face normal and vice-versa. This results from the fact that $\mathcal{A}$ lies entirely in the negative half-space of the plane containing one of its faces. Since $\mathbf{e}^a_{2a}\left(\vec{\Theta}\right)$ points away from the tail vertex of edge i of $\mathcal{A}$, it points into the negative half-space of the plane containing face $1a$ of $\mathcal{A}$ or

$$\left\langle \mathbf{e}^a_{2a}\left(\vec{\Theta}\right), \mathbf{u}^a_{1a}\left(\vec{\Theta}\right) \right\rangle < 0$$

Suppose $\mathbf{e}^a_{1a}\left(\vec{\Theta}\right) \times \mathbf{e}^a_i\left(\vec{\Theta}\right) = +k_{1a}\,\mathbf{u}^a_{1a}\left(\vec{\Theta}\right)$. Then

$$\left\langle \mathbf{e}^a_{2a}\left(\vec{\Theta}\right), \mathbf{e}^a_{1a}\left(\vec{\Theta}\right) \times \mathbf{e}^a_i\left(\vec{\Theta}\right)\right\rangle = k_{1a}\left\langle \mathbf{e}^a_{2a}\left(\vec{\Theta}\right), \mathbf{u}^a_{1a}\left(\vec{\Theta}\right)\right\rangle < 0$$

From the scalar triple product identity

$$\begin{aligned}\left\langle \mathbf{e}^a_{2a}\left(\vec{\Theta}\right), \mathbf{e}^a_{1a}\left(\vec{\Theta}\right) \times \mathbf{e}^a_i\left(\vec{\Theta}\right)\right\rangle &= \left\langle \mathbf{e}^a_{1a}\left(\vec{\Theta}\right), \mathbf{e}^a_i\left(\vec{\Theta}\right) \times \mathbf{e}^a_{2a}\left(\vec{\Theta}\right)\right\rangle \\ &= \left\langle \mathbf{e}^a_{1a}\left(\vec{\Theta}\right), -\left(\mathbf{e}^a_{2a}\left(\vec{\Theta}\right) \times \mathbf{e}^a_i\left(\vec{\Theta}\right)\right)\right\rangle\end{aligned}$$

This means that

$$\left\langle \mathbf{e}^a_{1a}\left(\vec{\Theta}\right), -\left(\mathbf{e}^a_{2a}\left(\vec{\Theta}\right) \times \mathbf{e}^a_i\left(\vec{\Theta}\right)\right)\right\rangle < 0$$

But $\mathbf{e}^a_{1a}\left(\vec{\Theta}\right)$ points away from the tail vertex of edge i of $\mathcal{A}$. Therefore, it points into the negative half-space of the plane containing face $2a$ of $\mathcal{A}$ so that

$$\left\langle \mathbf{e}^a_{1a}\left(\vec{\Theta}\right), \mathbf{u}^a_{2a}\left(\vec{\Theta}\right)\right\rangle < 0$$

Therefore $-\left(\mathbf{e}^a_{2a}\left(\vec{\Theta}\right) \times \mathbf{e}^a_i\left(\vec{\Theta}\right)\right)$ points along $\mathbf{u}^a_{2a}\left(\vec{\Theta}\right)$ and

$$\mathbf{e}^a_{2a}\left(\vec{\Theta}\right) \times \mathbf{e}^a_i\left(\vec{\Theta}\right) = -k_{2a}\,\mathbf{u}^a_{2a}\left(\vec{\Theta}\right)$$

A similar argument gives the result that $\mathbf{e}^a_{2a}\left(\vec{\Theta}\right) \times \mathbf{e}^a_i\left(\vec{\Theta}\right) = +k_{2a}\,\mathbf{u}^a_{2a}\left(\vec{\Theta}\right)$ if $\mathbf{e}^a_{1a}\left(\vec{\Theta}\right) \times \mathbf{e}^a_i\left(\vec{\Theta}\right) = -k_{1a}\,\mathbf{u}^a_{1a}\left(\vec{\Theta}\right)$.

The direction of either $\mathbf{e}^a_{1a}\left(\vec{\Theta}\right) \times \mathbf{e}^a_i\left(\vec{\Theta}\right)$ or $\mathbf{e}^a_{2a}\left(\vec{\Theta}\right) \times \mathbf{e}^a_i\left(\vec{\Theta}\right)$ can be checked by actually computing the cross product vector and checking the sign of its inner product with either $\mathbf{u}^a_{1a}\left(\vec{\Theta}\right)$ or $\mathbf{u}^a_{2a}\left(\vec{\Theta}\right)$. This need only be done for one of the cross products since the direction of the other in relation to the corresponding face normal is opposite to that for the cross product that is checked.

Once the relation of $\mathbf{e}^a_{1a}\left(\vec{\Theta}\right) \times \mathbf{e}^a_i\left(\vec{\Theta}\right)$ and $\mathbf{e}^a_{2a}\left(\vec{\Theta}\right) \times \mathbf{e}^a_i\left(\vec{\Theta}\right)$ to the face normals $\mathbf{u}^a_{1a}\left(\vec{\Theta}\right)$ and $\mathbf{u}^a_{2a}\left(\vec{\Theta}\right)$ has been established, the ACFs $g^c_{1a}\left(\vec{\Theta}\right)$ and $g^c_{2a}\left(\vec{\Theta}\right)$ can be replaced by

$$\begin{aligned}g^c_{1a}\left(\vec{\Theta}\right) &= \left\langle \mathbf{e}^b_j, \pm\,\mathbf{u}^a_{1a}\left(\vec{\Theta}\right)\right\rangle \\ g^c_{2a}\left(\vec{\Theta}\right) &= \left\langle \mathbf{e}^b_j, \mp\,\mathbf{u}^a_{2a}\left(\vec{\Theta}\right)\right\rangle\end{aligned}$$

These new functions are proportional to the original ones and behave identically to the original ACFs in the applicability tests.

Each ACF for a type (b) C-function is an inner product of a normal vector to a face of $\mathcal{B}$ and an edge vector of $\mathcal{A}$. The last two type (c) ACFs, $g^c_{1b}\left(\vec{\Theta}\right)$ and $g^c_{2b}\left(\vec{\Theta}\right)$, can be replaced by inner products of this form. From the scalar triple product identity,

$$\begin{aligned} g^c_{1b}\left(\vec{\Theta}\right) &= \left\langle \mathbf{e}^b_{1b}, \mathbf{e}^a_i\left(\vec{\Theta}\right) \times \mathbf{e}^b_j \right\rangle = \left\langle \mathbf{e}^a_i\left(\vec{\Theta}\right), \mathbf{e}^b_j \times \mathbf{e}^b_{1b} \right\rangle \\ g^c_{2b}\left(\vec{\Theta}\right) &= \left\langle \mathbf{e}^b_{2b}, \mathbf{e}^a_i\left(\vec{\Theta}\right) \times \mathbf{e}^b_j \right\rangle = \left\langle \mathbf{e}^a_i\left(\vec{\Theta}\right), \mathbf{e}^b_j \times \mathbf{e}^b_{2b} \right\rangle \end{aligned}$$

Since both $\mathbf{e}^b_{1b}$ and $\mathbf{e}^b_j$ lie on face $1b$ of $\mathcal{B}$, the cross product vector $\mathbf{e}^b_j \times \mathbf{e}^b_{1b}$ must point either along or opposite to $\mathbf{u}^b_{1b}$, the unit outward-pointing normal vector to face $1b$ of $\mathcal{B}$. Both $\mathbf{e}^b_{2b}$ and $\mathbf{e}^b_j$ lie on face $2b$ of $\mathcal{B}$, so $\mathbf{e}^b_j \times \mathbf{e}^b_{2b}$ must point either along or opposite to $\mathbf{u}^b_{2b}$, the unit outward-pointing normal vector to face $2b$ of $\mathcal{B}$. The two cross products of edges of $\mathcal{B}$ can be written as

$$\begin{aligned} \mathbf{e}^b_j \times \mathbf{e}^b_{1b} &= \pm k_{1b}\, \mathbf{u}^b_{1b} \\ \mathbf{e}^b_j \times \mathbf{e}^b_{2b} &= \mp k_{2b}\, \mathbf{u}^b_{2b} \end{aligned}$$

where k_{1b} and k_{2b} are positive constants. The signs in the above equation are determined by actually computing one of the two cross product vectors and finding the sign of its inner product with the appropriate normal vector. Only one of the cross products need be checked because the other will have the opposite relationship to its corresponding face normal. The proof of this follows the proof given for the edges and faces of $\mathcal{A}$ in the preceding paragraph.

Once the relation of $\mathbf{e}^b_j \times \mathbf{e}^b_{1b}$ and $\mathbf{e}^b_j \times \mathbf{e}^b_{2b}$ to the face normals $\mathbf{u}^b_{1b}$ and $\mathbf{u}^b_{2b}$ is known, the ACFs $g^c_{1b}\left(\vec{\Theta}\right)$ and $g^c_{2b}\left(\vec{\Theta}\right)$ can be replaced by

$$\begin{aligned} g^c_{1b}\left(\vec{\Theta}\right) &= \left\langle \mathbf{e}^a_i\left(\vec{\Theta}\right), \pm \mathbf{u}^b_{1b} \right\rangle \\ g^c_{2b}\left(\vec{\Theta}\right) &= \left\langle \mathbf{e}^a_i\left(\vec{\Theta}\right), \mp \mathbf{u}^b_{2b} \right\rangle \end{aligned}$$

These new functions are proportional to the original ones and behave identically to the original ACFs in the applicability tests.

There are now only two distinct forms of the ACFs as functions of $\vec{\Theta}$ used in the applicability tests. A type (i) ACF, denoted $g^i\left(\vec{\Theta}\right)$, is the inner product of an edge vector of $\mathcal{B}$ with either an outward-pointing face

normal for $\mathcal{A}$ or its opposite

$$g^{i}\left(\vec{\Theta}\right) = \left\langle \mathbf{e}^{b}, \pm \mathbf{u}^{a}\left(\vec{\Theta}\right)\right\rangle$$

There is one type (i) ACF for each combination of an edge of $\mathcal{B}$ with a face of $\mathcal{A}$. A type (ii) ACF is denoted $g^{ii}\left(\vec{\Theta}\right)$ and is the inner product of an edge vector of $\mathcal{A}$ with either an outward-pointing face normal of $\mathcal{B}$ or its opposite.

$$g^{ii}\left(\vec{\Theta}\right) = \left\langle \mathbf{e}^{a}\left(\vec{\Theta}\right), \pm \mathbf{u}^{b}\right\rangle$$

There is one type (ii) ACF for each combination of an edge of $\mathcal{A}$ with a face of $\mathcal{B}$. The choice of the sign for the face normal vector and the tail vertex for the edge vector for a particular type (i) or (ii) ACF depends on the geometry of $\mathcal{A}$ and $\mathcal{B}$ and on the type of C-function being tested for applicability.

The applicability tests for type (a), (b), and (c) C-functions in terms of the two types of ACFs are summarized below.

Applicability of a Type (a) C-function All of the ACFs for the type (a) C-function $f^{a}_{i,j}$ corresponding to the ith face of $\mathcal{A}$ and the jth vertex of $\mathcal{B}$ are type (i) ACFs. One type (i) ACF is defined for each of the edges of $\mathcal{B}$ which meet at its jth vertex. The ACF for the kth edge of $\mathcal{B}$ intersecting vertex j is

$$g^{i}_{k}\left(\vec{\Theta}\right) = \left\langle \mathbf{e}^{b}_{k}, +\mathbf{u}^{a}_{i}\left(\vec{\Theta}\right)\right\rangle$$

where the tail vertex of $\mathbf{e}^{b}_{k}$ must be vertex j. Then, the type (a) C-function $f^{a}_{i,j}$ is applicable at orientation $\vec{\Theta}$ if

$$g^{i}_{k}\left(\vec{\Theta}\right) \geq 0 \quad \forall \quad k \in \{\text{edges of } \mathcal{B} \text{ meeting at its } j\text{th vertex}\}$$

The fact that nonnegative values of these ACFs imply that $f^{a}_{i,j}$ is applicable results from the choice of the plus sign for $\mathbf{u}^{a}_{i}\left(\vec{\Theta}\right)$ and the jth vertex of $\mathcal{B}$ for the tail vertex of $\mathbf{e}^{b}_{k}$ in the definition of $g^{i}_{k}\left(\vec{\Theta}\right)$.

Applicability of a Type (b) C-function All of the ACFs for the type (b) C-function $f^{b}_{i,j}$ corresponding to the ith vertex of $\mathcal{A}$ and the jth face of $\mathcal{B}$ are type (ii) ACFs. One type (ii) ACF is defined for each of the edges

of $\mathcal{A}$ which meet at its ith vertex. The ACF for the kth edge of $\mathcal{A}$ which intersects vertex i is

$$g_k^{ii}\left(\vec{\Theta}\right) = \left\langle \mathbf{e}_k^a\left(\vec{\Theta}\right), +\mathbf{u}_j^b \right\rangle$$

where the tail vertex of $\mathbf{e}_k^a\left(\vec{\Theta}\right)$ must be vertex i. Then, the type (b) C-function $f_{i,j}^b$ is applicable at orientation $\vec{\Theta}$ if

$$g_k^{ii}\left(\vec{\Theta}\right) \geq 0 \quad \forall \quad k \in \{\text{edges of } \mathcal{A} \text{ meeting at its } i\text{th vertex}\}$$

Nonnegative values of these ACFs imply that $f_{i,j}^b$ is applicable because the plus sign is chosen for $\mathbf{u}_j^b$ and the ith vertex of $\mathcal{A}$ is chosen for the tail vertex of $\mathbf{e}_k^a\left(\vec{\Theta}\right)$ in the definition of $g_k^{ii}\left(\vec{\Theta}\right)$.

Applicability of a Type (c) C-function Two type (i) ACFs and two type (ii) ACFs are used to test the applicability of the type (c) C-function $f_{i,j}^c$ corresponding to edge i of $\mathcal{A}$ and edge j of $\mathcal{B}$. Let vertex i_1 of $\mathcal{A}$ be one of the endpoints of its ith edge and let $\mathbf{e}_i^a\left(\vec{\Theta}\right)$ be the edge vector along the ith edge with tail vertex i_1. Also let vertex j_1 of $\mathcal{B}$ be one of the endpoints of its jth edge and let $\mathbf{e}_j^b$ be the edge vector along edge j with tail vertex j_1. One type (i) ACF is defined for each of the two faces of $\mathcal{A}$, $1a$ and $2a$, which meet at edge i. Consider the two edge vectors $\mathbf{e}_{1a}^a\left(\vec{\Theta}\right)$ and $\mathbf{e}_{2a}^a\left(\vec{\Theta}\right)$ with tail vertex i_1 bounding faces $1a$ and $2a$, respectively. If $\mathbf{e}_{1a}^a\left(\vec{\Theta}\right) \times \mathbf{e}_i^a\left(\vec{\Theta}\right)$ points along $\mathbf{u}_{1a}^a\left(\vec{\Theta}\right)$, then $\mathbf{e}_{2a}^a\left(\vec{\Theta}\right) \times \mathbf{e}_i^a\left(\vec{\Theta}\right)$ points opposite to $\mathbf{u}_{2a}^a\left(\vec{\Theta}\right)$ and the two type (i) ACFs are

$$\begin{aligned} g_{1a}^i\left(\vec{\Theta}\right) &= \left\langle \mathbf{e}_j^b, +\mathbf{u}_{1a}^a\left(\vec{\Theta}\right)\right\rangle \\ g_{2a}^i\left(\vec{\Theta}\right) &= \left\langle \mathbf{e}_j^b, -\mathbf{u}_{2a}^a\left(\vec{\Theta}\right)\right\rangle \end{aligned}$$

If $\mathbf{e}_{1a}^a\left(\vec{\Theta}\right) \times \mathbf{e}_i^a\left(\vec{\Theta}\right)$ points opposite to $\mathbf{u}_{1a}^a\left(\vec{\Theta}\right)$, then $\mathbf{e}_{2a}^a\left(\vec{\Theta}\right) \times \mathbf{e}_i^a\left(\vec{\Theta}\right)$ points along $\mathbf{u}_{2a}^a\left(\vec{\Theta}\right)$ and the two type (i) ACFs are

$$\begin{aligned} g_{1a}^i\left(\vec{\Theta}\right) &= \left\langle \mathbf{e}_j^b, -\mathbf{u}_{1a}^a\left(\vec{\Theta}\right)\right\rangle \\ g_{2a}^i\left(\vec{\Theta}\right) &= \left\langle \mathbf{e}_j^b, +\mathbf{u}_{2a}^a\left(\vec{\Theta}\right)\right\rangle \end{aligned}$$

One type (ii) ACF is defined for each of the two faces of $\mathcal{B}$, $1b$ and $2b$, which meet at edge j. Consider the two edge vectors $\mathbf{e}_{1b}^b$ and $\mathbf{e}_{2b}^b$ with tail

vertex j_1 bounding faces $1b$ and $2b$, respectively. If $\mathbf{e}_j^b \times \mathbf{e}_{1b}^b$ points along $\mathbf{u}_{1b}^b$, then $\mathbf{e}_j^b \times \mathbf{e}_{2b}^b$ points opposite to $\mathbf{u}_{2b}^b$ and the two type (ii) ACFs are

$$\begin{aligned} g_{1b}^{ii}\left(\vec{\Theta}\right) &= \left\langle \mathbf{e}_i^a\left(\vec{\Theta}\right), +\mathbf{u}_{1b}^b \right\rangle \\ g_{2b}^{ii}\left(\vec{\Theta}\right) &= \left\langle \mathbf{e}_i^a\left(\vec{\Theta}\right), -\mathbf{u}_{2b}^b \right\rangle \end{aligned}$$

If $\mathbf{e}_j^b \times \mathbf{e}_{1b}^b$ points opposite to $\mathbf{u}_{1b}^b$, then $\mathbf{e}_j^b \times \mathbf{e}_{2b}^b$ points along $\mathbf{u}_{2b}^b$ and the two type (ii) ACFs are

$$\begin{aligned} g_{1b}^{ii}\left(\vec{\Theta}\right) &= \left\langle \mathbf{e}_i^a\left(\vec{\Theta}\right), -\mathbf{u}_{1b}^b \right\rangle \\ g_{2b}^{ii}\left(\vec{\Theta}\right) &= \left\langle \mathbf{e}_i^a\left(\vec{\Theta}\right), +\mathbf{u}_{2b}^b \right\rangle \end{aligned}$$

The type (c) C-function $f_{i,j}^c$ is applicable and has positive orientation at $\vec{\Theta}$ if

$$g_{1a}^{i}\left(\vec{\Theta}\right) \geq 0 \quad , \quad g_{2a}^{i}\left(\vec{\Theta}\right) \geq 0 \quad \text{and} \quad g_{1b}^{ii}\left(\vec{\Theta}\right) \leq 0 \quad , \quad g_{2b}^{ii}\left(\vec{\Theta}\right) \leq 0$$

It is applicable and has negative orientation at $\vec{\Theta}$ if

$$g_{1a}^{i}\left(\vec{\Theta}\right) \leq 0 \quad , \quad g_{2a}^{i}\left(\vec{\Theta}\right) \leq 0 \quad \text{and} \quad g_{1b}^{ii}\left(\vec{\Theta}\right) \geq 0 \quad , \quad g_{2b}^{ii}\left(\vec{\Theta}\right) \geq 0$$

2.5 The Third Collision Condition

As stated in section 2.3.3, the third collision condition is a check of the values of all the applicable C-functions at a configuration of interest. Those with positive orientation must all have nonpositive values and those with negative orientation must all have nonnegative values at a true collision point. Suppose that a configuration has been found for which the first and second collision conditions are satisfied. For such a configuration there is at least one applicable C-function whose value is zero.[1] The third collision condition must still be checked at such a configuration to determine whether it is a valid collision of $\mathcal{A}$ and $\mathcal{B}$. One way to check the third collision condition is to compute the values of all the other applicable C-functions, besides the ones having zero value, at such a configuration. This section

[1]Recall from section 2.3.3 that methods for finding such configurations will not be discussed in this chapter.

will discuss an alternate method for checking the third collision condition that does not require the values of the other applicable C-functions.

The third collision condition is used to insure that the features of $\mathcal{A}$ and $\mathcal{B}$ defining one of the applicable, zero-valued C-functions are touching. The new method for checking the third collision condition computes the actual position of each feature of $\mathcal{A}$ defining an applicable, zero-valued C-function to check that it is in contact with the corresponding feature of $\mathcal{B}$. This is equivalent to checking the third collision condition by computing the values of the other applicable C-functions since for any configuration of $\mathcal{A}$ that lies on the boundary of $\mathbf{CO}_{\mathcal{B}}^{\mathcal{A}}$, there is at least one applicable, zero-valued C-function whose defining features are in contact.

The conditions that determine when a feature of $\mathcal{A}$ is in contact with a feature of $\mathcal{B}$ depend on the type of C-function. These conditions are described in the following paragraphs for the type (a), type (b), and type (c) C-functions, respectively.

When the type (a) C-function $f_{i,j}^{a}$ is applicable and has zero value, the position vector of vertex j of $\mathcal{B}$, $\mathbf{b}_j$, must terminate on or inside of face i of $\mathcal{A}$ to satisfy the third collision condition. The vector $\mathbf{b}_j$ can be decomposed into a component in the plane P_i^a of face i of $\mathcal{A}$ and one normal to it. All of the position vectors for points in P_i^a have identical components normal to P_i^a. The vector $\mathbf{b}_j$ has this same component normal to P_i^a because $f_{i,j}^{a}$ is zero. The component of $\mathbf{b}_j$ in P_i^a itself will determine whether or not the vertex lies on or within the face of $\mathcal{A}$. Any face of $\mathcal{A}$ is a convex polygon. A point will lie on or inside of this polygon if it lies in the closed negative half-planes of all the lines containing edges of this polygon. This test can be implemented using inner products of the component of $\mathbf{b}_j$ in P_i^a with the outward-pointing normal vectors to the edges bounding face i of $\mathcal{A}$ rotated to the orientation of interest.

When the type (b) C-function $f_{i,j}^{b}$ is applicable and is equal to zero, the position vector of the vertex of $\mathcal{A}$ at the configuration of interest, $\mathbf{x}+\mathbf{a}_i\left(\vec{\Theta}\right)$, must terminate on or inside of face j of $\mathcal{B}$. The zero value of $f_{i,j}^{b}$ means that the component of $\mathbf{x}+\mathbf{a}_i\left(\vec{\Theta}\right)$ normal to plane P_j^b containing face j of $\mathcal{B}$ is such that it terminates in P_j^b. The component of $\mathbf{x}+\mathbf{a}_i\left(\vec{\Theta}\right)$ lying in P_j^b determines whether or not vertex i of $\mathcal{A}$ lies on or inside of the convex polygon representing face j of $\mathcal{B}$. This can be tested using inner products of the component of $\mathbf{x}+\mathbf{a}_i\left(\vec{\Theta}\right)$ lying in P_j^b with the outward-pointing normal

vectors to the edges bounding face j of $\mathcal{B}$.

If the type (c) C-function $f^c_{i,j}$ is both applicable and zero, edge i of $\mathcal{A}$ must have at least one point in common with edge j of $\mathcal{B}$ to satisfy the third collision condition. Since the value of $f^c_{i,j}$ is zero, the lines containing these two edges are coplanar at the configuration of interest. The location of the intersection point of these two (nonparallel) lines in the plane determines whether or not the two edges are in contact. When the two edges are in contact, this point lies in the segments of each line which is the edge of $\mathcal{A}$ or $\mathcal{B}$. The position vector of any point which lies on an edge of $\mathcal{A}$ or $\mathcal{B}$ can be expressed as the sum of the position of the tail vertex of that edge and a constant between zero and one times the edge vector itself. The third collision condition can be implemented by checking the inner product of the position vector of the intersection point minus the position vector of the tail vertex of each edge vector with the edge vector itself. The inner products must be between zero and one for both edge i of $\mathcal{A}$ and edge j of $\mathcal{B}$ for the edges to be in contact.

2.6 Type (c) Contacts with Aligned Edges

The type (a) or type (b) contacts involve a vertex of $\mathcal{B}$ or a vertex of $\mathcal{A}$, respectively. A vertex is a single point. Therefore, no special alignment can exist between this point and the plane containing a face of the other object. Thus, the type (a) and (b) C-functions are true measures of the minimum distances of the vertices of the objects from the planes containing the faces of the other object.

A type (c) contact involves an edge of $\mathcal{A}$ and an edge of $\mathcal{B}$. Because both of these edges are finite line segments, there exist configurations of $\mathcal{A}$ where these edges are aligned. As stated in section 2.2, "aligned" means that the lines containing these two edges are parallel. This includes the special case where the lines containing the two edges coincide. The type (c) C-functions are not always true measures of the minimum distances between the lines containing the two edges for these special configurations of $\mathcal{A}$.

The implications of edge-edge alignments on the use of C-functions to detect collisions of $\mathcal{A}$ and $\mathcal{B}$ are discussed below. This discussion includes a description of the behavior of the type (c) C-function and its ACFs when the edges are aligned. It will be shown that some of the type (a) and type

(b) C-functions can be used to detect contact between two aligned edges.

2.6.1 The Type (c) C-function for Aligned Edges

For an orientation $\vec{\Theta}$ where the lines containing edge i of $\mathcal{A}$ and edge j of $\mathcal{B}$ are parallel, the cross product of $\mathbf{e}_i^a\left(\vec{\Theta}\right)$ and $\mathbf{e}_j^b$ which appears in the definition of the type (c) C-function is the zero vector and $f_{i,j}^c = 0$. There is no unique plane which contains edge j of $\mathcal{B}$ and is parallel to edge i of $\mathcal{A}$ in this case. All of the infinitely many planes passing through edge j of $\mathcal{B}$ are parallel to edge i of $\mathcal{A}$. This includes the planes P_{1b}^b and P_{2b}^b containing faces $1b$ and $2b$ of $\mathcal{B}$ which meet at its jth edge. If the two edges are not also colinear, only one of the infinitely many planes passing through edge j of $\mathcal{B}$ also contains edge i of $\mathcal{A}$. In this case, the two edges are coplanar, but the value of $f_{i,j}^c$ does not measure the distance between the lines containing these two edges. If the two edges are colinear, any plane which passes through edge j of $\mathcal{B}$ must also pass through edge i of $\mathcal{A}$. The distance between the lines containing these two edges is zero and $f_{i,j}^c$ is a valid measure of this minimum distance.

Let the two endpoints of edge i of $\mathcal{A}$ be vertices i_1 and i_2 of $\mathcal{A}$ and let the two endpoints of edge j of $\mathcal{B}$ be vertices j_1 and j_2 of $\mathcal{B}$. When these two edges are aligned, all of the infinitely many planes passing through edge i of $\mathcal{A}$ are parallel to edge j of $\mathcal{B}$, including the planes P_{1a}^a and P_{2a}^a containing faces $1a$ and $2a$ of $\mathcal{A}$ which meet at edge i. The type (a) C-functions defined by face $1a$ and the two vertices j_1 and j_2, f_{1a,j_1}^a and f_{1a,j_2}^a, have the same value because all of the points along the line containing edge j of $\mathcal{B}$ are at the same distance from plane P_{1a}^a. These two C-functions measure the distance of edge j of $\mathcal{B}$ from P_{1a}^a along its outward-pointing normal vector $\mathbf{u}_{1a}^a\left(\vec{\Theta}\right)$. This distance will be nonzero if the two edges are not colinear. The type (a) C-functions defined by face $2a$ and vertices j_1 and j_2 of $\mathcal{B}$, f_{2a,j_1}^a and f_{2a,j_2}^a, also have the same value whenever edge i of $\mathcal{A}$ is aligned with edge j of $\mathcal{B}$. These two C-functions measure the distance of edge j of $\mathcal{B}$ from P_{2a}^a along its outward-pointing normal vector $\mathbf{u}_{2a}^a\left(\vec{\Theta}\right)$. This distance is also nonzero unless the two edges are colinear.

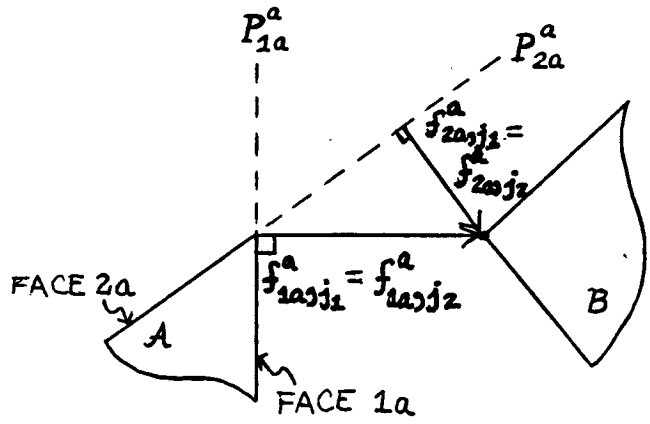
Now consider the type (b) C-functions $f_{i_1,1b}^b$ and $f_{i_2,1b}^b$ defined by vertices i_1 and i_2 of $\mathcal{A}$ and face $1b$ of $\mathcal{B}$. Because edge i of $\mathcal{A}$ is parallel to the plane P_{1b}^b, these two C-functions have the same value and measure the distance of edge i of $\mathcal{A}$ from P_{1b}^b along its outward-pointing normal vector $\mathbf{u}_{1b}^b$. Similarly,

the type (b) C-functions $f^b_{i_1,2b}$ and $f^b_{i_2,2b}$ defined by vertices i_1 and i_2 and face $2b$ have the same value. They measure the distance of edge i of $\mathcal{A}$ from plane P^b_{2b} along its outward-pointing normal vector $\mathbf{u}^b_{2b}$. The distance of edge i of $\mathcal{A}$ from planes P^b_{1b} and P^b_{2b} is nonzero whenever it is not colinear with edge j of $\mathcal{B}$.

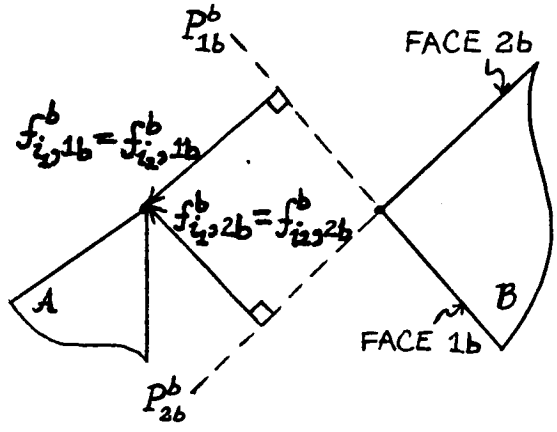
Thus, there are four type (a) and four type (b) C-functions which determine whether or not edge i of $\mathcal{A}$ is colinear with edge j of $\mathcal{B}$ even though the type (c) C-function $f^c_{i,j}$ itself no longer provides this information for this special configuration. When the two edges are colinear, all eight of these C-functions are zero along with $f^c_{i,j}$. The distances representing the values of these eight C-functions when edge i of $\mathcal{A}$ is aligned with edge j of $\mathcal{B}$ are indicated in Figure 2.13.

2.6.2 Applicability of the Type (c) C-function

The applicability tests were used above to determine if the two features defining the C-function can be brought into contact such that there exists a plane which separates $\mathcal{A}$ and $\mathcal{B}$. When two aligned edges are brought into contact, they are colinear and any separating plane must pass through the line containing both these edges. For some orientations of $\mathcal{A}$, there may be no such separating plane. For other orientations of $\mathcal{A}$, there will be one or more separating planes for $\mathcal{A}$ and $\mathcal{B}$. These possibilities are shown in Figure 2.14. Since there is no unique plane P^c_{ij} defined by the aligned edge vectors, there is no obvious candidate for the separating plane. The four ACFs defined for $f^c_{i,j}$ assuming P^c_{ij} was the unique candidate for the separating plane will not distinguish when the interiors of $\mathcal{A}$ and $\mathcal{B}$ overlap. In fact, they are all equal to zero regardless of whether $\mathcal{A}$ and $\mathcal{B}$ are overlapping or nonintersecting. This is most easily seen by looking at the original definition of the four ACFs involving inner products of the cross product vector $\mathbf{e}^a_i\left(\vec{\Theta}\right) \times \mathbf{e}^b_j$ with edge vectors of $\mathcal{A}$ or $\mathcal{B}$. Since this cross product is the zero vector when the edges are aligned, all four ACFs are zero. The type (i) and (ii) ACFs which replaced the original type (c) ACFs are also zero when the edges are aligned. The two type (i) ACFs are inner products of $\mathbf{e}^b_j$ with $\pm\mathbf{u}^a_{1a}\left(\vec{\Theta}\right)$ and $\pm\mathbf{u}^a_{2a}\left(\vec{\Theta}\right)$. By definition, $\mathbf{u}^a_{1a}\left(\vec{\Theta}\right)$ and $\mathbf{u}^a_{2a}\left(\vec{\Theta}\right)$ are perpendicular to edge i of $\mathcal{A}$. Since edge i is parallel to edge j, $\pm\mathbf{u}^a_{1a}\left(\vec{\Theta}\right)$ and $\pm\mathbf{u}^a_{2a}\left(\vec{\Theta}\right)$ are also perpendicular to $\mathbf{e}^b_j$ and the two



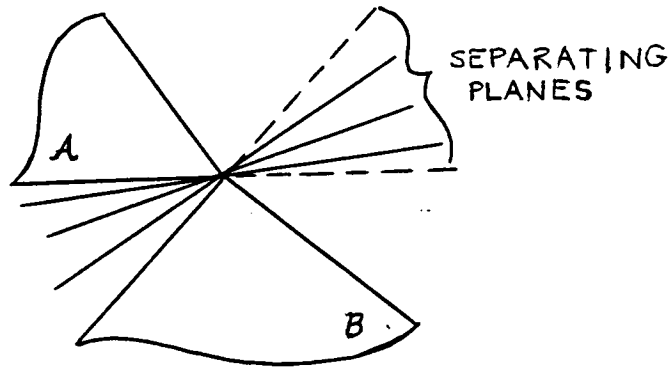
The four type (a) C-functions



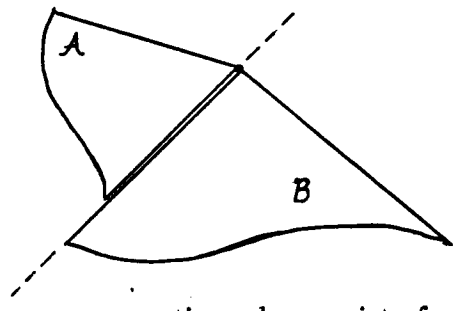
The four type (b) C-functions

Edge i of $\mathcal{A}$ and edge j of $\mathcal{B}$ are perpendicular to the page. Vertices i_1 and i_2 of $\mathcal{A}$ are depicted as a single point on $\mathcal{A}$. Vertices j_1 and j_2 are depicted as a single point on $\mathcal{B}$.

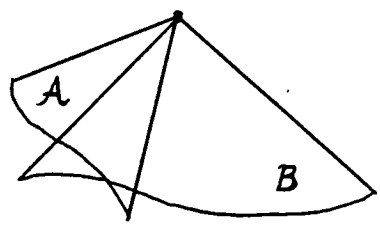
Figure 2.13: The type (a) and (b) C-functions for aligned edges



Multiple separating planes exist for this orientation



Only one separating plane exists for this orientation



No separating planes exist – $\mathcal{A}$ and $\mathcal{B}$ overlap for this orientation

Edge i of $\mathcal{A}$ and edge j of $\mathcal{B}$ are colinear and are perpendicular to the page.

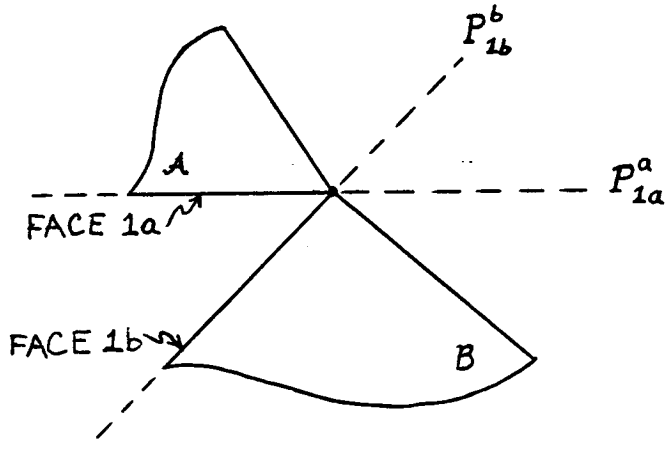
Figure 2.14: Separating planes for the type (c) C-function for the case of aligned edges

inner products are zero. The two type (ii) ACFs are inner products of $\mathbf{e}_i^a\left(\vec{\Theta}\right)$ with $\pm\mathbf{u}_{1b}^b$ and $\pm\mathbf{u}_{2b}^b$. But $\mathbf{u}_{1b}^b$ and $\mathbf{u}_{2b}^b$ are perpendicular to edge j of $\mathcal{B}$ which is parallel to edge i of $\mathcal{A}$. Therefore, $\mathbf{e}_i^a\left(\vec{\Theta}\right)$ is perpendicular to $\pm\mathbf{u}_{1b}^b$ and $\pm\mathbf{u}_{2b}^b$ and the two type (ii) ACFs are zero.

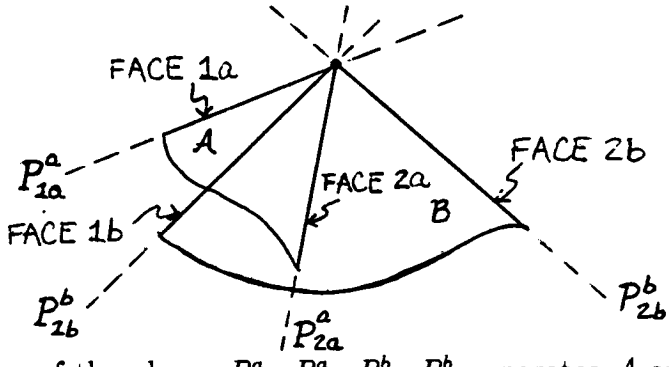
Four of the many potential separating planes for $\mathcal{A}$ and $\mathcal{B}$ which pass through the line containing edge i of $\mathcal{A}$ and edge j of $\mathcal{B}$ are the planes P_{1a}^a, P_{2a}^a, P_{1b}^b, and P_{2b}^b. Because of convexity, $\mathcal{A}$ lies entirely in the intersection of the closed negative half-spaces of P_{1a}^a and P_{2a}^a and $\mathcal{B}$ lies entirely in the intersection of the closed negative half-spaces of P_{1b}^b and P_{2b}^b. Part of $\mathcal{A}$ lies in the closed negative half-space of *both* P_{1b}^b and P_{2b}^b and part of $\mathcal{B}$ lies in the closed negative half-space of *both* P_{1a}^a and P_{2a}^a when the interiors of $\mathcal{A}$ and $\mathcal{B}$ overlap. If the interiors of $\mathcal{A}$ and $\mathcal{B}$ do not overlap, then $\mathcal{A}$ must lie entirely in the closed positive half-space of at least one of the planes P_{1b}^b and P_{2b}^b and $\mathcal{B}$ must lie in the closed positive half-space of at least one of the planes P_{1a}^a and P_{2a}^a. This means that one or both of the planes P_{1a}^a and P_{2a}^a as well as one or both of the planes P_{1b}^b and P_{2b}^b are separating planes for $\mathcal{A}$ and $\mathcal{B}$. Plane P_{1a}^a is the candidate separating plane for $\mathcal{A}$ and $\mathcal{B}$ in the tests of applicability for the type (a) C-functions f_{1a,j_1}^a and f_{1a,j_2}^a while P_{2a}^a is the candidate separating plane for $\mathcal{A}$ and $\mathcal{B}$ for the type (a) C-functions f_{2a,j_1}^a and f_{2a,j_2}^a. The candidate separating plane for $\mathcal{A}$ and $\mathcal{B}$ in the applicability tests for the type (b) C-functions $f_{i_1,1b}^b$ and $f_{i_2,1b}^b$ is P_{1b}^b and for the type (b) C-functions $f_{i_1,2b}^b$ and $f_{i_2,2b}^b$ the candidate separating plane is P_{2b}^b. At least two of these four type (a) C-functions and two of these four type (b) C-functions are applicable when the interiors of $\mathcal{A}$ and $\mathcal{B}$ do not overlap. This is illustrated in Figure 2.15. The applicability tests for these type (a) and (b) C-functions provide the information about overlap of $\mathcal{A}$ and $\mathcal{B}$ which cannot be determined using the applicability tests for $f_{i,j}^c$ at orientations where the edges are aligned.

2.6.3 The Third Collision Condition

The preceding sections showed how the four type (a) C-functions f_{1a,j_1}^a, f_{1a,j_2}^a, f_{2a,j_1}^a, and f_{2a,j_2}^a and the four type (b) C-functions $f_{i_1,1b}^b$, $f_{i_2,1b}^b$, $f_{i_1,2b}^b$, and $f_{i_2,2b}^b$ can be used to determine when edge i of $\mathcal{A}$ is colinear with edge j of $\mathcal{B}$ and when the interiors of $\mathcal{A}$ and $\mathcal{B}$ do not overlap for the given orientation of $\mathcal{A}$. Even if this is the case, it is still possible that there is



P^a_{1a} separates $\mathcal{A}$ and $\mathcal{B}$ $\Rightarrow$ f^a_{1a,j_1} and f^a_{1a,j_2} are applicable
P^b_{1b} separates $\mathcal{A}$ and $\mathcal{B}$ $\Rightarrow$ $f^b_{i_1,1b}$ and $f^b_{i_2,1b}$ are applicable



None of the planes P^a_{1a}, P^a_{2a}, P^b_{1b}, P^b_{2b} separates $\mathcal{A}$ and $\mathcal{B}$
None of the eight C-functions are applicable $\Rightarrow$ $\mathcal{A}$ and $\mathcal{B}$ are overlapping

Edge i of $\mathcal{A}$ and edge j of $\mathcal{B}$ are colinear and are perpendicular to the page.

Figure 2.15: Applicability of the type (a) and (b) C-functions for aligned edges

no contact between the edges of $\mathcal{A}$ and $\mathcal{B}$ because it is not known whether there are any points on the line containing the two edges which lie in both of the *finite* line segments that are the two edges. This is the reason for the third collision condition.

In this casee, the third collision condition can be implemented by computing the position of the features defining the type (a) and (b) C-functions as well as by checking the values of all the applicable C-functions at the configuration of interest. If these edges do have one or more points in common then at least one of the vertices i_1 and i_2 is touching the boundary of the two faces $1b$ and $2b$ of $\mathcal{B}$ or at least one of the vertices j_1 and j_2 is touching the boundary of the faces $1a$ and $2a$ of $\mathcal{A}$. The position vectors for any of these vertices at the configuration of interest will terminate on the boundary of the faces. If the edges do not have any points in common, then neither of the vertices i_1 and i_2 will be touching faces $1b$ and $2b$ of $\mathcal{B}$ and neither of the vertices j_1 and j_2 will be touching the faces $1a$ and $2a$ of $\mathcal{A}$. The corresponding position vectors will then not terminate on the boundary of the faces and the edges will not be in contact.

2.6.4 Detecting Collisions of Aligned Edges

Since the type (c) C-function and its ACFs do not provide any useful information at orientations where the two edges are aligned, $f^c_{i,j}$ can either be ignored or included in the check of the three collision conditions. For the remainder of this thesis, it is assumed that $f^c_{i,j}$ is applicable for *all* orientations where the two edges defining it are aligned and where all four of its ACFs as well as the function itself are equal to zero. With this assumption, the first two collision conditions are always satisfied at these orientations for any position $\mathbf{x}$. In addition $f^c_{i,j}$ satisfies the third collision condition regardless of the assumed orientation. Hence, the answer obtained by testing the three collision conditions on the remaining C-functions is not changed by taking $f^c_{i,j}$ to be one of the applicable C-functions. Actual collisions between the two aligned edges are detected from the four type (a) C-functions f^a_{1a,j_1}, f^a_{1a,j_2}, f^a_{2a,j_1}, and f^a_{2a,j_2} and the four type (b) C-functions $f^b_{i_1,1b}$, $f^b_{i_2,1b}$, $f^b_{i_1,2b}$, and $f^b_{i_2,2b}$. These C-functions are said to *subsume* the type (c) C-function at these orientations because they provide the information necessary to distinquish collisions between the aligned edges without checking $f^c_{i,j}$.

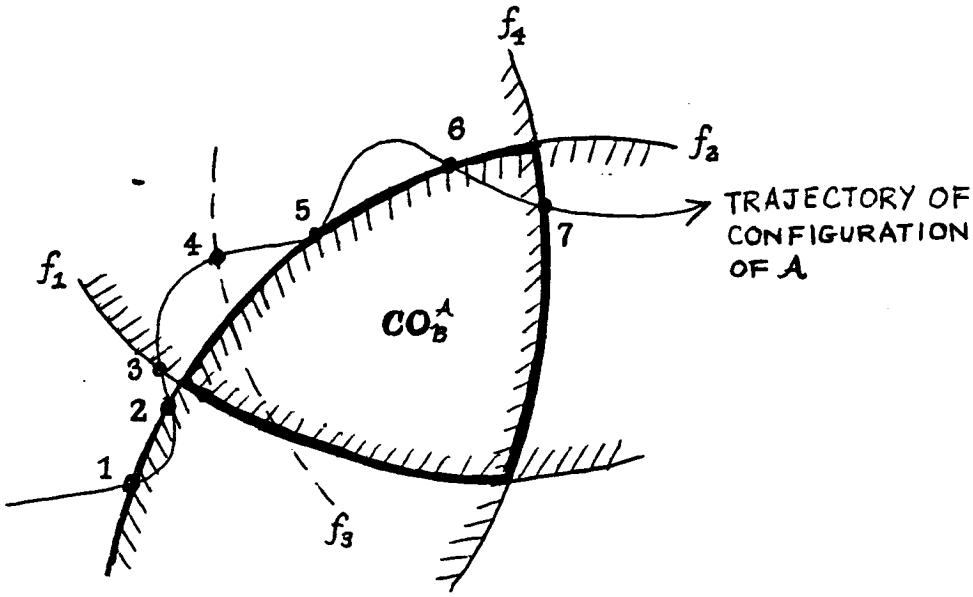
2.7 Allowable Zeros for C-functions

If the configuration of $\mathcal{A}$ is viewed dynamically as the trajectory of a moving point in $C_{space}(\mathcal{A})$, then there are two different ways in which the trajectory can intersect the boundary of $\mathbf{CO}_{\mathcal{B}}^{\mathcal{A}}$. In the first and only relevant case, the trajectory approaches the boundary of $\mathbf{CO}_{\mathcal{B}}^{\mathcal{A}}$ from free space. The first contact between $\mathcal{A}$ and $\mathcal{B}$ will occur when the trajectory of $\mathcal{A}$ first intersects the boundary of $\mathbf{CO}_{\mathcal{B}}^{\mathcal{A}}$ in this manner.

The second way that a trajectory of $\mathcal{A}$ can approach the boundary of $\mathbf{CO}_{\mathcal{B}}^{\mathcal{A}}$ is from a point inside of $\mathbf{CO}_{\mathcal{B}}^{\mathcal{A}}$. When the trajectory of $\mathcal{A}$ approaches the boundary of $\mathbf{CO}_{\mathcal{B}}^{\mathcal{A}}$ in this way, $\mathcal{A}$ is moving away from configurations where its interior overlaps with the interior of $\mathcal{B}$. This case does not represent a valid collision because it is not physically possible for the interiors of the rigid solids representing $\mathcal{A}$ and $\mathcal{B}$ to overlap. Unfortunately, a static check of the three collision conditions at some configuration of $\mathcal{A}$ which lies on the boundary of $\mathbf{CO}_{\mathcal{B}}^{\mathcal{A}}$ cannot distinguish between these two cases.

The applicable C-functions determine the boundary of $\mathbf{CO}_{\mathcal{B}}^{\mathcal{A}}$ and at least one of them is zero when the trajectory intersects $\mathbf{CO}_{\mathcal{B}}^{\mathcal{A}}$. Since a physically possible trajectory always begins outside of $\mathbf{CO}_{\mathcal{B}}^{\mathcal{A}}$, then at least one of these C-functions must approach zero from a region where its value is positive if it has positive orientation or from a region where its value is negative if it has negative orientation. For the purposes of collision detection, only the zeros of a C-function which occur when it is applicable and which also satisfy the above criterion represent potential collisions. These zeros are called the *allowable* zeros of the C-function. Some examples of allowable zeros are shown in Figure 2.16.

The definition of allowable zeros adds to the first two collision conditions a knowledge of the way in which the trajectory of $\mathcal{A}$ behaves as it approaches $\mathbf{CO}_{\mathcal{B}}^{\mathcal{A}}$. An allowable zero can occur when the trajectory intersects that part of the C-surface where the C-function is applicable but which *does not* lie in the appropriate half-spaces of the other applicable C-functions and hence is not part of the boundary of $\mathbf{CO}_{\mathcal{B}}^{\mathcal{A}}$. The third collision condition is still necessary to determine if an allowable zero represents a point on the boundary of $\mathbf{CO}_{\mathcal{B}}^{\mathcal{A}}$.



f_1, f_2, and f_4 are applicable C-functions
f_3 is a C-function which is not applicable

Function	f_1	f_2	f_3	f_4
Allowable zeros	3	1, 5, 6	–	–
Other zeros	–	2	4	7

Points 1 and 3 do not satisfy the third collision condition
Point 5 is the first contact between $\mathcal{A}$ and $\mathcal{B}$

Shading indicates the open half-space where an applicable C-function with positive orientation is negative or the open half-space where an applicable C-function with negative orientation is positive.

Figure 2.16: Allowable Zeros for C-functions

2.8 Configuration Space Obstacles for Nonconvex Polyhedra

The discussion of the configuration space obstacle in the preceding sections of this chapter has assumed that both $\mathcal{A}$ and $\mathcal{B}$ are represented by single *convex* polyhedra. This section considers the more general case where $\mathcal{A}$ and $\mathcal{B}$ are assumed to be *nonconvex* polyhedra.

Any nonconvex polyhedron can be modelled as the union of two or more convex polyhedra. If either the moving object $\mathcal{A}$ or the obstacle $\mathcal{B}$ is nonconvex, it can be represented as a union of convex polyhedra. This union is generally not unique because there can be more than one way to decompose a nonconvex polyhedron into convex pieces. Let $\mathcal{S}_a$ be a set of n_a *convex* polyhedra $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_{n_a}$ whose union is $\mathcal{A}$ and let $\mathcal{S}_b$ be a set of n_b *convex* polyhedra $\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_{n_b}$ whose union is $\mathcal{B}$.

This model of $\mathcal{A}$ or $\mathcal{B}$ as the union of the elements of a set of convex polyhedra can be extended to include the special case where either $\mathcal{A}$ or $\mathcal{B}$ is a single convex polyhedron. In this case there will be only one element in the set of convex polyhedra representing $\mathcal{A}$ or $\mathcal{B}$. If $\mathcal{A}$ is a convex polyhedron, $n_a = 1$ and $\mathcal{S}_a = \{\mathcal{A}_1\} \equiv \mathcal{A}$. Similarly, if $\mathcal{B}$ is a convex ployhedron, then $n_b = 1$ and $\mathcal{S}_b = \{\mathcal{B}_1\} \equiv \mathcal{B}$.

Each pair of convex polyhedra $\mathcal{A}_i \in \mathcal{S}_a$ and $\mathcal{B}_j \in \mathcal{S}_b$ generates a configuration space obstacle $\mathbf{CO}_{\mathcal{B}_j}^{\mathcal{A}_i}$ which can be characterized using the methods of the preceding sections. If the surface of $\mathcal{A}$ is just touching the surface of $\mathcal{B}$ then the surface of at least one of the convex polyhedra $\mathcal{A}_i$ is touching the surface of at least one of the convex polyhedra $\mathcal{B}_j$. This means that any configuration on the boundary of $\mathbf{CO}_{\mathcal{B}}^{\mathcal{A}}$ is on the boundary of at least one of the $\mathbf{CO}_{\mathcal{B}_j}^{\mathcal{A}_i}$. If the interiors of $\mathcal{A}$ and $\mathcal{B}$ overlap, then the interiors of at least one pair of convex polyhedra $\mathcal{A}_i$ and $\mathcal{B}_j$ are also overlapping. Any configuration inside of $\mathbf{CO}_{\mathcal{B}}^{\mathcal{A}}$ must also lie inside of at least one of the $\mathbf{CO}_{\mathcal{B}_j}^{\mathcal{A}_i}$. Therefore, the overall configuration space obstacle $\mathbf{CO}_{\mathcal{B}}^{\mathcal{A}}$ can be written as $\mathbf{CO}_{\mathcal{B}_1}^{\mathcal{A}_1} \cup \mathbf{CO}_{\mathcal{B}_2}^{\mathcal{A}_1} \ldots \cup \mathbf{CO}_{\mathcal{B}_{n_b}}^{\mathcal{A}_{n_a}}$. The first point where a trajectory of the configuration of $\mathcal{A}$ which starts in free space intersects the overall $\mathbf{CO}_{\mathcal{B}}^{\mathcal{A}}$ must be the first intersection point of the trajectory with any of the boundaries of the $\mathbf{CO}_{\mathcal{B}_j}^{\mathcal{A}_i}$ for the pairs of convex polyhedra.

2.9 A Conceptual Collision Detection Procedure

The concept of allowable zeros of a C-function and the third collision condition form the basis for a simple collision detection procedure. Following the notation of section 2.8, let the moving object $\mathcal{A}$ be represented by the set $\mathcal{S}_a$ of n_a *convex* polyhedra $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_{n_a}$ and let the obstacle $\mathcal{B}$ be represented by the set $\mathcal{S}_b$ of n_b *convex* polyhedra $\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_{n_b}$. The procedure considers each of the $n_a \cdot n_b$ pairs of convex polyhedra $\mathcal{A}_i \in \mathcal{S}_a$ and $\mathcal{B}_j \in \mathcal{S}_b$ separately. The steps listed below are performed for each of the C-functions defined by the features of $\mathcal{A}_i$ and $\mathcal{B}_j$.

First, all the values of time which represent allowable zeros of the C-functions along the trajectory of $\mathcal{A}$ are found. These values form a list $\{t'\}$ of potential collision points between the features of $\mathcal{A}_i$ and $\mathcal{B}_j$ which define each C-function. Then, for all t' in the list, the third collision condition is checked by using either of the two methods previously discussed. Any value of t' for which this condition is not satisfied is discarded since it cannot represent a valid collision of $\mathcal{A}_i$ and $\mathcal{B}_j$.

After these steps have been performed for all of the C-functions, all the lists of potential times of collision are sorted to find the one with the smallest value, t_{ij}. This is the time of first contact between $\mathcal{A}_i$ and $\mathcal{B}_j$. If no values of time remain in any of the lists, then no collision will occur between $\mathcal{A}_i$ and $\mathcal{B}_j$ for the given trajectory of $\mathcal{A}$.

The result of performing the above procedure for all of the pairs of convex polyhedra $\mathcal{A}_i$ and $\mathcal{B}_j$ is a list of the times of first contact $\{t_{ij}\}$. The final step in this collision detection procedure is to sort the t_{ij} for the one with the smallest value. This is the point where the polyhedron $\mathcal{A} \equiv \mathcal{A}_1 \cup \mathcal{A}_2 \cup \ldots \cup \mathcal{A}_{n_a}$ first comes into contact with the polyhedron $\mathcal{B} \equiv \mathcal{B}_1 \cup \mathcal{B}_2 \cup \ldots \cup \mathcal{B}_{n_b}$. If there are no values of time t_{ij} generated by any of the pairs of convex polyhedra making up $\mathcal{A}$ and $\mathcal{B}$, then no collision will occur between $\mathcal{A}$ and $\mathcal{B}$ along the given trajectory for $\mathcal{A}$.

In the special case where both $\mathcal{A}$ and $\mathcal{B}$ are single convex polyhedra, there is only one pair of convex polyhedra to be considered by the collision detection procedure. The only C-functions to be examined are then those defined by the features of $\mathcal{A}_1 \equiv \mathcal{A}$ and $\mathcal{B}_1 \equiv \mathcal{B}$. The smallest value of time, t_{11}, where all three collision conditions are satisfied is then the time of first

contact between $\mathcal{A}$ and $\mathcal{B}$. If there is no value of time t_{11} where all three collision conditions are satisfied, then $\mathcal{A}$ and $\mathcal{B}$ do not collide at any point along the given trajectory of $\mathcal{A}$.

One of the factors influencing the efficiency of the collision detection procedure described above is the number of convex polyhedra n_a in set $\mathcal{S}_a$ and n_b in set $\mathcal{S}_b$. As mentioned in section 2.8, when $\mathcal{A}$ or $\mathcal{B}$ is a nonconvex polyhedron, these sets are not unique. A nonconvex polyhedron can be divided into convex polyhedra in many ways. In particular, the convex polyhedra can be chosen such that they are overlapping or such that they are nonoverlapping. The collision detection procedure is valid for sets of either overlapping or nonoverlapping convex polyhedra whose union is $\mathcal{A}$ or $\mathcal{B}$. If the convex polyhedra are chosen to be overlapping, there is some redundancy involved in checking each pair of convex polyhedra separately. However, the overall efficiency of the procedure may improve if the representation using overlapping convex polyhedra results in fewer elements in $\mathcal{S}_a$ or $\mathcal{S}_b$ than a representation using nonoverlapping convex polyhedra. This is due to the reduction in the number of C-functions to be examined resulting from the smaller value for n_a or n_b.

One of the most important steps in the collision detection procedure outlined above is finding the allowable zeros of a C-function. This requires methods for finding the zeros of the C-function and for finding the regions where it is applicable along the trajectory of $\mathcal{A}$. Clearly, the details of any such method depends on the assumptions that are made regarding the motion of $\mathcal{A}$ relative to $\mathcal{B}$. Methods of finding the applicability regions and the zeros of the C-functions when $\mathcal{A}$ is moving with constant linear and angular velocities relative to $\mathcal{B}$ are developed in Chapters 3, 4, and 5. A complete collision detection algorithm is described in Chapter 6. This algorithm is similar to the conceptual procedure discussed in this section, but it incorporates some modifications that are possible only for the particular trajectory of $\mathcal{A}$ to be considered in the following chapters.

Chapter 3

Applicability for Constant Angular Velocity

The concept of applicability as it is first presented in section 2.3.1.2 refers to a fixed configuration, or more precisely a fixed orientation, of the moving object $\mathcal{A}$. Later in section 2.3.2, a more global view is taken with the introduction of applicability regions. The applicability region of a C-function is the set of all points in the rotation space of $\mathcal{A}$ for which the function is applicable. For these orientations of $\mathcal{A}$, there is no overlap of the interiors of $\mathcal{A}$ and $\mathcal{B}$ when $\mathcal{A}$ has been translated such that the features of $\mathcal{A}$ and $\mathcal{B}$ defining the C-function are in contact. The applicability regions are determined by the shapes of the two polyhedra representing $\mathcal{A}$ and $\mathcal{B}$. As the trajectory of $\mathcal{A}$ moves through $C_{space}(\mathcal{A})$ it will typically pass through many applicability regions for different C-functions. It may or may not pass through the region where a particular C-function is applicable.

The collision detection algorithm must search for points along the trajectory of $\mathcal{A}$ where a C-function has allowable zeros. Since by definition allowable zeros of a C-function can only occur when it is applicable, it would be advantageous to predict when the trajectory of $\mathcal{A}$ lies inside the applicability region of a given C-function. The search for zeros of the C-function could then be restricted to these parts of the trajectory of $\mathcal{A}$. If the trajectory of $\mathcal{A}$ were such that it always remained outside of the applicability region of a C-function, no zero search would be required for that function.

The ability to predict when the trajectory of $\mathcal{A}$ will pass through the applicability region for a C-function depends on knowledge of the rotational motion of $\mathcal{A}$. The applicability tests for the three types of C-functions developed in section 2.4 introduced the ACFs as functions of the orientation of $\mathcal{A}$. If the orientation of $\mathcal{A}$ as a function of time along the assumed trajectory is known, then the vectors representing the features of $\mathcal{A}$ can be expressed as vector-valued functions of time in reference coordinates. The ACFs can then be converted to scalar functions of time using these expressions for the edge vectors and face normals of $\mathcal{A}$ as functions time.

Those intervals of time when the values of the ACFs are such that the applicability tests are satisfied are the *applicability intervals* for this C-function. These are the only intervals where allowable zeros of this C-function may occur.

The remainder of this chapter will discuss how to find the applicability intervals for the case where $\mathcal{A}$ is rotating with constant angular velocity. The form of the type (i) and (ii) ACFs as functions of time is derived in section 3.1. A brief discussion of the special case of zero angular velocity, where $\mathcal{A}$ has fixed orientation, is included in this section. A graphical interpretation of the time behavior of the ACFs based on the behavior of the vectors representing the edges and faces of $\mathcal{A}$ and $\mathcal{B}$ is presented in section 3.2. Then, section 3.3 defines the applicability and orientation timelines that result from merging the ACFs for a C-function according to its applicability criteria. This section also gives an example of finding these two timelines for each of the three types of C-functions.

3.1 The ACFs for Constant Angular Velocity

In order to derive the form of the ACFs as functions of time, it is necessary to choose a means of expressing the orientation of $\mathcal{A}$, previously denoted by $\vec{\Theta}$. This thesis will use quaternions in all subsequent derivations, although identical results can be obtained using rotation matrices or Euler angles. First, a brief review of the use of quaternions to express the orientation of a rigid body will be given. The relationship between the elements of the quaternion and the yaw, pitch, and roll Euler angles is included in this review. Next, the equation for a vector representing a feature of $\mathcal{A}$ as a function of time is derived using the quaternions describing rotation of $\mathcal{A}$ with constant angular velocity. These vector-valued functions of time are then used to derive the form of the two types of ACFs as functions of time. Finally, the different cases for the time behavior of the ACFs are described.

3.1.1 Quaternions and Quaternion Algebra

A quaternion is a set of four parameters that can be used to describe the orientation of a rigid body relative to a reference coordinate system. These

four parameters can be represented as an ordered sequence of a scalar and a three-dimensional vector. The quaternion $\mathbf{Q} \in \Re^4$ will be denoted by

$$\mathbf{Q} = [\ q \quad \mathbf{q}^T\ ]^T$$

where q is the scalar representing the first element of the quaternion and $\mathbf{q}$ is a vector representing the last three elements of the quaternion.

Some properties of quaternion algebra that are useful in expressing rotations are listed below:

- The identity quaternion is $\mathbf{Q}_i = [\ 1 \quad \mathbf{0}^T\ ]^T$.
- The inverse of the quaternion $\mathbf{Q}$ will be denoted $\mathbf{Q}^{-1}$ and is defined to be $\mathbf{Q}^{-1} = [\ q \quad -\mathbf{q}^T\ ]^T$.
- The product of the two quaternions

  $$\mathbf{Q}_1 = [\ q_1 \quad \mathbf{q}_1^T\ ]^T \text{ and } \mathbf{Q}_2 = [\ q_2 \quad \mathbf{q}_2^T\ ]^T$$

  is defined to be the quaternion $\mathbf{Q}_3 = \mathbf{Q}_1\mathbf{Q}_2$ where

  $$\begin{aligned} \mathbf{Q}_3 &= [\ q_3 \quad \mathbf{q}_3^T\ ]^T \\ &= [\ q_1 q_2 - \langle \mathbf{q}_1, \mathbf{q}_2 \rangle \quad (q_1\mathbf{q}_2 + q_2\mathbf{q}_1 + \mathbf{q}_1 \times \mathbf{q}_2)^T\ ]^T \end{aligned}$$

- In general, the terms in a quaternion product do not commute so that $\mathbf{Q}_1\mathbf{Q}_2 \neq \mathbf{Q}_2\mathbf{Q}_1$.
- From the definition of quaternion products, it can be shown that

  $$\mathbf{Q}_3^{-1} = (\mathbf{Q}_1\mathbf{Q}_2)^{-1} = \mathbf{Q}_2^{-1}\mathbf{Q}_1^{-1}$$

- A unit quaternion $\mathbf{Q}_u$ is a quaternion whose elements q_u and $\mathbf{q}_u$ satisfy the condition $q_u^2 + \langle \mathbf{q}_u, \mathbf{q}_u \rangle = 1$.
- The product of a unit quaternion and its inverse quaternion is the identity quaternion and, in this special case, the terms in the product do commute. Hence, $\mathbf{Q}_u\mathbf{Q}_u^{-1} = \mathbf{Q}_u^{-1}\mathbf{Q}_u = \mathbf{Q}_i$.

Euler's theorem states that any sequence of rigid body rotations is equivalent to a single rotation about one axis. Let $\mathbf{r}$ be the unit vector along this single axis of rotation. The three components of $\mathbf{r}$ are the direction cosines of the axis of rotation relative to the three reference coordinate axes. Let θ represent the angle of rotation about the axis $\mathbf{r}$. Then, the quaternion

$$\mathbf{Q}_r = [\ \cos(\theta/2) \quad \sin(\theta/2)\mathbf{r}^T\ ]^T$$

can be used to express this rotation. $\mathbf{Q}_r$ is a unit quaternion because $\cos^2(\theta/2) + \sin^2(\theta/2)\langle \mathbf{r}, \mathbf{r}\rangle = \cos^2(\theta/2) + \sin^2(\theta/2) = 1$. The restriction that $\mathbf{Q}_r$ be a unit quaternion means that only three of its elements are independent. This reflects the fact that the general motion of a rigid body has only three rotational degrees of freedom.

Consider a vector from the center of rotation of the rigid body to one of the points on the body. Let $\mathbf{v}$ represent this vector expressed in reference coordinates *before* the rotation of the body about axis $\mathbf{r}$ through angle θ. If $\mathbf{v}'$ denotes this same vector expressed in reference coordinates *after* the rotation, then

$$\mathbf{v}' = \mathbf{Q}_r\,\mathbf{v}\mathbf{Q}_r^{-1}$$

When the vector $\mathbf{v}$ is treated as a quaternion having zero scalar component, this expression for $\mathbf{v}'$ becomes the product of three quaternions and the result can be computed using the definition of the product of quaternions given above.

A composite rotation resulting from a sequence of rotations can be expressed as a product of quaternions. For example, suppose that the rigid body is first rotated through an angle θ_1 about axis $\mathbf{r}_1$ and then through an angle θ_2 about axis $\mathbf{r}_2$. These two rotations are represented by the quaternions

$$\begin{aligned}\mathbf{Q}_{r_1} &= [\ \cos(\theta_1/2) \quad \sin(\theta_1/2)\,\mathbf{r}_1^T\ ]^T \\ \mathbf{Q}_{r_2} &= [\ \cos(\theta_2/2) \quad \sin(\theta_2/2)\,\mathbf{r}_2^T\ ]^T\end{aligned}$$

The composite rotation is represented by $\mathbf{Q}_{r_2}\mathbf{Q}_{r_1}$. Therefore, the vector initially expressed as $\mathbf{v}$ in reference coordinates becomes

$$\begin{aligned}\mathbf{v}' &= (\mathbf{Q}_{r_2}\mathbf{Q}_{r_1})\,\mathbf{v}(\mathbf{Q}_{r_2}\mathbf{Q}_{r_1})^{-1} \\ &= (\mathbf{Q}_{r_2}\mathbf{Q}_{r_1})\,\mathbf{v}\left(\mathbf{Q}_{r_1}^{-1}\mathbf{Q}_{r_2}^{-1}\right)\end{aligned}$$

after these two rotations of the rigid body.

Three parameters that are frequently used to describe the orientation of a spacecraft are the yaw, pitch, and roll Euler angles. Any orientation represented by Euler angles is viewed as the result of three sequential rotations from the initial orientation where the body coordinate axes of the rigid body representing the spacecraft are parallel to the corresponding reference axes. For the yaw, pitch, roll Euler angle sequence, the first rotation is a rotation about the $Z_a = Z$ axis through the yaw angle θ_y. The second rotation is a rotation about the Y_a body axis through the pitch angle θ_p. As a result of the first rotation, this axis may no longer be aligned with the Y axis. The third and final rotation is a rotation about the X_a body axis through the roll angle θ_r. Again, the X_a body axis is, in general, no longer aligned with any of the reference axes after the first and second rotations. The elements of the quaternion $\mathbf{Q}_{ypr}$ that represents the composite rotation resulting from these three sequential rotations are given, without proof, below:

$$q_{ypr} = +\sin(\theta_y/2)\sin(\theta_p/2)\sin(\theta_r/2) + \cos(\theta_y/2)\cos(\theta_p/2)\cos(\theta_r/2)$$

and

$$\mathbf{q}_{ypr} = \begin{bmatrix} -\sin(\theta_y/2)\sin(\theta_p/2)\cos(\theta_r/2) + \cos(\theta_y/2)\cos(\theta_p/2)\sin(\theta_r/2) \\ +\sin(\theta_y/2)\cos(\theta_p/2)\sin(\theta_r/2) + \cos(\theta_y/2)\sin(\theta_p/2)\cos(\theta_r/2) \\ +\sin(\theta_y/2)\cos(\theta_p/2)\cos(\theta_r/2) - \cos(\theta_y/2)\sin(\theta_p/2)\sin(\theta_r/2) \end{bmatrix}$$

It will be convenient to express the initial orientation of $\mathcal{A}$ for the examples presented later in this chapter and in Chapters 6 and 7 by giving the yaw, pitch, and roll angles, θ_y, θ_p, and θ_r, instead of listing the elements of the corresponding quaternion.

More detailed treatments of quaternions and other methods of representing rigid body rotations can be found in references [19,20,21].

3.1.2 Expressing the Vectors Representing the Features of $\mathcal{A}$ as Functions of Time

Let $\mathbf{Q}_o$ be the quaternion representing the initial orientation of $\mathcal{A}$, that is the orientation of $\mathcal{A}$ when $t = 0$. Also let $\vec{\omega}$ be the constant vector representing the angular velocity of $\mathcal{A}$. Then the angular speed ω is the magnitude

of this vector and the unit vector $\mathbf{u}_\omega$ lies along the axis of rotation where

$$\omega = \|\vec{\omega}\| \quad , \quad \mathbf{u}_\omega = \frac{\vec{\omega}}{\omega}$$

Since $\vec{\omega}$ is constant, the quaternion $\mathbf{Q}_\omega(t)$ representing rotation of $\mathcal{A}$ from its initial orientation as a function of time is

$$\mathbf{Q}_\omega(t) = [\ \cos(\omega t/2) \quad \sin(\omega t/2)\,\mathbf{u}_\omega^T\ ]^T$$

The quantity ωt is the angle of rotation about the axis $\mathbf{u}_\omega$. The quaternion $\mathbf{Q}(t)$ representing the overall orientation of $\mathcal{A}$ as a function of time is simply the product of the two quaternions $\mathbf{Q}_\omega(t)$ and $\mathbf{Q}_o$, $\mathbf{Q}(t) = \mathbf{Q}_\omega(t)\,\mathbf{Q}_o$.

Any vector $\mathbf{v}^a$ representing a feature of $\mathcal{A}$ can be written as a function of time using the above equation for $\mathbf{Q}(t)$. From the preceding section,

$$\mathbf{v}^a(t) = \mathbf{Q}(t)\,\mathbf{v}^a\,\mathbf{Q}(t)^{-1} = \mathbf{Q}_\omega(t)\left(\mathbf{Q}_o\,\mathbf{v}^a\,\mathbf{Q}_o{}^{-1}\right)\mathbf{Q}_\omega(t)^{-1}$$

The term in parentheses is a constant vector which represents the vector $\mathbf{v}^a$ expressed in reference coordinates at the initial time. If this vector is denoted $\mathbf{v}^a(0)$, then

$$\mathbf{v}^a(t) = \mathbf{Q}_\omega(t)\,\mathbf{v}^a(0)\,\mathbf{Q}_\omega(t)^{-1}$$

Performing the indicated quaternion multiplications treating $\mathbf{v}^a(0)$ as a quaternion with zero scalar component gives

$$\mathbf{v}^a(t) = \\ [\langle \mathbf{v}^a(0),\, \mathbf{u}_\omega\rangle\ \mathbf{u}_\omega] + [\mathbf{v}^a(0) - \langle \mathbf{v}^a(0),\, \mathbf{u}_\omega\rangle\ \mathbf{u}_\omega]\cos(\omega t) + [\mathbf{u}_\omega \times \mathbf{v}^a(0)]\sin(\omega t)$$

The three vectors in square brackets are constant vectors. Let the first constant vector be

$$\mathbf{v}_1^a = \langle \mathbf{v}^a(0),\, \mathbf{u}_\omega\rangle\ \mathbf{u}_\omega$$

This is just the component of $\mathbf{v}^a(0)$ along the rotation axis $\mathbf{u}_\omega$. Let the vector which is multiplied by the cosine term be

$$\mathbf{v}_2^a = \mathbf{v}^a(0) - \langle \mathbf{v}^a(0),\, \mathbf{u}_\omega\rangle\ \mathbf{u}_\omega = \mathbf{v}^a(0) - \mathbf{v}_1^a$$

This is the component of $\mathbf{v}^a(0)$ normal to the rotation axis $\mathbf{u}_\omega$. Finally, let the vector which is multiplied by the sine term be

$$\mathbf{v}_3^a = \mathbf{u}_\omega \times \mathbf{v}^a(0)$$

This vector is normal to both $\mathbf{v}^a(0)$ and the rotation axis $\mathbf{u}_\omega$. Except for the special case where $\mathbf{v}^a(0)$ lies along the rotation axis, $\mathbf{v}_1^a$, $\mathbf{v}_2^a$, and $\mathbf{v}_3^a$ form an orthogonal triad. The vector $\mathbf{v}^a$ as a function of time is

$$\mathbf{v}^a(t) = \mathbf{v}_1^a + \mathbf{v}_2^a \cos(\omega t) + \mathbf{v}_3^a \sin(\omega t)$$

When $\mathbf{v}^a(0)$ lies along the rotation axis, both $\mathbf{v}_2^a$ and $\mathbf{v}_3^a$ are zero and $\mathbf{v}^a = \mathbf{v}_1^a = \mathbf{v}^a(0)$ is constant.

3.1.3 Derivation of the Type (i) and (ii) ACFs as Functions of Time

A type (i) ACF is the inner product of an edge vector of $\mathcal{B}$ and either the outward-pointing normal to a face of $\mathcal{A}$ or its negative. Assume without loss of generality that a particular ACF g^i is defined using the plus sign for the outward-pointing normal vector to the face of $\mathcal{A}$ so that

$$g^i\left(\vec{\Theta}\right) = \left\langle \mathbf{e}^b, \mathbf{u}^a\left(\vec{\Theta}\right)\right\rangle$$

The face normal vector $\mathbf{u}^a$ can be written as the sum of three terms following the procedure described above. Let $\mathbf{u}^a(0) = \mathbf{Q}_o\, \mathbf{u}^a\, \mathbf{Q}_o{}^{-1}$ be $\mathbf{u}^a$ expressed in reference coordinates at the initial time. Then the outward-pointing normal vector $\mathbf{u}^a$ as a function of time is

$$\mathbf{u}^a(t) = \mathbf{u}_1^a + \mathbf{u}_2^a \cos(\omega t) + \mathbf{u}_3^a \sin(\omega t)$$

where

$$\begin{aligned} \mathbf{u}_1^a &= \langle \mathbf{u}^a(0), \mathbf{u}_\omega \rangle\, \mathbf{u}_\omega \\ \mathbf{u}_2^a &= \mathbf{u}^a(0) - \mathbf{u}_1^a \\ \mathbf{u}_3^a &= \mathbf{u}_\omega \times \mathbf{u}^a(0) \end{aligned}$$

Substituting this into the definition of the type (i) ACF gives the following expression for g^i as a function of time

$$g^i(t) = \left\langle \mathbf{e}^b, \mathbf{u}_1^a\right\rangle + \left\langle \mathbf{e}^b, \mathbf{u}_2^a\right\rangle \cos(\omega t) + \left\langle \mathbf{e}^b, \mathbf{u}_3^a\right\rangle \sin(\omega t)$$

The three inner products appearing in this equation are all constant.

A type (ii) ACF is the inner product of an edge vector of $\mathcal{A}$ and either the outward-pointing normal to a face of $\mathcal{B}$ or its negative. Again,

assume that a particular ACF g^{ii} is define using the plus sign in front of the outward-pointing normal vector to the face of $\mathcal{B}$ so that

$$g^{ii}\left(\vec{\Theta}\right) = \left\langle \mathbf{e}^a\left(\vec{\Theta}\right), \mathbf{u}^b \right\rangle$$

If $\mathbf{e}^a(0) = \mathbf{Q}_o\, \mathbf{e}^a\, \mathbf{Q}_o{}^{-1}$ is the edge vector $\mathbf{e}^a$ expressed in reference coordinates at the initial time, then $\mathbf{e}^a$ as a function of time is

$$\mathbf{e}^a(t) = \mathbf{e}^a_1 + \mathbf{e}^a_2 \cos(\omega t) + \mathbf{e}^a_3 \sin(\omega t)$$

where

$$\begin{aligned} \mathbf{e}^a_1 &= \langle \mathbf{e}^a(0), \mathbf{u}_\omega \rangle\, \mathbf{u}_\omega \\ \mathbf{e}^a_2 &= \mathbf{e}^a(0) - \mathbf{e}^a_1 \\ \mathbf{e}^a_3 &= \mathbf{u}_\omega \times \mathbf{e}^a(0) \end{aligned}$$

Substituting this into the definition of the type (ii) ACF yields

$$g^{ii}(t) = \left\langle \mathbf{e}^a_1, \mathbf{u}^b \right\rangle + \left\langle \mathbf{e}^a_2, \mathbf{u}^b \right\rangle \cos(\omega t) + \left\langle \mathbf{e}^a_3, \mathbf{u}^b \right\rangle \sin(\omega t)$$

The three inner products appearing in this equation are also constant.

Now suppose that $\mathcal{A}$ is not rotating so that $\vec{\omega} = \mathbf{0}$. Then the vectors representing features of $\mathcal{A}$ do not vary with time. Let $\mathbf{Q}_o$ denote the fixed orientation of $\mathcal{A}$. The outward-pointing normal vector $\mathbf{u}^a$ for any face of $\mathcal{A}$ is simply the constant vector $\mathbf{u}^a(t) = \mathbf{u}^a(0) = \mathbf{Q}_o\, \mathbf{u}^a\, \mathbf{Q}_o{}^{-1}$. The type (i) ACF g^i as a function of time is

$$g^i(t) = \left\langle \mathbf{e}^b, \mathbf{u}^a(0) \right\rangle$$

Similarly, the edge vector $\mathbf{e}^a$ for any edge of $\mathcal{A}$ is the constant vector $\mathbf{e}^a(t) = \mathbf{e}^a(0) = \mathbf{Q}_o\, \mathbf{e}^a\, \mathbf{Q}_o{}^{-1}$ when $\vec{\omega} = \mathbf{0}$. The type (ii) ACF g^{ii} as a function of time is

$$g^{ii}(t) = \left\langle \mathbf{e}^a(0), \mathbf{u}^b \right\rangle$$

In this case, both g^i and g^{ii} are constants.

Thus, when $\vec{\omega}$ is constant, the two types of ACFs have identical forms as functions of time. Let g denote a single ACF of either type (i) or type (ii). The general form of g can be written as

$$g(t) = g_1 + g_2 \cos(\omega t) + g_3 \sin(\omega t)$$

Table 3.1 lists the values of the scalar coefficients g_1, g_2, and g_3 for the two types of ACFs when $\vec{\omega}$ is constant.

$g(t) = g_1 + g_2 \cos(\omega t) + g_3 \sin(\omega t)$			
Type of ACF	g_1	g_2	g_3
$\vec{\omega} \neq 0$			
(i)	$\langle \mathbf{e}^b, \mathbf{u}_1^a \rangle$	$\langle \mathbf{e}^b, \mathbf{u}_2^a \rangle$	$\langle \mathbf{e}^b, \mathbf{u}_3^a \rangle$
(ii)	$\langle \mathbf{e}_1^a, \mathbf{u}^b \rangle$	$\langle \mathbf{e}_2^a, \mathbf{u}^b \rangle$	$\langle \mathbf{e}_3^a, \mathbf{u}^b \rangle$
$\vec{\omega} = 0$			
(i)	$\langle \mathbf{e}^b, \mathbf{u}^a(0) \rangle$	0	0
(ii)	$\langle \mathbf{e}^a(0), \mathbf{u}^b \rangle$	0	0

Table 3.1: The constant coefficients for the two types of ACFs as functions of time

3.1.4 The Behavior of an ACF as a Function of Time

There are two possibilities for the behavior of $g(t)$ as a function of time. One possibility is that the coefficients of the two sinusoidal terms are both zero, that is $g_2 = g_3 = 0$. Then the ACF is equal to the constant term g_1, which may be positive, negative, or zero. There are three ways that this can occur. First, the angular velocity of $\mathcal{A}$ may be zero. Second, if the angular velocity is nonzero, the vector representing the edge or face normal of $\mathcal{A}$ may lie along the axis of rotation. In both of these situations the vectors representing features of $\mathcal{A}$ are constant so that the inner products defining the ACFs are also constant. Finally, the vector representing the edge or face normal of $\mathcal{B}$ may lie along the axis of rotation. In this case, this vector is perpendicular to the two components of the edge vector or face normal of $\mathcal{A}$ which appear in the inner products defining g_2 and g_3.

The second possibility for the behavior of $g(t)$ is that one or both of the coefficients of the two sinusoidal terms is nonzero, that is either $g_2 \neq 0$ or $g_3 \neq 0$ or both. The presence of the sinusoidal term means that $g(t)$ is a periodic function of time with period $2\pi/\omega$. The last two terms in the equation for $g(t)$ can then be combined into a single sinsoidal term as

$$g(t) = g_1 + G \cos(\omega t + \phi_g)$$

where

$$\begin{aligned} G &= \sqrt{(g_2)^2 + (g_3)^3} \quad , \quad G > 0 \\ \phi_g &= \tan^{-1}\left(\frac{-g_3}{g_2}\right) \quad , \quad \phi_g \in [0, 2\pi) \end{aligned}$$

G is the amplitude of the cosine wave. The constant term g_1 represents a vertical shift of the cosine wave from its original symmetry around zero. The phase angle ϕ_g determines the horizontal shift of the cosine wave along the time axis.

The shifted sinusoidal form of $g(t)$ makes it easy to find the points, if any, where $g(t) = 0$. These are the points where $g(t)$ changes sign and they bound the regions where its value is positive or negative. For convenience, $g(t)$ will be regarded as a periodic function of $\omega t + \phi_g$ in the following discussion.

There are three cases for the behavior of $g(t)$ depending on the magnitude of the vertical shift g_1 relative to the amplitude of the cosine term G. These three cases are illustrated in Figure 3.1. First, if $\|g_1/G\| \geq 1$ and $g_1 \leq -G < 0$, then the value of the ACF $g(t)$ is always less than or equal to zero. In other words, $g(t)$ is nonpositive for the entire period $\omega t + \phi_g \in [0, 2\pi]$. When $g_1 < -G$, $g(t)$ is strictly negative. For the case where $g_1 = -G$, $g(t)$ equals zero only at the point where $\cos(\omega t + \phi_g) = +1$ which is where $\omega t + \phi_g = 0$. The value of $g(t)$ is less than zero at all other values of $\omega t + \phi_g$.

Now suppose that $\|g_1/G\| \geq 1$ but $g_1 \geq G > 0$. Then the value of the ACF is always greater than or equal to zero. Hence, the function $g(t)$ is nonnegative for the entire period $\omega t + \phi_g \in [0, 2\pi]$. When $g_1 > G$, $g(t)$ is strictly positive. If $g_1 = G$, then $g(t)$ is equal to zero only at the point where $\cos(\omega t + \phi_g) = -1$ which is where $\omega t + \phi_g = \pi$. The value of $g(t)$ is greater than zero at all other values of $\omega t + \phi_g$.

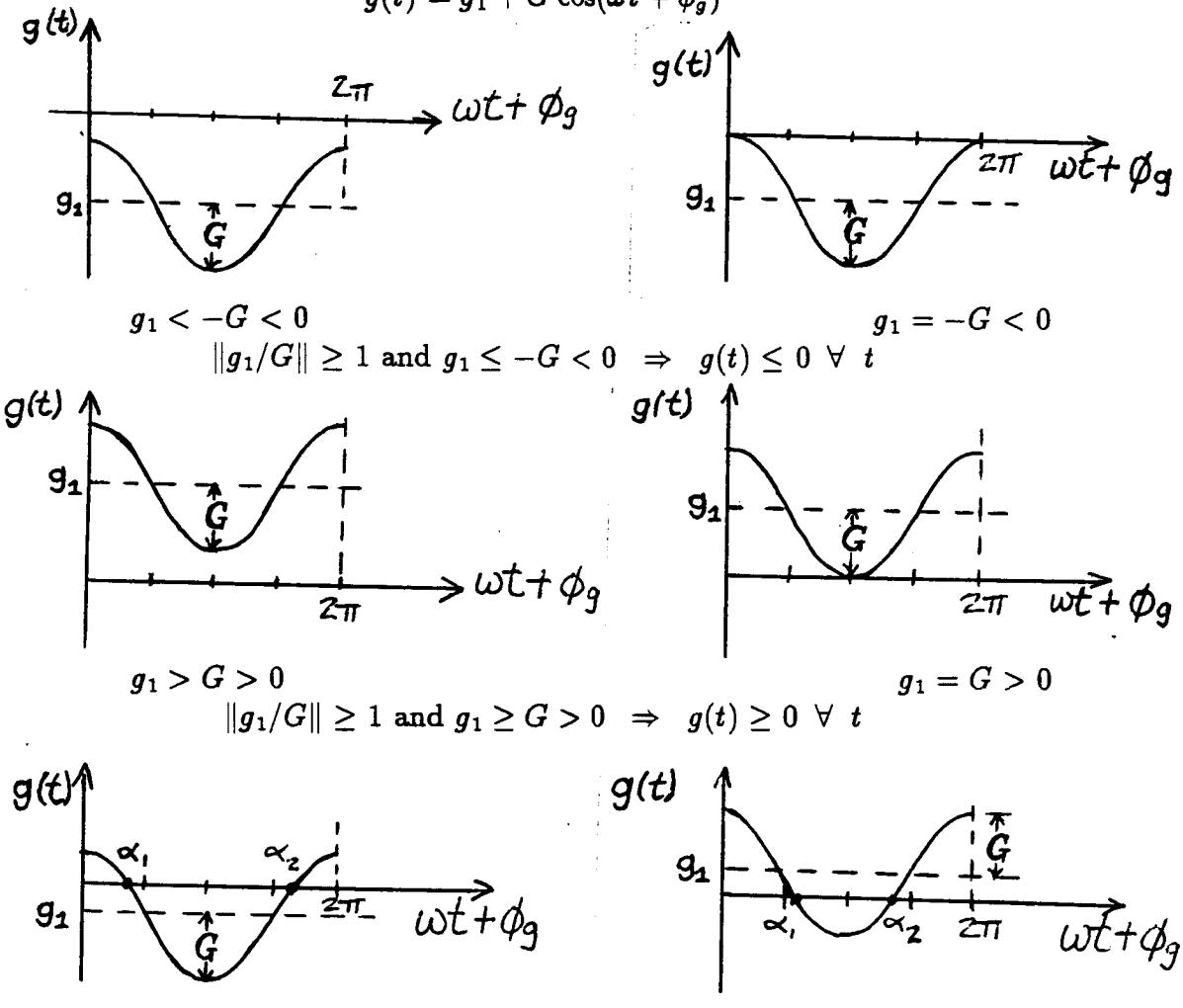
Finally, if $\|g_1/G\| < 1$, then the ACF takes on both positive and negative values in a single period of $\omega t + \phi_g$. The value of $g(t)$ is zero at the two points where

$$\cos(\omega t + \phi_g) = \frac{-g_1}{G}$$

or where

$$\omega t + \phi_g = \begin{cases} \cos^{-1}\left(\frac{-g_1}{G}\right) \\ 2\pi - \cos^{-1}\left(\frac{-g_1}{G}\right) \end{cases}$$

$$g(t) = g_1 + G\cos(\omega t + \phi_g)$$



$g_1 < -G < 0$ $\qquad$ $g_1 = -G < 0$

$\|g_1/G\| \geq 1$ and $g_1 \leq -G < 0 \;\Rightarrow\; g(t) \leq 0 \;\forall\; t$

$g_1 > G > 0$ $\qquad$ $g_1 = G > 0$

$\|g_1/G\| \geq 1$ and $g_1 \geq G > 0 \;\Rightarrow\; g(t) \geq 0 \;\forall\; t$

$-G < g_1 < 0$ $\qquad$ $G > g_1 > 0$

$\|g_1/G\| < 1 \;\Rightarrow\; g(t)$ changes sign

Figure 3.1: The three cases for $g(t)$ when $G \neq 0$

Let these two points be called α_1 and α_2 where $\alpha_2 > \alpha_1$ and $\alpha_1, \alpha_2 \in (0, 2\pi)$. The ACF is nonnegative in the intervals $\omega t + \phi_g \in [0, \alpha_1]$ and $\omega t + \phi_g \in [\alpha_2, 2\pi]$. It is nonpositive in the interval $\omega t + \phi_g \in [\alpha_1, \alpha_2]$.

3.2 A Graphical Interpretation of the ACFs

The sinusoidal nature of the ACFs can be predicted by visualizing the behavior of the vectors representing the edges and faces of $\mathcal{A}$ when $\mathcal{A}$ is rotating with constant angular velocity. This section presents a qualitative description of the ACFs based on this graphical interpretation of the vectors used in their definitions. Quantitative information needed for the applicability tests, such as the values of ωt where an ACF changes sign, is obtained from the equations for the ACFs derived in the preceding section. Therefore, this section may be skipped if the reader is satisfied with the mathematical treatment of the ACFs already presented.

In section 3.1.2, the time behavior of any vector $\mathbf{v}^a$ representing a feature of $\mathcal{A}$ was shown to take the form

$$\mathbf{v}^a(t) = \mathbf{v}_1^a + \mathbf{v}_2^a \cos(\omega t) + \mathbf{v}_3^a \sin(\omega t)$$

where the three constant vectors $\mathbf{v}_1^a$, $\mathbf{v}_2^a$, and $\mathbf{v}_3^a$ are

$$\begin{aligned} \mathbf{v}_1^a &= \langle \mathbf{v}^a(0), \mathbf{u}_\omega \rangle \, \mathbf{u}_\omega \\ \mathbf{v}_2^a &= \mathbf{v}^a(0) - \mathbf{v}_1^a \\ \mathbf{v}_3^a &= \mathbf{u}_\omega \times \mathbf{v}^a(0) \end{aligned}$$

Consider the most general case where all three component vectors for $\mathbf{v}^a$ are nonzero and mutually orthogonal. The first vector $\mathbf{v}_1^a$ lies along the rotation axis and contributes a constant component to $\mathbf{v}^a(t)$. The contribution of the second and third vectors, $\mathbf{v}_2^a$ and $\mathbf{v}_3^a$, to $\mathbf{v}^a(t)$ varies with time. These two vectors lie in the plane whose normal vector is $\mathbf{u}_\omega$. Furthermore, the magnitudes of these two component vectors are equal. This can be proven by comparing the two inner products

$$\begin{aligned} \|\mathbf{v}_2^a\|^2 &= \langle \mathbf{v}_2^a, \mathbf{v}_2^a \rangle \\ \|\mathbf{v}_3^a\|^2 &= \langle \mathbf{v}_3^a, \mathbf{v}_3^a \rangle \end{aligned}$$

The first inner product is

$$\begin{aligned} \|\mathbf{v}_2^a\|^2 &= \langle (\mathbf{v}^a(0) - \mathbf{v}_1^a), (\mathbf{v}^a(0) - \mathbf{v}_1^a) \rangle \\ &= \langle \mathbf{v}^a(0), \mathbf{v}^a(0) \rangle - 2 \langle \mathbf{v}^a(0), \mathbf{v}_1^a \rangle + \langle \mathbf{v}_1^a, \mathbf{v}_1^a \rangle \end{aligned}$$

Using the definition of $\mathbf{v}_1^a$ to evaluate the inner products on the right-hand side of the above equation and combining similar terms gives

$$\|\mathbf{v}_2^a\|^2 = \langle \mathbf{v}^a(0), \mathbf{v}^a(0)\rangle - \langle \mathbf{v}^a(0), \mathbf{u}_\omega\rangle^2$$

The second inner product is

$$\|\mathbf{v}_3^a\|^2 = \langle(\mathbf{u}_\omega \times \mathbf{v}^a(0)), (\mathbf{u}_\omega \times \mathbf{v}^a(0))\rangle$$

This can be rewritten using the scalar triple product identity $\langle \mathbf{a}, \mathbf{b}\times \mathbf{c}\rangle = \langle \mathbf{b}, \mathbf{c}\times \mathbf{a}\rangle$ where $(\mathbf{u}_\omega \times \mathbf{v}^a(0))$ takes the place of $\mathbf{a}$, $\mathbf{u}_\omega$ takes the place of $\mathbf{b}$, and $\mathbf{v}^a(0)$ takes the place of $\mathbf{c}$. This produces

$$\|\mathbf{v}_3^a\|^2 = \langle \mathbf{u}_\omega, \mathbf{v}^a(0) \times (\mathbf{u}_\omega \times \mathbf{v}^a(0))\rangle$$

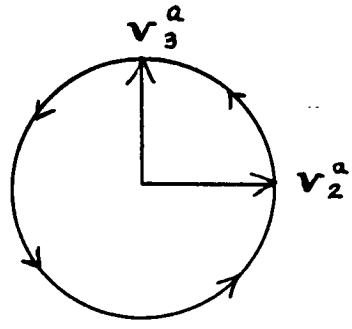
The second term in the above inner product is a vector triple product. It can be simplified using the vector triple identity $\mathbf{a}\times(\mathbf{b}\times \mathbf{c}) = \langle \mathbf{a}, \mathbf{c}\rangle \mathbf{b} - \langle \mathbf{a}, \mathbf{b}\rangle \mathbf{c}$. The result is

$$\begin{aligned}\|\mathbf{v}_3^a\|^2 &= \langle \mathbf{u}_\omega, \{\langle \mathbf{v}^a(0), \mathbf{v}^a(0)\rangle\, \mathbf{u}_\omega - \langle \mathbf{v}^a(0), \mathbf{u}_\omega\rangle\, \mathbf{v}^a(0)\}\rangle \\ &= \langle \mathbf{v}^a(0), \mathbf{v}^a(0)\rangle - \langle \mathbf{v}^a(0), \mathbf{u}_\omega\rangle^2\end{aligned}$$

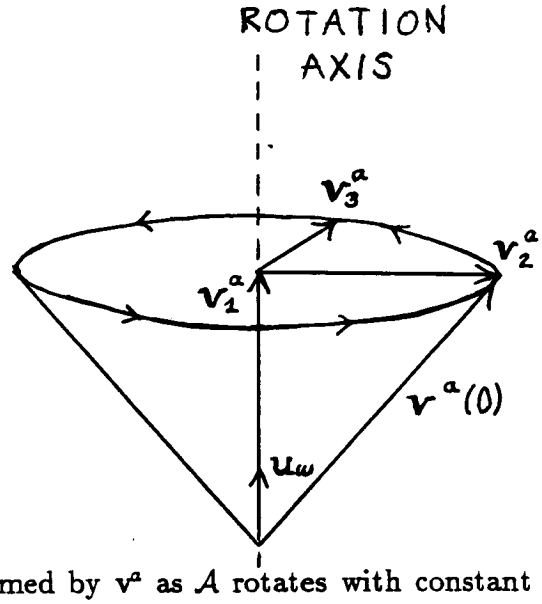
Therefore $\|\mathbf{v}_2^a\| = \|\mathbf{v}_3^a\|$.

Because $\mathbf{v}_2^a$ and $\mathbf{v}_3^a$ have equal magnitudes and are multiplied by sine and cosine terms in the equation for $\mathbf{v}^a(t)$, the time-varying component of $\mathbf{v}^a(t)$ sweeps out a circle in the plane containing these two vectors. The radius of the circle is the common magnitude of the two vectors. The time-varying component of $\mathbf{v}^a(t)$ starts at $\mathbf{v}_2^a$ when $\omega t = 0°$ and travels counterclockwise around the circle when viewed from a point in the half-space of the plane containing $\mathbf{v}_2^a$ and $\mathbf{v}_3^a$ that also contains $\mathbf{v}_1^a$. This component reaches $\mathbf{v}_3^a$ when $\omega t = 90°$ and returns to $\mathbf{v}_2^a$ at the end of the period when $\omega t = 360°$. The circle formed by the time-varying component of $\mathbf{v}^a(t)$ is shown in Figure 3.2.

When the circular behavior of the time-varying component of $\mathbf{v}^a(t)$ is viewed in combination with the constant component contributed by $\mathbf{v}_1^a$, it can be seen that the vector $\mathbf{v}^a(t)$ traces out the surface of a right circular cone as $\mathcal{A}$ rotates about $\mathbf{u}_\omega$. This cone is illustrated in Figure 3.2. The axis of this cone is the constant term $\mathbf{v}_1^a$, which lies along the rotation axis. The "base" of this cone is the circle formed by the time-varying component due



The circle traced out by the time-varying component of $\mathbf{v}^a(t)$, $\mathbf{v}_2^a \cos(\omega t) + \mathbf{v}_3^a \sin(\omega t)$. This circle is the "base" of the cone shown below and has radius $\|\mathbf{v}_2^a\| = \|\mathbf{v}_3^a\|$. $\mathbf{v}_1^a$ points out of the page.



The cone formed by $\mathbf{v}^a$ as $\mathcal{A}$ rotates with constant angular velocity.
$\mathbf{v}^a(t) = \mathbf{v}_1^a + \mathbf{v}_2^a \cos(\omega t) + \mathbf{v}_3^a \sin(\omega t)$

Figure 3.2: A pictorial representation of the behavior of $\mathbf{v}^a$ as a function of time

to $\mathbf{v}_2^a$ and $\mathbf{v}_3^a$. The height of this cone is $\|\mathbf{v}_1^a\|$ and the radius of the base is $\|\mathbf{v}_2^a\| = \|\mathbf{v}_3^a\|$.

The cones swept out by the edge vectors and face normals of $\mathcal{A}$ can be used to give a graphical interpretation of the time behavior of the two types of ACFs presented in section 3.1.4. The following three sections discuss this interpretation for sinusoidal type (i) and (ii) ACFs and for the special case of a constant ACF. For convenience, it is assumed that the plus sign is chosen for the face normals appearing in the definition of the ACFs.

3.2.1 The Type (i) ACFs

Recall that a type (i) ACF is the inner product of an edge vector of $\mathcal{B}$ and a face normal vector for $\mathcal{A}$

$$g^i(t) = \left\langle \mathbf{e}^b, \mathbf{u}^a(t) \right\rangle$$

The value of $g^i(t)$ depends on the location of the edge vector $\mathbf{e}^b$ relative to the rotating plane P^a passing through the origin and having normal vector $\mathbf{u}^a(t)$. P^a remains parallel to the plane that contains the face of $\mathcal{A}$ whose outward-pointing normal vector is $\mathbf{u}^a(t)$. As $\mathcal{A}$ rotates about $\mathbf{u}_\omega$ with constant speed ω, $\mathbf{u}^a(t)$ sweeps out a cone with axis along either $+\mathbf{u}_\omega$ or $-\mathbf{u}_\omega$ depending on the sign of the constant component of $\mathbf{u}^a(t)$, $\mathbf{u}_1^a$. Consider the line in P^a which lies in the same plane as $\mathbf{u}_\omega$ and $\mathbf{u}^a$. As the plane P^a rotates, this line sweeps out two cones, one with axis along $+\mathbf{u}_\omega$ and one with axis along $-\mathbf{u}_\omega$. P^a itself always remains tangent to these two cones as it rotates.

One of the two cones swept out by P^a will have its axis pointing in the same direction as the axis of the cone swept out by $\mathbf{u}^a(t)$. Any vector which lies inside this cone remains in the positive half-space of P^a as it rotates around $\mathbf{u}_\omega$. For this reason, this cone will be referred to as the *positive* cone. The other cone formed by P^a as it rotates will be called the *negative* cone. A vector which is fixed inside the negative cone remains in the negative half-space of P^a as $\mathcal{A}$ rotates around $\mathbf{u}_\omega$. Any vector which lies outside of both the positive and negative cones will be in the positive half-space of P^a for some part of a rotation cycle and in the negative half-space of P^a for the remaining part of the cycle. The positive and negative cones formed by the rotation of P^a are shown in Figure 3.3.

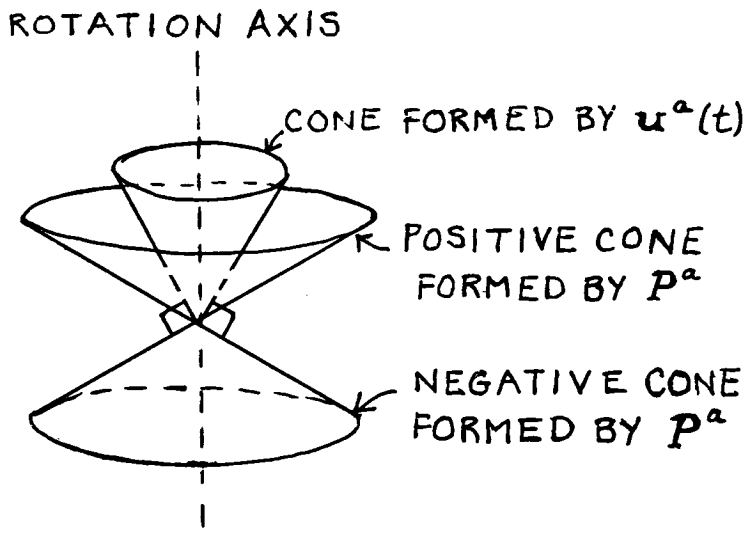
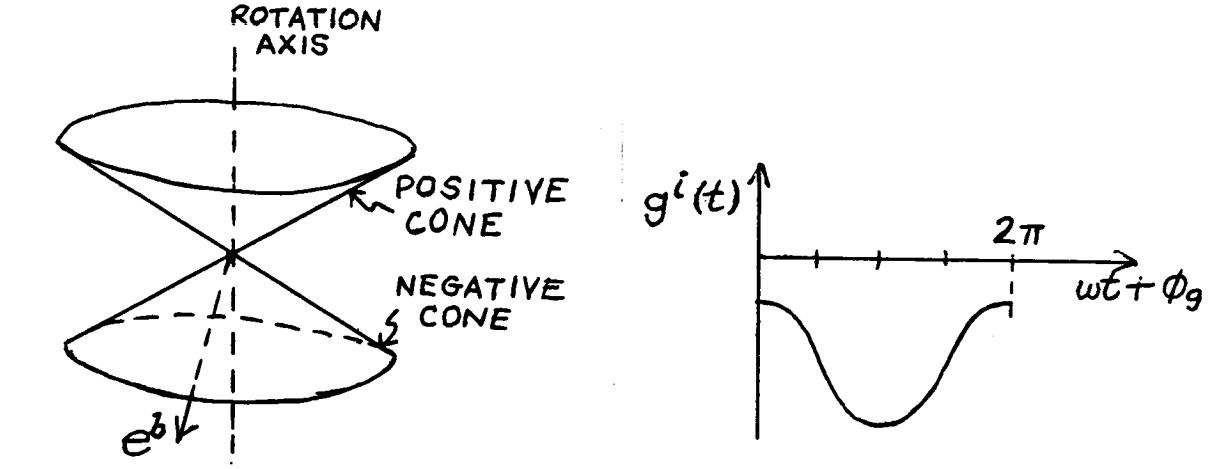


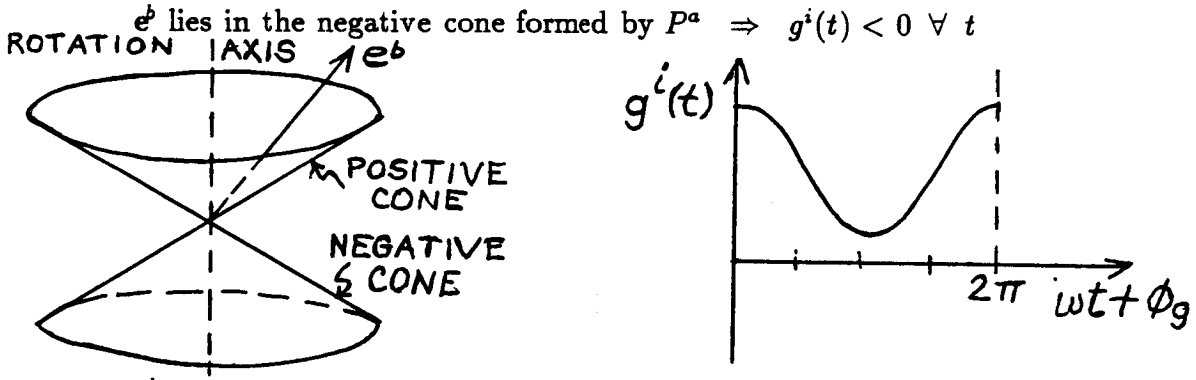
Figure 3.3: The two cones formed by the plane P^a

The three cases for the behavior of a sinusoidal type (i) ACF can be explained in terms of the location of the edge vector e^b relative to the positive and negative cones. The ACF is always less than or equal to zero when e^b lies inside or on the surface of the negative cone. When e^b lies on the surface of the negative cone, there is one point in a period of ωt where the value of the ACF equals zero. For this and only this value of ωt, the edge vector e^b lies in P^a. If e^b lies inside or on the surface of the positive cone, the value of the ACF is always greater than or equal to zero. Again, there is one point in a rotation period when the ACF is equal to zero if e^b lies on the surface of the positive cone. At this point, e^b lies in the plane P^a. Finally, the ACF will take on both positive and negative values in a single period of ωt if e^b lies outside of both the positive and negative cones. The actual values of ωt at the two points in the period where e^b lies in P^a depend on the initial location of P^a as represented by $u^a(0)$. These three cases are illustrated in Figure 3.4. This figure shows the relationship of e^b to the positive and negative cones formed by P^a and a plot of the corresponding sinusoids representing $g^i(t)$ as a function of time.

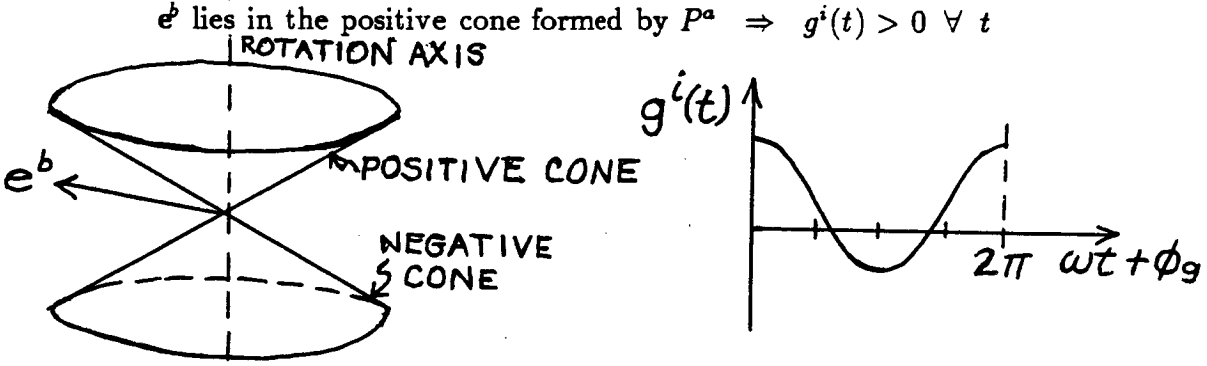
Using the notation of section 3.1.4, the ratio $\|g_1/G\|$ for the type (i) ACF determines whether or not e^b lies inside either of the cones. It is a comparison of the angle made by the lines on the surface of the positive and negative cones with the rotation axis $\mathbf{u}_\omega$ and the angle that e^b makes with the rotation axis. If $\|g_1/G\| > 1$, the half-angle of the cones is larger than the angle of e^b. Therefore, e^b lies inside one of the cones and the value of $g^i(t)$ has constant sign for all values of ωt. The sign of g_1 in this case



e^b lies in the negative cone formed by P^a $\Rightarrow$ $g^i(t) < 0 \ \forall \ t$



e^b lies in the positive cone formed by P^a $\Rightarrow$ $g^i(t) > 0 \ \forall \ t$



e^b lies outside of both the positive and negative cones formed by P^a $\Rightarrow$ $g^i(t)$ changes sign.

Figure 3.4: A graphical interpretation of sinusoidal type (i) ACFs

determines which of the two cones contains $\mathbf{e}^b$. If $\|g_1/G\| < 1$, the angle of $\mathbf{e}^b$ is greater than the half-angle of the cones and so $\mathbf{e}^b$ must lie outside both the cones. The value of $g^i(t)$ then changes sign as ωt proceeds through a cycle. The initial location of $\mathbf{u}^a$ on its cone, and hence the initial location of P^a, is reflected by the value of the phase angle ϕ_g. This determines the two values of ωt where $g^i(t)$ equals zero.

3.2.2 The Type (ii) ACFs

A type (ii) ACF is the inner product of an edge vector of $\mathcal{A}$ and a face normal vector for $\mathcal{B}$

$$g^{ii}(t) = \left\langle \mathbf{e}^a(t), \mathbf{u}^b \right\rangle$$

The location of the rotating edge vector $\mathbf{e}^a(t)$ relative to the fixed plane P^b that passes through the origin and has normal vector $\mathbf{u}^b$ determines the value of $g^{ii}(t)$. P^b is parallel to the plane that contains the face of $\mathcal{B}$ in question.

As $\mathcal{A}$ rotates about $\mathbf{u}_\omega$ with constant speed ω, $\mathbf{e}^a(t)$ sweeps out a cone with axis either $+\mathbf{u}_\omega$ or $-\mathbf{u}_\omega$ depending on the sign of the constant component $\mathbf{e}^a_1$. There are three possibilities for the location of this cone relative to the plane P^b. First, the cone can lie entirely in the closed negative half-space of P^b. If this is the case, then the edge vector $\mathbf{e}^a(t)$ is either in P^b itself or in its open negative half-space for all values of t so that the value of $g^{ii}(t)$ is always less than or equal to zero. If the cone swept out by $\mathbf{e}^a(t)$ lies entirely in the open negative half-space of P^b, then the value of the ACF is strictly negative. If the cone is tangent to P^b, the value of the ACF is zero at the one point in a period of rotation when $\mathbf{e}^a(t)$ lies along the line where the cone and P^b are tangent. The value of the ACF is negative at all other points in the period.

Second, the cone formed by the edge vector of $\mathcal{A}$ can lie entirely in the closed positive half-space of P^b. The value of $g^{ii}(t)$ is then always greater than or equal to zero because the edge vector $\mathbf{e}^a(t)$ is either in P^b itself or in its open positive half-space for all values of t. The value of $g^{ii}(t)$ is strictly positive when the cone swept out by $\mathbf{e}^a(t)$ lies entirely in the open positive half-space of P^b. If the cone is tangent to P^b, the value of $g^{ii}(t)$ is zero at the one point in a period of rotation when $\mathbf{e}^a(t)$ lies along the line where the cone and P^b are tangent. The value of $g^{ii}(t)$ is positive at all other points in the period.

The third possibility is that P^b cuts through the cone swept out by $\mathbf{e}^a(t)$. P^b then divides the cone into two pieces, one that lies in its positive half-space and the other that lies in its negative half-space. The value of the ACF $g^{ii}(t)$ is positive when $\mathbf{e}^a(t)$ lies on the part of the cone that is contained in the positive half-space of P^b. The value of $g^{ii}(t)$ is negative when $\mathbf{e}^a(t)$ lies on the part of its cone that is contained in the negative half-space of P^b. For the two points on the cone where $\mathbf{e}^a(t)$ lies in P^b itself, the value of $g^{ii}(t)$ is zero. The exact values of ωt at these two points depend on the initial location of the edge vector of $\mathcal{A}$ on its cone, as determined by $\mathbf{e}^a(0)$.

Figure 3.5 shows the three cases discussed above for a sinusoidal type (ii) ACF. The figure depicts the relationship of the cone formed by $\mathbf{e}^a$ to the plane P^b and a plot of the corresponding sinusoid expressing $g^{ii}(t)$ as a function of time.

In the case of a type (ii) ACF, the ratio $\|g_1/G\|$ defined in section 3.1.4 determines whether or not the cone formed by $\mathbf{e}^a$ lies entirely in one of the half-spaces of P^b. This ratio is a comparison of the half-angle of the cone swept out by $\mathbf{e}^a$ and the angle between plane P^b and the rotation axis.

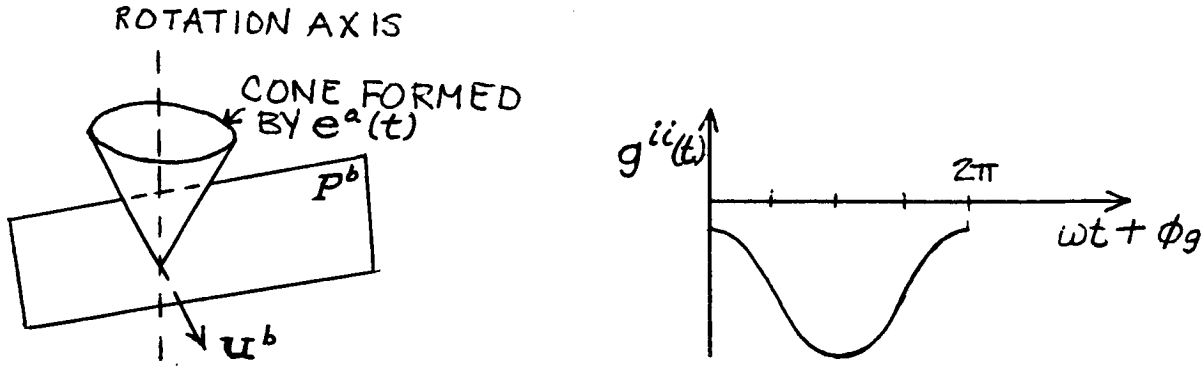
If $\|g_1/G\| > 1$, the angle of P^b is greater than the half-angle of the cone. This means that P^b does not intersect the cone and hence the cone must lie entirely in one of the half-spaces of P^b. The value of $g^{ii}(t)$ therefore has the same sign for all values of ωt. The sign of g_1 determines whether the cone is contained in the positive or negative half-space of P^b.

If $\|g_1/G\| < 1$, the half-angle of the cone is greater than the angle of P^b so that P^b intersects the cone. Consequently, the value of $g^{ii}(t)$ changes sign in a cycle of ωt. The initial location of $\mathbf{e}^a$ on the cone is reflected by the value of the phase angle ϕ_g. The phase angle determines the two values of ωt where $g^{ii}(t) = 0$.

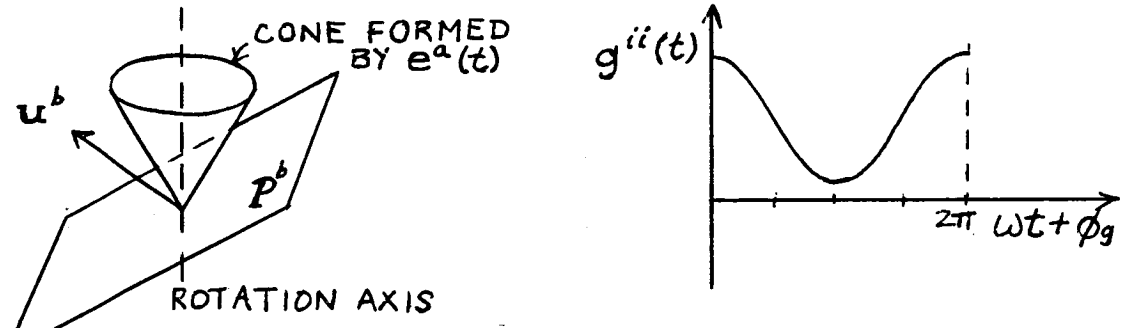
3.2.3 Constant Type (i) and (ii) ACFs

The preceding sections have presented a geometric interpretation for sinusoidal type (i) and (ii) ACFs. The other possibility for the form of an ACF is a constant. This section will briefly discuss the geometric interpretation of constant type (i) and (ii) ACFs.

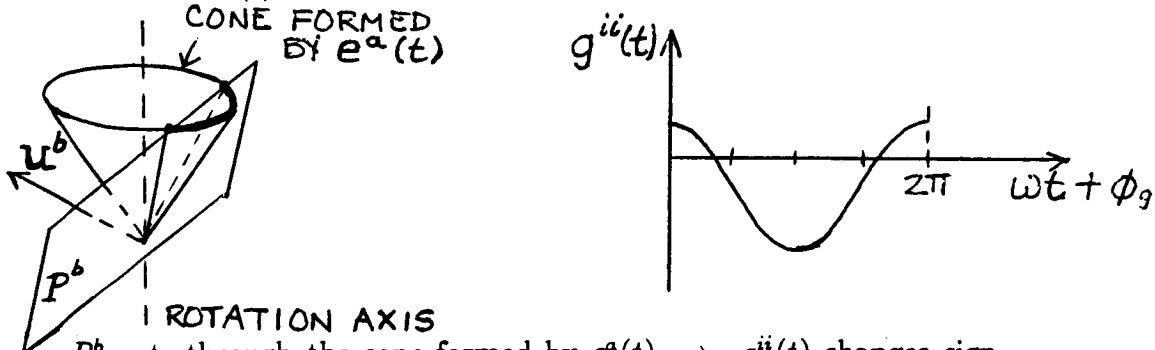
Recall from section 3.1.4 that there are three conditions under which an ACF can be constant. The first two scenarios are that $\mathcal{A}$ is not rotating



The cone formed by $e^a(t)$ lies entirely in the open negative half-space of $P^b \Rightarrow g^{ii}(t) < 0 \ \forall \ t$



The cone formed by $e^a(t)$ lies entirely in the open positive half-space of $P^b \Rightarrow g^{ii}(t) > 0 \ \forall \ t$



P^b cuts through the cone formed by $e^a(t) \Rightarrow g^{ii}(t)$ changes sign

Figure 3.5: A graphical interpretation of sinusoidal type (ii) ACFs

relative to $\mathcal{B}$ or that the vector representing the edge or face normal of $\mathcal{A}$ lies along the rotation axis. In either of these cases, the vector representing the feature of $\mathcal{A}$ is fixed. If $\mathbf{u}^a(t) = \mathbf{u}^a(0)$, the plane P^a is fixed. The resulting constant value of g^i determines which half-space of P^a contains $\mathbf{e}^b$. If $g^i > 0$, then $\mathbf{e}^b$ lies in the open positive half-space of P^a. If $g^i < 0$, then $\mathbf{e}^b$ lies in the open negative half-space of P^a. When $\mathbf{e}^b$ lies in P^a itself, $g^i = 0$. If $\mathbf{e}^a(t) = \mathbf{e}^a(0)$, then $\mathbf{e}^a$ is fixed and the type (ii) ACF g^{ii} is constant. The edge vector $\mathbf{e}^a$ lies in the open positive half-space of P^b when $g^{ii} > 0$ and in the open negative half-space of P^b when $g^{ii} < 0$. When $g^{ii} = 0$, $\mathbf{e}^a$ lies in P^b.

The third way in which an ACF can be constant is if the edge or face normal vector of $\mathcal{B}$ lies along the rotation axis. If $\mathbf{e}^b = \pm k\,\mathbf{u}_\omega$ where k is a positive constant, then $\mathbf{e}^b$ is perpendicular to $\mathbf{u}_2^a$ and $\mathbf{u}_3^a$. It is these two components of $\mathbf{u}^a(t)$ that cause the type (i) ACF g^i to vary as a function of time. Since these two components no longer contribute to the value of the ACF, g^i is constant. Similarly, if $\mathbf{u}^b = \pm\,\mathbf{u}_\omega$, then $\mathbf{u}^b$ is perpendicular to $\mathbf{e}_2^a$ and $\mathbf{e}_3^a$. Only the constant component of $\mathbf{e}^a(t)$, $\mathbf{e}_1^a$, can contribute to the value of the type (ii) ACF g^{ii}. Therefore, g^{ii} is constant.

The constant value of g^i when $\mathbf{e}^b = \pm k\,\mathbf{u}_\omega$ indicates that the relationship of $\mathbf{e}^b$ to plane P^a remains the same as $\mathcal{A}$ rotates. Assume that $\mathbf{u}^a$ has a nonzero component along $\mathbf{u}_\omega$, $\mathbf{u}_1^a \neq \mathbf{0}$. Because the rotation axis is the axis of the cone formed by $\mathbf{u}^a(t)$ and is also the axis of the positive and negative cones formed by P^a, the angles between $\mathbf{e}^b$ and $\mathbf{u}^a$ and between $\mathbf{e}^b$ and P^a remain the same as $\mathcal{A}$ rotates. The edge vector $\mathbf{e}^b$ then always lies in either the positive or negative cone formed by P^a. If $\mathbf{e}^b$ lies on the axis of the positive cone, then $\mathbf{e}^b$ points in the same direction as $\mathbf{u}_1^a$ and $g^i > 0$. If $\mathbf{e}^b$ lies on the axis of the negative cone, then $\mathbf{e}^b$ points opposite to $\mathbf{u}_1^a$ and $g^i < 0$. Now suppose that $\mathbf{u}_1^a = \mathbf{0}$. Then the face normal vector $\mathbf{u}^a(t)$ is perpendicular to $\mathbf{u}_\omega$ so the plane P^a contains the rotation axis. In this case, P^a also contains $\mathbf{e}^b$ and therefore $g^i = 0$.

There is also a fixed relationship between P^b and $\mathbf{e}^a(t)$ for constant type (ii) ACFs when $\mathbf{u}^b = \pm\,\mathbf{u}_\omega$. In this case, the plane P^b is perpendicular to the rotation axis and therefore is perpendicular to the axis of the cone formed by $\mathbf{e}^a(t)$. As $\mathcal{A}$ rotates, $\mathbf{e}^a$ maintains a constant angle with $\mathbf{u}^b$ and with P^b. If $\mathbf{e}_1^a \neq \mathbf{0}$, then the cone formed by $\mathbf{e}^a(t)$ lies entirely in one of the open half-spaces of P^b depending upon the direction of $\mathbf{e}_1^a$ relative to $\mathbf{u}^b$. When $\mathbf{e}_1^a$ points in the same direction as $\mathbf{u}^b$, the cone lies in the positive

half-space and $g^{ii} > 0$. When $\mathbf{e}_1^a$ points opposite to $\mathbf{u}^b$, then the cone lies in the negative half-space and $g^{ii} < 0$. If $\mathbf{e}_1^a = \mathbf{0}$, then $\mathbf{e}^a(t)$ rotates around a circle in a plane which is perpendicular to the rotation axis. In this case, this plane coincides with P^b. The value of g^{ii} is therefore zero because $\mathbf{e}^a$ always lies in plane P^b.

3.3 The Applicability and Orientation Timelines of a C-function

The applicability of a C-function can be tested by examining the signs of each of its ACFs. Since the ACFs are periodic functions of ωt for constant angular velocity, if the trajectory of $\mathcal{A}$ ever passes through the applicability region of a C-function, it will do so repeatedly for each cycle of ωt. Thus, it is only necessary to examine the various ACFs over a single cycle of ωt.

The ACF Timelines The behavior of a single ACF can be represented by a timeline indicating the values of ωt in the interval $(0, 2\pi)$ where the ACF changes sign. Because of the sinusoidal nature of an ACF, its sign changes will alternate. That is, a change in sign from positive to negative will be followed by a change in sign from negative to positive and vice versa. As a notational convenience, a "$\pm$" or "$\mp$" in front of a value of ωt where an ACF changes sign will be used to indicate the type of the sign change. If the value of the ACF changes from positive to negative at ωt, then the entry in its timeline will be $\pm\omega t$. Conversely, if the value of the ACF changes from negative to positive at ωt, then the entry in its timeline will be $\mp\omega t$.

The various possibilities for the time behavior of an ACF were presented in section 3.1.4. Each of these possibilities has a different timeline associated with it. There are no points where the value of a constant ACF changes sign. By convention, let the timeline for a constant ACF contain the single value $\omega t = +2\pi$ when its value is positive and the single value $\omega t = -2\pi$ when its value is negative. For the special case where the ACF is always zero, let the timeline contain only the value $wt = 0$.

The timeline representing a sinusoidal ACF depends on the relative magnitudes of g_1 and G. When $\|g_1/G\| > 1$, the vertical shift is so great that the value of the ACF can never pass through zero and the sign of the

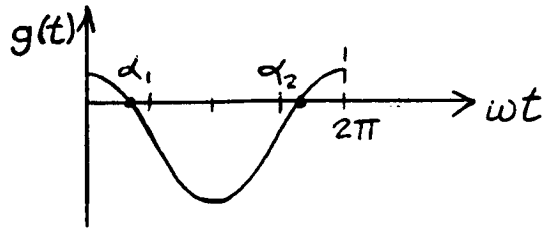
ACF is fixed for all values of ωt. The timeline for an ACF with this behavior will be identical to that of a constant ACF. It contains only one value which is $\omega t = +2\pi$ when $g_1 > G > 0$ and $\omega t = -2\pi$ when $g_1 < -G < 0$. When $\|g_1/G\| < 1$, the vertical shift is small enough so that the cosine curve can cross the zero axis and the ACF changes sign at the two points $\omega t + \phi_g = \alpha_1$ and $\omega t + \phi_g = \alpha_2$ in a cycle of $\omega t + \phi_g$. Each such ACF for one C-function will typically have a different phase angle ϕ_g. To facilitate the comparison of all of these ACFs, the points where each ACF changes sign will be determined as values of ωt, not $\omega t + \phi_g$. The number and location of these transition points depend on the values of ϕ_g, α_1, and α_2.

The four cases for which $g(t)$ changes sign twice in a single cycle of ωt are illustrated in Figure 3.6. There is only one transition point in a cycle of ωt when ϕ_g is equal to α_1 or α_2. These two cases are shown in Figure 3.7. The timelines corresponding to each of these cases for a sinusoidal ACF are listed in these two figures.

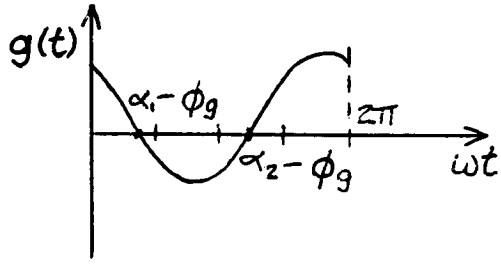
The Applicability Timeline for a C-function The timeline which results from merging the timelines for the individual ACFs according to the applicability criteria for the given C-function is called the *applicability timeline* for this C-function. It contains the intervals in a single cycle of ωt where the trajectory of $\mathcal{A}$ lies in the applicability region for the C-function. These intervals are called the *applicability intervals* for the C-function over the given trajectory of $\mathcal{A}$. Note that since the applicability intervals are defined as intervals of ωt, they are independent of the angular speed ω. In other words, the applicability intervals for the C-functions will be identical for two trajectories of $\mathcal{A}$ having the same rotation axis $\mathbf{u}_\omega$ but different angular speeds ω about this axis.

The applicability timeline for a C-function is similar to the timelines that represent its ACFs. The entries in the applicability timeline are values of $\omega t \in (0, 2\pi)$ that bound the applicability intervals. However, for an applicability timeline the convention shall be that a "$\pm$" in front of a value of ωt indicates that an applicability interval ends at ωt. A "$\mp$" in front of a value of ωt will indicate that an applicability interval begins at ωt.

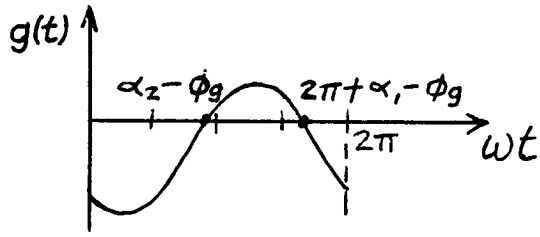
Let the ith entry in the applicability timeline be denoted ωt_i and let n be the total number of entries in the timeline. First, suppose that n is even. If $\{\pm\omega t_1, \mp\omega t_2, \ldots, \mp\omega t_n\}$ is the applicability timeline for the C-function, then its applicability intervals are $[0, \omega t_1], [\omega t_2, \omega t_3], \ldots, [\omega t_n, 2\pi]$.



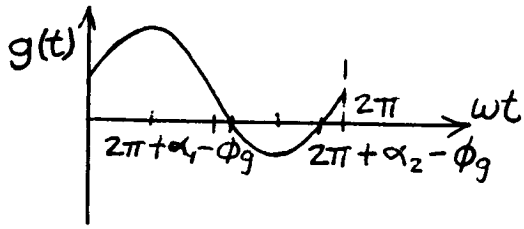
$\phi_g = 0$, $g_3 = 0$, $g_2 > 0$ – The timeline is $\{\pm\alpha_1, \mp\alpha_2\}$



$0 < \phi_g < \alpha_1$ – The timeline is $\{\pm(\alpha_1 - \phi_g), \mp(\alpha_2 - \phi_g)\}$



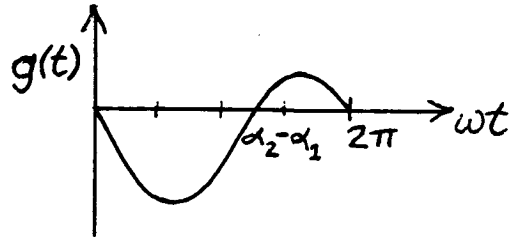
$\alpha_1 < \phi_g < \alpha_2$ – The timeline is $\{\mp(\alpha_2 - \phi_g), \pm(2\pi + \alpha_1 - \phi_g)\}$



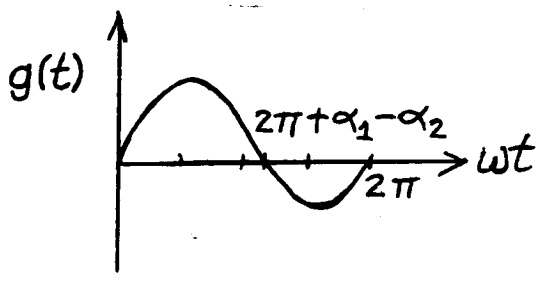
$\alpha_2 < \phi_g < 2\pi$ – The timeline is $\{\pm(2\pi + \alpha_1 - \phi_g), \mp(2\pi + \alpha_2 - \phi_g)\}$

Figure 3.6: The cases where a sinusoidal ACF has two sign changes

C - 2



$\phi_g = \alpha_1$ – The timeline is $\{\mp(\alpha_2 - \alpha_1)\}$



$\phi_g = \alpha_2$ – The timeline is $\{\pm(2\pi + \alpha_1 - \alpha_2)\}$

Figure 3.7: The cases where a sinusoidal ACF has one sign change

Conversely, if the applicability timeline is $\{\mp\omega t_1, \pm\omega t_2, \ldots, \pm\omega t_n\}$ for n an even integer, the applicability intervals of the C-function are $[\omega t_1, \omega t_2], \ldots, [\omega t_{n-1}, \omega t_n]$. Now, suppose n is an odd integer. In this case, if $\{\pm\omega t_1, \mp\omega t_2, \ldots, \pm\omega t_n\}$ is the applicability timeline for the C-function, then its applicability intervals are $[0, \omega t_1], [\omega t_2, \omega t_3], \ldots, [\omega t_{n-1}, \omega t_n]$. On the other hand, if $\{\mp\omega t_1, \pm\omega t_2, \ldots, \mp\omega t_n\}$ is the applicability timeline for the C-function and n is odd, its applicability intervals are $[\omega t_1, \omega t_2], \ldots, [\omega t_n, 2\pi]$.

By convention, an applicability timeline consisting only of the value $\omega t = -2\pi$ will indicate a C-function that is never applicable for the given trajectory of $\mathcal{A}$. In other words, the trajectory of the configuration of $\mathcal{A}$ does not pass through the region of $C_{space}(\mathcal{A})$ where the C-function is applicable. Also, an applicability timeline consisting only of the value $\omega t = +2\pi$ will indicate a C-function that is always applicable along the trajectory of $\mathcal{A}$. In this case, the trajectory of the configuration of $\mathcal{A}$ never leaves the region of $C_{space}(\mathcal{A})$ where the C-function is applicable.

The Orientation Timeline for a C-function The orientation of a C-function at each configuration in $C_{space}(\mathcal{A})$ where it is applicable must be known in order to find its allowable zeros. Fortunately, this information is available as a result of the applicability tests for the C-function. The behavior of the ACFs is such that the orientation of the C-function is the same for all points in one of its applicability intervals. Instead of using additional notation to indicate the orientation in the applicability timeline for the C-function, an auxiliary timeline will be introduced, called the *orientation timeline*. This new timeline gives the orientation of the C-function in each of its applicability intervals.

The orientation timeline for a C-function that is applicable for at least one interval in a cycle of ωt is a sequence of $+1$'s and -1's. There will be one entry in the timeline for each applicability interval of the C-function. An entry of $+1$ indicates that the corresponding applicability interval is an interval where the C-function has positive orientation while an entry of -1 indicates that the applicability interval is an interval where the C-function has negative orientation.

To illustrate the relationship of the applicability and orientation timelines, suppose that the applicability timeline for some C-function is $\{\pm\omega t_1, \mp\omega t_2, \pm\omega t_3\}$. From the convention for the use of "$\pm$" and "$\mp$", the two applicability intervals of the C-function are $[0, \omega t_1]$, and $[\omega t_2, \omega t_3]$. Suppose

also that the C-function has positive orientation in its first applicability interval, $[0, \omega t_1]$, and negative orientation in its second applicability interval, $[\omega t_2, \omega t_3]$. Then, the orientation timeline for this C-function will be $\{+1, -1\}$.

For the special case where the C-function is never applicable, its applicability timeline is $\{-2\pi\}$. The corresponding orientation timeline is taken to be $\{0\}$.

3.3.1 The Applicability Tests

The criteria for merging the ACF timelines to obtain the applicability and orientation timelines depend on the type of the C-function. This merging process is best explained by means of examples for each of the three types of C-functions. The next three sections will give detailed examples for one type (a), one type (b), and one type (c) C-function defined by using two rectangular boxes for $\mathcal{A}$ and $\mathcal{B}$. These two boxes are shown in Figure 3.8 with their vertices, edges, and faces numbered.

Recall that applicability of a C-function depends only on the orientation of $\mathcal{A}$. Also, as mentioned in the preceding section, the applicability intervals are computed as intervals of ωt and are independent of the angular speed ω. Therefore, only the initial orientation of $\mathcal{A}$, represented by $\mathbf{Q}_o$, and the rotation axis $\mathbf{u}_\omega$ need be specified to find the applicability and orientation timelines for the C-functions. The axis of rotation chosen for these examples is

$$\begin{aligned} \mathbf{u}_\omega &= \text{unit}[\ -1 \quad -5 \quad +7\ ]^T \\ &= [\ -0.11547005 \quad -0.57735027 \quad +0.80829038\ ]^T \end{aligned}$$

This axis points out of face 4 of $\mathcal{A}$ as pictured in Figure 3.8. The initial orientation of $\mathcal{A}$ is such that its body coordinate axes are parallel to the corresponding reference coordinate axes. For this orientation, the yaw, pitch, and roll angles are all equal to zero and the quaternion $\mathbf{Q}_o$ is simply the identity quaternion, $\mathbf{Q}_o = [\ 1 \quad \mathbf{0}^T\ ]^T$. The initial alignment of the reference and body coordinate axes means that any vector $\mathbf{v}^a(0)$ representing a feature of $\mathcal{A}$ at its initial orientation is the same as that vector expressed in the body coordinates of $\mathcal{A}$, $\mathbf{v}^a$. In other words, $\mathbf{v}^a(0) = \mathbf{Q}_o \mathbf{v}^a \mathbf{Q}_o^{-1} = \mathbf{v}^a$.

For convenience, the values of ωt in the examples will be given in degrees instead of radians. The entries in the timelines will therefore be values of ωt between 0 and 360°.

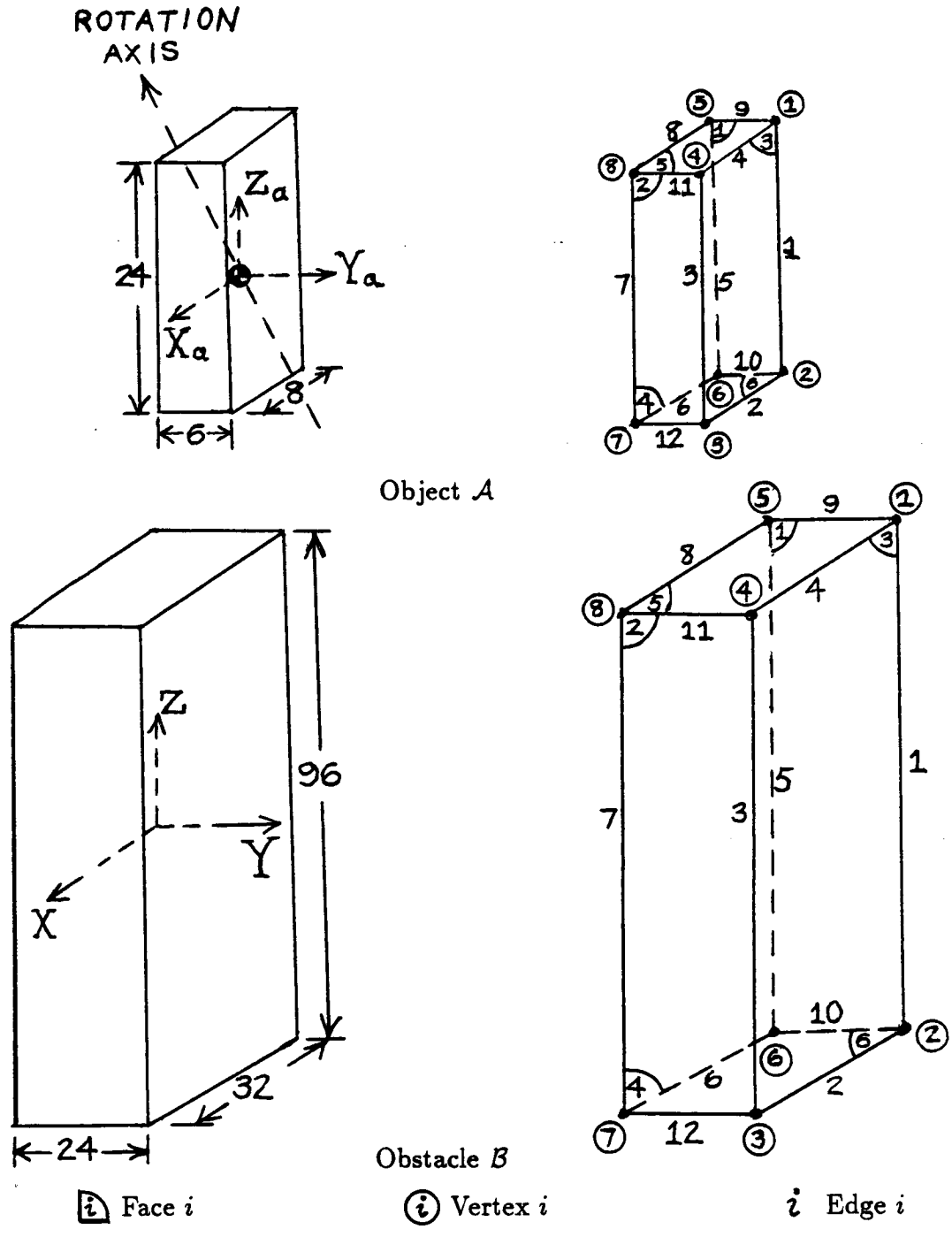


Figure 3.8: The two rectangular boxes representing $\mathcal{A}$ and $\mathcal{B}$.

Figures are included for each of the three examples based on the graphical interpretation of the ACFs given in section 3.2. These figures and the paragraphs discussing them can be ignored if section 3.2 has not been read.

3.3.1.1 A Type (a) Example

The type (a) C-function chosen to illustrate the process of finding the applicability timeline is the function $f^a_{6,1}$ defined by face 6 of $\mathcal{A}$ and vertex 1 of $\mathcal{B}$. There are three edges of $\mathcal{B}$ which meet at vertex 1, namely edges 1, 4, and 9. The three type (i) ACFs for $f^a_{6,1}$ corresponding to these edges of $\mathcal{B}$ are

$$\begin{aligned} g^i_1(t) &= \left\langle \mathbf{e}^b_1, \mathbf{u}^a_6(t) \right\rangle \\ g^i_2(t) &= \left\langle \mathbf{e}^b_4, \mathbf{u}^a_6(t) \right\rangle \\ g^i_3(t) &= \left\langle \mathbf{e}^b_9, \mathbf{u}^a_6(t) \right\rangle \end{aligned}$$

where

$$\begin{aligned} \mathbf{e}^b_1 &= [\ 0 \quad 0 \quad -96\ ]^T \\ \mathbf{e}^b_4 &= [\ +32 \quad 0 \quad 0\ ]^T \\ \mathbf{e}^b_9 &= [\ 0 \quad -24 \quad 0\ ]^T \end{aligned}$$

are the vectors along edges 1, 4, and 9 with tail vertex at vertex 1 of $\mathcal{B}$ and

$$\mathbf{u}^a_6 = \mathbf{u}^a_6(0) = [\ 0 \quad 0 \quad -1\ ]^T$$

is the unit, outward-pointing normal vector to face 6 of $\mathcal{A}$. In order to express these three ACFs as functions of time, the expression for $\mathbf{u}^a_6$ as a function of time must be found. From section 3.1,

$$\mathbf{u}^a_6(t) = \mathbf{u}^a_{6,1} + \mathbf{u}^a_{6,2}\cos(\omega t) + \mathbf{u}^a_{6,3}\sin(\omega t)$$

where

$$\begin{aligned} \mathbf{u}^a_{6,1} &= \langle \mathbf{u}^a_6(0), \mathbf{u}_\omega \rangle\, \mathbf{u}_\omega &&= [\ +0.09\overline{3} \quad +0.4\overline{6} \quad -0.65\overline{3}\ ]^T \\ \mathbf{u}^a_{6,2} &= \mathbf{u}^a_6(0) - \mathbf{u}^a_{6,1} &&= [\ -0.09\overline{3} \quad -0.4\overline{6} \quad -0.34\overline{6}\ ]^T \\ \mathbf{u}^a_{6,3} &= \mathbf{u}_\omega \times \mathbf{u}^a_6(0) &&= [\ +0.57735027 \quad -0.11547005 \quad 0\ ]^T \end{aligned}$$

The three constant coefficients of the ACFs as functions of time are found by substituting the expression for $\mathbf{u}^a_6(t)$ into the definitions of the three ACFs. These coefficients are listed in Table 3.2. Since one of g_2 or g_3 is nonzero for each ACF, all three of them are sinusoidal. They

$g(t) = g_1 + g_2\cos(\omega t) + g_3\sin(\omega t)$				
ACF	k	$g_1 = \left\langle \mathbf{e}_k^b, \mathbf{u}_{6,1}^a \right\rangle$	$g_2 = \left\langle \mathbf{e}_k^b, \mathbf{u}_{6,2}^a \right\rangle$	$g_3 = \left\langle \mathbf{e}_k^b, \mathbf{u}_{6,3}^a \right\rangle$
$g_1^i(t)$	1	+62.719986	+33.280014	0
$g_2^i(t)$	4	+2.9866660	-2.9866660	+18.475209
$g_3^i(t)$	9	-11.199998	+11.199998	+2.7712812

Table 3.2: The constant coefficients of the three ACFs for $f_{6,1}^a$

$g(t) = g_1 + G\cos(\omega t + \phi_g)$					
ACF	G	$\phi_g(^\circ)$	$\|g_1/G\|$	$\alpha_1(^\circ)$	$\alpha_2(^\circ)$
$g_1^i(t)$	+33.280014	0	+1.8846142	–	–
$g_2^i(t)$	+18.715061	260.81712	+0.15958623	99.182880	260.81712
$g_3^i(t)$	+11.537762	346.10211	+0.97072535	13.897886	346.10211

Table 3.3: G, ϕ_g, $\|g_1/G\|$, α_1, and α_2 for the three ACFs for $f_{6,1}^a$

can each be written as the sum of a constant term and a single cosine term. The amplitude, G, and the phase angle, ϕ_g, of the cosine term for each ACF are shown in Table 3.3. This table also shows the ratio $\|g_1/G\|$ and the values of $\omega t + \phi_g$, α_1 and α_2, where the ACF equals zero. For the first ACF, $\|g_1/G\| > 1$ and $g_1 > 0$. This means that $g_1^i(t)$ is always positive and its timeline is $\{+360^\circ\}$. For the second and third ACFs, $\|g_1/G\| < 1$ and $\phi_g = \alpha_2$. Both $g_2^i(t)$ and $g_3^i(t)$ have one sign change from positive to negative at $\omega t = 360^\circ + \alpha_1 - \alpha_2$. The timeline for $g_2^i(t)$ is $\{\pm(360^\circ + 99.182880^\circ - 260.81712^\circ)\} = \{\pm 198.36576^\circ\}$. The timeline for $g_3^i(t)$ is $\{\pm(360^\circ + 13.897886^\circ - 346.10211^\circ)\} = \{\pm 27.797776^\circ\}$.

The applicability criterion for a type (a) C-function states that the function is applicable when all of its ACFs are nonnegative. The applicability intervals of a type (a) C-function are those intervals in a period of ωt where all of its type (i) ACFs are greater than or equal to zero. From the timelines for the three ACFs of the example, the first ACF $g_1^i(t)$ is nonnegative

for the entire period. The second ACF $g_2^i(t)$ is nonnegative in the interval $[0°, 198.36576°]$ and the third ACF $g_3^i(t)$ is nonnegative in the interval $[0°, 27.795776°]$. Clearly, all three ACFs are simultaneously nonnegative only in the interval $[0°, 27.795776°]$. Thus $f_{6,1}^a$ has a single applicability interval for the given rotational motion of $\mathcal{A}$. Its applicability timeline is $\{\pm 27.795776°\}$. By definition, all type (a) C-functions have positive orientation whenever they are applicable. Therefore, the orientation timeline for $f_{6,1}^a$ is $\{+1\}$. Plots of $g_1^i(t)$, $g_2^i(t)$, and $g_3^i(t)$ for one period of ωt are shown in Figure 3.9; the applicability interval of $f_{6,1}^a$ is the region shaded with plus signs to indicate the positive orientation.

Let P_6^a be the rotating plane that passes through the origin and remains parallel to the plane containing face 6 of $\mathcal{A}$. Figure 3.10 shows the locations of the three edge vectors of $\mathcal{B}$, $\mathbf{e}_1^b$, $\mathbf{e}_4^b$, and $\mathbf{e}_9^b$, relative to the positive and negative cones formed by P_6^a as $\mathcal{A}$ rotates. As expected from the behavior of $g_1^i(t)$, $\mathbf{e}_1^b$ lies inside the positive cone. The other two edge vectors, $\mathbf{e}_4^b$ and $\mathbf{e}_9^b$, both lie outside of either cone. Thus, there are some values of ωt where each ACF $g_2^i(t)$ and $g_3^i(t)$ will be positive. However, this is not sufficient to guarantee that $f_{6,1}^a$ will be applicable for some interval ωt. The locations of the three edge vectors must be such that all of them can *simultaneously* lie in the positive half-space of P_6^a for some interval of ωt. This will depend on the exact configuration of the three edge vectors, as indicated by the different phase angles for the three ACFs. In this particular case, the configuration of the three vectors $\mathbf{e}_1^b$, $\mathbf{e}_4^b$, and $\mathbf{e}_9^b$ allows them all to lie in the positive half-space of P_6^a for the interval $\omega t \in [0°, 27.795776°]$.

3.3.1.2 A Type (b) Example

As an example of a type (b) C-function, consider the function $f_{2,3}^b$ defined by vertex 2 of $\mathcal{A}$ and face 3 of $\mathcal{B}$. The three edges of $\mathcal{A}$ which meet at vertex 2 are edges 1, 2, and 10. The vectors along edges 1, 2, and 10 with tail vertex at vertex 2 of $\mathcal{A}$ expressed in the body coordinates of $\mathcal{A}$ are

$$\begin{aligned} \mathbf{e}_1^a &= [\ 0 \quad 0 \quad +24\ ]^T \\ \mathbf{e}_2^a &= [\ +8 \quad 0 \quad 0\ ]^T \\ \mathbf{e}_{10}^a &= [\ 0 \quad -6 \quad 0\ ]^T \end{aligned}$$

The unit, outward-pointing normal vector to face 3 of $\mathcal{B}$ is

$$\mathbf{u}_3^b = [\ 0 \quad +1 \quad 0\ ]^T$$

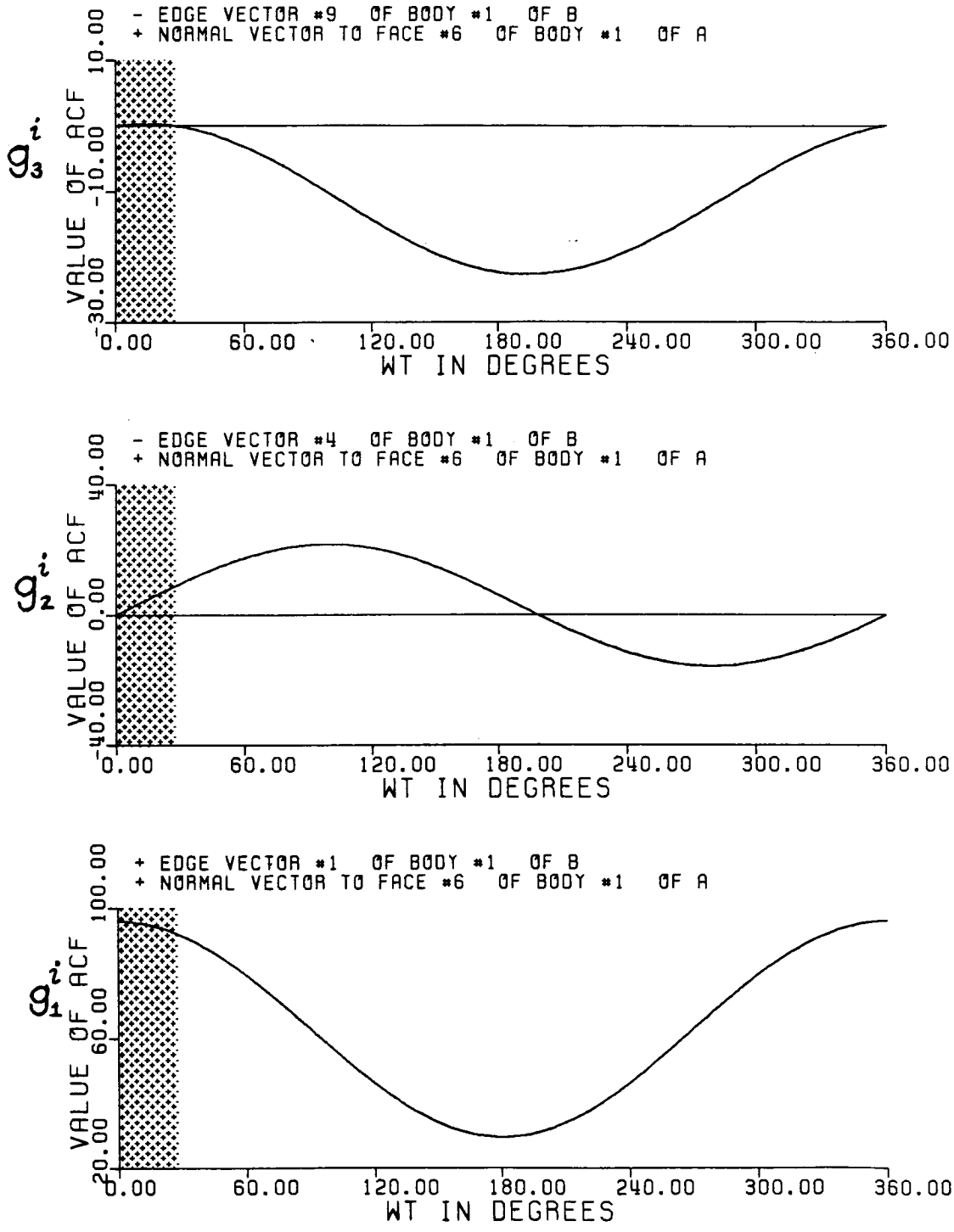
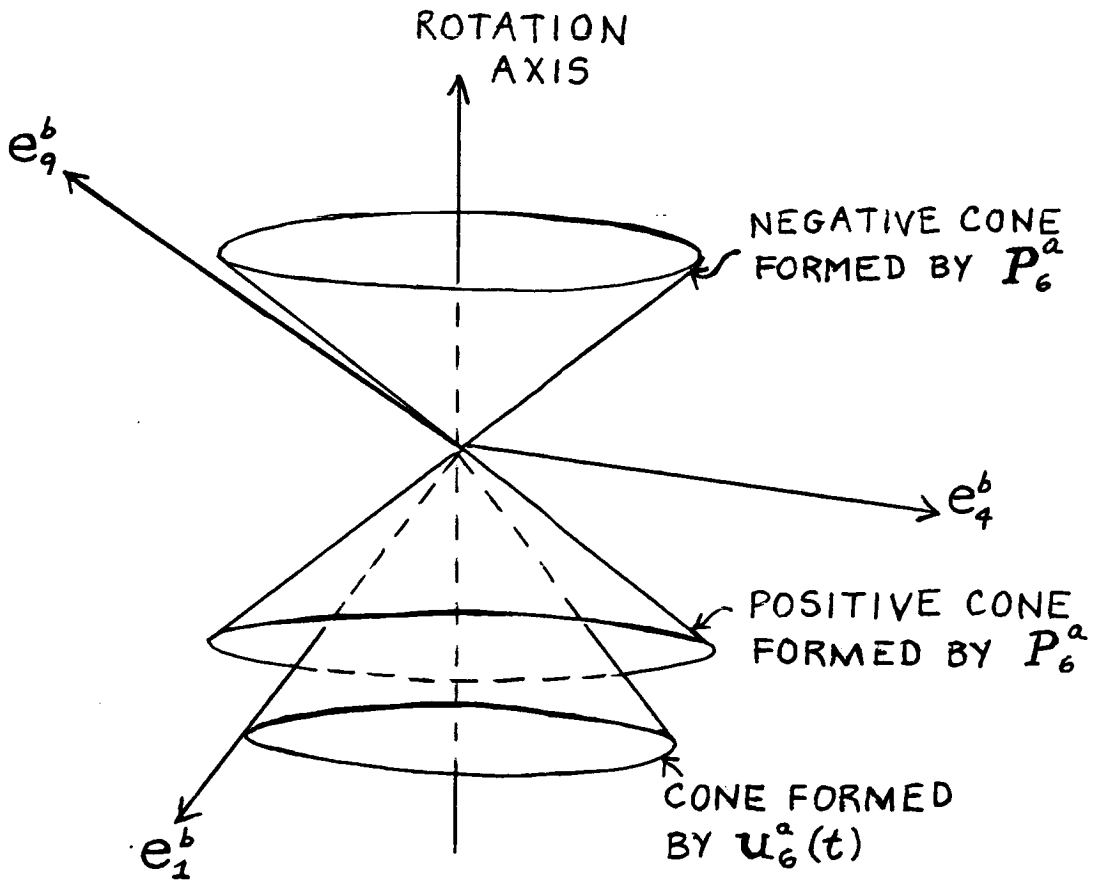


Figure 3.9: The three type (i) ACFs for $f^a_{6,1}$



The ACF corresponding to edge 1 of $\mathcal{B}$ is always positive since e_1^b lies in the positive cone formed by P_6^a. Edge vectors e_4^b and e_9^b lie outside of both the positive and negative cones formed by P_6^a. The two corresponding ACFs, $g_2^i(t)$ and $g_3^i(t)$, both change sign in a cycle of ωt.

Figure 3.10: A graphical interpretation of the type (i) ACFs for $f_{6,1}^a$

$$\mathbf{e}_i^a(t) = \mathbf{e}_{i,1}^a + \mathbf{e}_{i,2}^a \cos(\omega t) + \mathbf{e}_{i,3}^a \sin(\omega t)$$

i	$\mathbf{e}_{i,1}^a = \langle \mathbf{e}_i^a(0), \mathbf{u}_\omega \rangle \, \mathbf{u}_\omega$
1	$[\ -2.24 \quad -11.2 \quad +15.68\ ]^T$
2	$[\ +0.10\overline{6} \quad +0.5\overline{3} \quad -0.74\overline{6}\ ]^T$
10	$[\ -0.4 \quad -2.0 \quad +2.8\ ]^T$

i	$\mathbf{e}_{i,2}^a = \mathbf{e}_i^a(0) - \mathbf{e}_{i,1}^a$
1	$[\ +2.24 \quad +11.2 \quad +8.32\ ]^T$
2	$[\ +7.89\overline{3} \quad -0.5\overline{3} \quad +0.74\overline{6}\ ]^T$
10	$[\ +0.4 \quad -4.0 \quad -2.8\ ]^T$

i	$\mathbf{e}_{i,3}^a = \mathbf{u}_\omega \times \mathbf{e}_i^a(0)$
1	$[\ -13.856406 \quad +2.7712812 \quad 0\ ]^T$
2	$[\ 0 \quad +6.4\overline{6} \quad +4.61\overline{8}\ ]^T$
10	$[\ +4.8497423 \quad 0 \quad +0.69282030\ ]^T$

Table 3.4: The three component vectors of $\mathbf{e}_1^a(t)$, $\mathbf{e}_2^a(t)$, and $\mathbf{e}_{10}^a(t)$

The three type (ii) ACFs for $f_{2,3}^b$ are

$$\begin{aligned} g_1^{ii}(t) &= \left\langle \mathbf{e}_1^a(t), \mathbf{u}_3^b \right\rangle \\ g_2^{ii}(t) &= \left\langle \mathbf{e}_2^a(t), \mathbf{u}_3^b \right\rangle \\ g_3^{ii}(t) &= \left\langle \mathbf{e}_{10}^a(t), \mathbf{u}_3^b \right\rangle \end{aligned}$$

In this case, the three edge vectors of $\mathcal{A}$ must each be found as functions of time. Table 3.4 shows the three component vectors for edges 1, 2, and 10 of $\mathcal{A}$ as functions of time for the given value of $\mathbf{u}_\omega$.

The three constant coefficients for each of the type (ii) ACFs of $f_{2,3}^b$ as functions of time are listed in Table 3.5. These coefficients were computed using the components of the appropriate edge vector of $\mathcal{A}$ from Table 3.4. All three ACFs are sinusoidal because at least one of g_2 and g_3 is nonzero for each ACF. The values of G, ϕ_g, $\|g_1/G\|$, α_1, and α_2 which result from combining the sine and cosine terms are given in Table 3.6.

$g(t) = g_1 + g_2\cos(\omega t) + g_3\sin(\omega t)$				
ACF	k	$g_1 = \left\langle \mathbf{e}^a_{k,1}, \mathbf{u}^b_3 \right\rangle$	$g_2 = \left\langle \mathbf{e}^a_{k,2}, \mathbf{u}^b_3 \right\rangle$	$g_3 = \left\langle \mathbf{u}^a_{k,3}, \mathbf{u}^b_3 \right\rangle$
$g_1^{ii}(t)$	1	-11.2	+11.2	+2.7712812
$g_2^{ii}(t)$	2	$+0.5\overline{3}$	$-0.5\overline{3}$	$+6.4\overline{6}$
$g_3^{ii}(t)$	10	-2.0	-4.0	0

Table 3.5: The constant coefficients of the three ACFs for $f^b_{2,3}$

$g(t) = g_1 + G\cos(\omega t + \phi_g)$					
ACF	G	$\phi_g(^\circ)$	$\|g_1/G\|$	$\alpha_1(^\circ)$	$\alpha_2(^\circ)$
$g_1^{ii}(t)$	+11.537764	346.10211	+0.97072535	13.897886	346.10211
$g_2^{ii}(t)$	+6.4886225	265.28525	+0.082195155	94.714755	265.28525
$g_3^{ii}(t)$	+4.0	180	+0.5	60	300

Table 3.6: G, ϕ_g, $\|g_1/G\|$, α_1, and α_2 for the three ACFs for $f^b_{2,3}$

Because $\|g_1/G\| < 1$ for all three ACFs, $g_1^{ii}(t)$, $g_2^{ii}(t)$, and $g_3^{ii}(t)$ take on both positive and negative values in a single cycle of ωt. For the first two ACFs, $\phi_g = \alpha_2$ and the single sign change is from positive to negative at $\omega t = 360^\circ + \alpha_1 - \alpha_2$. The timeline corresponding to $g_1^{ii}(t)$ is then $\{\pm(360^\circ + 13.897886^\circ - 346.10211^\circ)\} = \{\pm 27.795776^\circ\}$. The timeline corresponding to $g_2^{ii}(t)$ is $\{\pm(360^\circ + 94.714755^\circ - 265.28525^\circ)\} = \{\pm 189.42951^\circ\}$. For the third ACF, $\alpha_1 < \phi_g < \alpha_2$. There are two sign changes for $g_3^{ii}(t)$. The first sign change is from negative to positive at $\omega t = \alpha_2 - \phi_g = 120^\circ$ and the second sign change is from positive to negative at $\omega t = 360^\circ + \alpha_1 - \phi_g = 240^\circ$. The timeline representing $g_3^{ii}(t)$ is then $\{\mp 120^\circ, \pm 240^\circ\}$.

A type (b) C-function is applicable when all of its ACFs are nonnegative. The applicability intervals of a type (b) C-function are those intervals in a period of ωt where all of its type (ii) ACFs are greater than or equal to zero. From the timelines for the three ACFs of $f^b_{2,3}$ in the example, the first ACF $g_1^{ii}(t)$ is nonnegative in the interval $[0^\circ, 27.795776^\circ]$. The second

ACF $g_2^{ii}(t)$ is nonnegative in the interval $[0°, 189.42951°]$. Finally, the third ACF $g_3^{ii}(t)$ is nonnegative in the interval $[120°, 240°]$. These three intervals do not overlap. Therefore, there are no intervals in a cycle of ωt where all three ACFs are nonnegative. This means that $f_{2,3}^b$ is never applicable for the given rotational motion of $\mathcal{A}$. In this case, the applicability timeline for $f_{2,3}^b$ is $\{-360°\}$ and the orientation timeline is $\{0\}$. Plots of $g_1^{ii}(t)$, $g_2^{ii}(t)$, and $g_3^{ii}(t)$ are shown in Figure 3.11. These plots confirm that there are no values of ωt where all three sinusoidal ACFs are greater than or equal to zero.

Let P_3^b be the fixed plane that passes through the origin and is parallel to the plane containing face 3 of $\mathcal{B}$. The relationship of the three cones formed by edge vectors $\mathbf{e}_1^a$, $\mathbf{e}_2^a$, and $\mathbf{e}_{10}^a$ as $\mathcal{A}$ rotates to P_3^b is illustrated in Figure 3.12. In this case, P_3^b cuts through all three of the cones formed by the three edge vectors. This agrees with the fact that all three ACFs change sign in a cycle of ωt. There is some part of each cone in the positive half-space of P_3^b, but this does not guarantee that $f_{2,3}^b$ will be applicable for some interval of a rotation cycle. In fact, $f_{2,3}^b$ is never applicable for the given rotation axis. This is because the configuration of the edges is not such that all three of them can *simultaneously* lie in the positive half-space of P_3^b. Two of the edges will always lie on those parts of their respective cones which are in different half-spaces of P_3^b. The influence of the relative configuration of the three edge vectors on the applicability of $f_{2,3}^b$ is seen in the three phase angles of the ACFs. These phase angles are such that the intervals where the individual ACFs are positive do not overlap.

3.3.1.3 A Type (c) Example

The final example of finding the applicability and orientation timelines will be the type (c) C-function $f_{8,2}^c$ defined by edge 8 of $\mathcal{A}$ and edge 2 of $\mathcal{B}$. Let the tail vertex of edge 8 of $\mathcal{A}$ be vertex 8. Then the vector along edge 8 pointing away from vertex 8 expressed in the body coordinates of $\mathcal{A}$ is

$$\mathbf{e}_8^a = [\ -8 \quad 0 \quad 0\ ]^T$$

The tail vertex of edge 2 of $\mathcal{B}$ is chosen to be vertex 2. The edge vector pointing along edge 2 away from vertex 2 is

$$\mathbf{e}_2^b = [\ +32 \quad 0 \quad 0\ ]^T$$

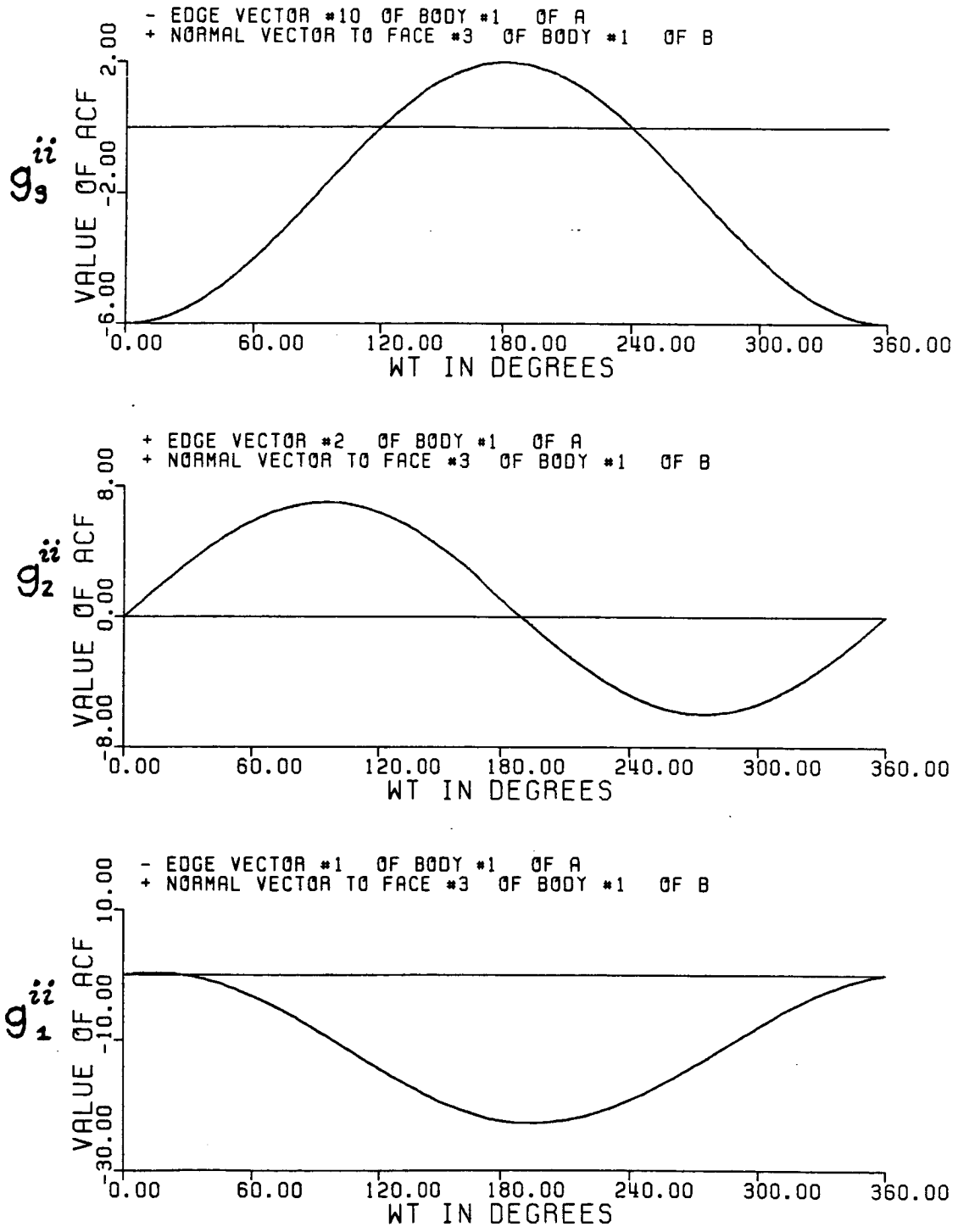
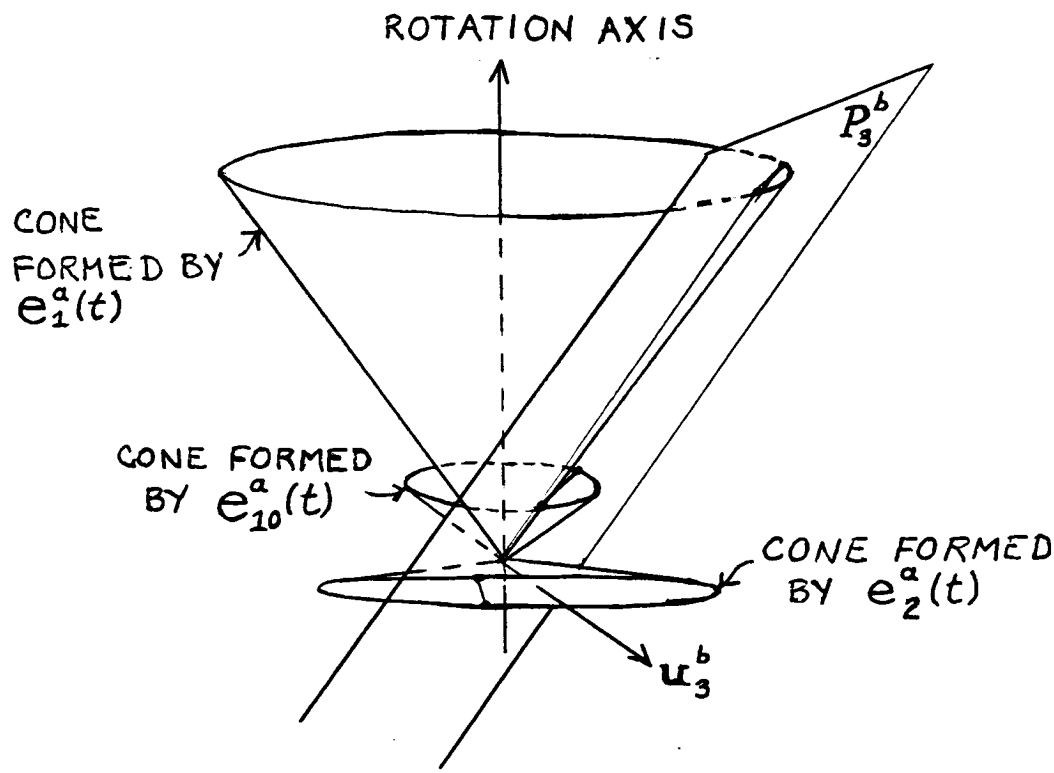


Figure 3.11: The three type (ii) ACFs for $f^b_{2,3}$



All three ACFs change sign in a cycle of ωt because P_3^b intersects the cones formed by e_1^a, e_2^a, and e_{10}^a.

Figure 3.12: A graphical interpretation of the ACFs for $f_{2,3}^b$

There are four ACFs for this type (c) C-function, two type (i) ACFs and two type (ii) ACFs. The two type (i) ACFs involve edge 2 of $\mathcal{B}$ and faces 4 and 5 of $\mathcal{A}$ which meet at edge 8 of $\mathcal{A}$. The unit, outward-pointing normal vectors for these two faces expressed in the body coordinates of $\mathcal{A}$ are

$$\mathbf{u}_4^a = [\ 0 \quad -1 \quad 0\ ]^T \text{ and } \mathbf{u}_5^a = [\ 0 \quad 0 \quad +1\ ]^T$$

The two type (i) ACFs are

$$\begin{aligned} g_1^i(t) &= \left\langle \mathbf{e}_2^b, \pm \mathbf{u}_4^a(t) \right\rangle \\ g_2^i(t) &= \left\langle \mathbf{e}_2^b, \mp \mathbf{u}_5^a(t) \right\rangle \end{aligned}$$

The choice of signs for $\mathbf{u}_4^a$ and $\mathbf{u}_5^a$ in the above equations depends on the cross product of $\mathbf{e}_8^a$ and another edge vector of $\mathcal{A}$ which bounds face 4 or 5 of $\mathcal{A}$ and has the same tail vertex as $\mathbf{e}_8^a$. Edge 7 is the edge of $\mathcal{A}$ which bounds face 4 and has vertex 8 as one of its endpoints. The vector along edge 7 of $\mathcal{A}$ with tail vertex at vertex 8 is

$$\mathbf{e}_7^a = [\ 0 \quad 0 \quad -24\ ]^T$$

The cross product of $\mathbf{e}_7^a$ and $\mathbf{e}_8^a$ is

$$\mathbf{e}_7^a \times \mathbf{e}_8^a = [\ 0 \quad 192 \quad 0\ ]^T = -192\ \mathbf{u}_4^a$$

This means that

$$g_1^i(t) = \left\langle \mathbf{e}_2^b, -\mathbf{u}_4^a(t) \right\rangle$$

and also that

$$g_2^i(t) = \left\langle \mathbf{e}_2^b, +\mathbf{u}_5^a(t) \right\rangle$$

The choice of sign for $\mathbf{u}_5^a$ in g_2^i can be verified as follows. Edge 11 of $\mathcal{A}$ bounds face 5 and has vertex 8 as one of its endpoints. The vector pointing along edge 11 away from vertex 8 is

$$\mathbf{e}_{11}^a = [\ 0 \quad +6 \quad 0\ ]^T$$

The cross product of $\mathbf{e}_{11}^a$ and $\mathbf{e}_8^a$ is

$$\mathbf{e}_{11}^a \times \mathbf{e}_8^a = [\ 0 \quad 0 \quad +48\ ]^T = +48\ \mathbf{u}_5^a$$

This confirms the choice of the plus sign in g_2^i.

The two type (ii) ACFs for $f^c_{8,2}$ involve edge 8 of $\mathcal{A}$ and faces 3 and 6 of $\mathcal{B}$ which meet at edge 2 of $\mathcal{B}$. The unit, outward-pointing normal vectors for these two faces of $\mathcal{B}$ are

$$\mathbf{u}^b_3 = [\ 0 \quad +1 \quad 0\ ]^T \text{ and } \mathbf{u}^b_6 = [\ 0 \quad 0 \quad -1\ ]^T$$

The definitions of the two type (ii) ACFs are

$$\begin{aligned} g^{ii}_1(t) &= \left\langle \mathbf{e}^a_8(t), \pm \mathbf{u}^b_3 \right\rangle \\ g^{ii}_2(t) &= \left\langle \mathbf{e}^a_8(t), \mp \mathbf{u}^b_6 \right\rangle \end{aligned}$$

The signs for $\mathbf{u}^b_3$ and $\mathbf{u}^b_6$ in the above equations are chosen by comparing the cross product of $\mathbf{e}^b_2$ and another edge vector of $\mathcal{B}$ with the appropriate face normal $\mathbf{u}^b_3$ or $\mathbf{u}^b_6$. The other edge vector of $\mathcal{B}$ used in the cross product must bound face 3 or 6 of $\mathcal{B}$ and have the same tail vertex as $\mathbf{e}^b_2$. Edge 1 is the edge of $\mathcal{B}$ which bounds face 3 and has vertex 2 as one of its endpoints. The vector along edge 1 of $\mathcal{B}$ with tail vertex at vertex 2 is

$$\mathbf{e}^b_1 = [\ 0 \quad 0 \quad +96\ ]^T$$

The cross product of $\mathbf{e}^b_2$ and $\mathbf{e}^b_1$ is

$$\mathbf{e}^b_2 \times \mathbf{e}^b_1 = [\ 0 \quad -3072 \quad 0\ ]^T = -3072\ \mathbf{u}^b_3$$

Consequently,

$$g^{ii}_1(t) = \left\langle \mathbf{e}^a_8(t), -\mathbf{u}^b_3 \right\rangle$$

and

$$g^{ii}_2(t) = \left\langle \mathbf{e}^a_8(t), +\mathbf{u}^b_6 \right\rangle$$

The choice of sign for $\mathbf{u}^b_6$ in g^{ii}_2 can be verified in a similar manner. Edge 10 of $\mathcal{B}$ bounds face 6 and has vertex 2 as one of its endpoints. The vector pointing along edge 10 away from vertex 2 is

$$\mathbf{e}^b_{10} = [\ 0 \quad -24 \quad 0\ ]^T$$

The cross product of $\mathbf{e}^b_2$ and $\mathbf{e}^b_{10}$ is

$$\mathbf{e}^b_2 \times \mathbf{e}^b_{10} = [\ 0 \quad 0 \quad -768\ ]^T = +768\ \mathbf{u}^b_6$$

which confirms the choice of the plus sign in g^{ii}_2.

$$\pm \mathbf{u}_i^a(t) = \mathbf{u}_{i,1}^a + \mathbf{u}_{i,2}^a \cos(\omega t) + \mathbf{u}_{i,3}^a \sin(\omega t)$$

$\pm \mathbf{u}_i^a$	$\mathbf{u}_{i,1}^a = \langle \pm \mathbf{u}_i^a(0), \mathbf{u}_\omega \rangle \, \mathbf{u}_\omega$
$-\mathbf{u}_4^a$	$[\ +0.0\overline{6} \quad +0.\overline{3} \quad -0.4\overline{6}\ ]^T$
$+\mathbf{u}_5^a$	$[\ -0.09\overline{3} \quad -0.4\overline{6} \quad +0.65\overline{3}\ ]^T$

$\pm \mathbf{u}_i^a$	$\mathbf{u}_{i,2}^a = \pm \mathbf{u}_i^a(0) - \mathbf{u}_{i,1}^a$
$-\mathbf{u}_4^a$	$[\ -0.0\overline{6} \quad +0.\overline{6} \quad +0.4\overline{6}\ ]^T$
$+\mathbf{u}_5^a$	$[\ +0.09\overline{3} \quad +0.4\overline{6} \quad +0.34\overline{6}\ ]^T$

$\pm \mathbf{u}_i^a$	$\mathbf{u}_{i,3}^a = \mathbf{u}_\omega \times \pm \mathbf{u}_i^a(0)$
$-\mathbf{u}_4^a$	$[\ -0.80829038 \quad 0 \quad -0.11547005\ ]^T$
$+\mathbf{u}_5^a$	$[\ -0.57735027 \quad +0.11547005 \quad 0\ ]^T$

Table 3.7: The components of the face normal vectors $-\mathbf{u}_4^a(t)$ and $\mathbf{u}_5^a(t)$

$g(t) = g_1 + g_2 \cos(\omega t) + g_3 \sin(\omega t)$				
ACF	k	$g_1 = \left\langle \mathbf{e}_2^b, \mathbf{u}_{k,1}^a \right\rangle$	$g_2 = \left\langle \mathbf{e}_2^b, \mathbf{u}_{k,2}^a \right\rangle$	$g_3 = \left\langle \mathbf{e}_2^b, \mathbf{u}_{k,3}^a \right\rangle$
$g_1^i(t)$	4	$+2.1\overline{3}$	$-2.1\overline{3}$	-25.865292
$g_2^i(t)$	5	$-2.98\overline{6}$	$+2.98\overline{6}$	-18.475209

Table 3.8: The constant coefficients of the type (i) ACFs for $f_{8,2}^c$

$g(t) = g_1 + G\cos(\omega t + \phi_g)$					
ACF	G	$\phi_g(°)$	$\|g_1/G\|$	$\alpha_1(°)$	$\alpha_2(°)$
$g_1^i(t)$	+25.953120	94.715004	+0.082199494	94.715004	265.28500
$g_2^i(t)$	+18.715061	80.817118	+0.15958626	80.817118	279.18288

Table 3.9: G, ϕ_g, $\|g_1/G\|$, α_1, and α_2 for the type (i) ACFs for $f^c_{8,2}$

The time behavior of the four ACFs must now be determined. The type (i) ACFs will be considered first. The three constant vectors which appear in the equations for $-\mathbf{u}^a_4(t)$ and $\mathbf{u}^a_5(t)$ are shown in Table 3.7. Using these component vectors to compute the constant coefficients of the two type (i) ACFs as functions of time gives the values listed in Table 3.8. Both $g^i_1(t)$ and $g^i_2(t)$ are sinusoidal and can be rewritten by combining the sine and cosine terms. The resulting magnitude, G, and phase angle, ϕ_g, as well as the ratio $\|g_1/G\|$ and the angles α_1 and α_2 for g^i_1 and g^i_2 are given in Table 3.9. The single sign change for both of these ACFs is from negative to positive and occurs at $\omega t = \alpha_2 - \alpha_1$ since $\phi_g = \alpha_1$. The timeline for $g^i_1(t)$ is $\{\mp(265.28500° - 94.715004°)\} = \{\mp 170.57000°\}$ while the timeline for $g^i_2(t)$ is $\{\mp(279.18288° - 80.817118°)\} = \{\mp 198.36576°\}$.

One of the criteria for applicability of a type (c) C-function is that the two type (i) ACFs be either both nonnegative or both nonpositive. The timeline for $g^i_1(t)$ implies that it is nonnegative only in the interval $[170.5700°, 360°]$. Similarly from the timeline of $g^i_2(t)$, the interval where it is nonnegative is $[198.36576°, 360°]$. The interval $[198.36576°, 360°]$ where these two intervals overlap is the interval where both of the type (i) ACFs for $f^c_{8,2}$ are greater than or equal to zero. Conversely, $g^i_1(t)$ is nonpositive in the interval $[0°, 170.57000°]$ while $g^i_2(t)$ is nonpositive in the interval $[0°, 198.36576°]$. The interval where both of these ACFs are nonpositive is $[0°, 170.5700°]$. In the interval $(170.57000°, 198.36576°)$, the values of the two ACFs have opposite signs.

Now the behavior of the type (ii) ACFs for $f^c_{8,2}$ will be examined. The equation expressing the edge vector of $\mathcal{A}$, $\mathbf{e}^a_8$, appearing in the definition of these two ACFs as a function of time is

$$\mathbf{e}^a_8(t) = \mathbf{e}^a_{8,1} + \mathbf{e}^a_{8,2}\cos(\omega t) + \mathbf{e}^a_{8,3}\sin(\omega t)$$

$g(t) = g_1 + g_2\cos(\omega t) + g_3\sin(\omega t)$				
ACF	$\pm \mathbf{u}_k^b$	$g_1 = \left\langle \mathbf{e}_{8,1}^a, \pm \mathbf{u}_k^b \right\rangle$	$g_2 = \left\langle \mathbf{e}_{8,2}^a, \pm \mathbf{u}_k^b \right\rangle$	$g_3 = \left\langle \mathbf{e}_{8,3}^a, \pm \mathbf{u}_k^b \right\rangle$
$g_1^{ii}(t)$	$-\mathbf{u}_3^b$	$+0.5\overline{3}$	$-0.5\overline{3}$	$+6.4663230$
$g_2^{ii}(t)$	$+\mathbf{u}_6^b$	$-0.74\overline{6}$	$+0.74\overline{6}$	$+4.6188022$

Table 3.10: The constant coefficients of the type (ii) ACFs for $f_{8,2}^c$

$g(t) = g_1 + G\cos(\omega t + \phi_g)$					
ACF	G	$\phi_g(^\circ)$	$\lVert g_1/G \rVert$	$\alpha_1(^\circ)$	$\alpha_2(^\circ)$
$g_1^{ii}(t)$	$+6.4882800$	265.28500	$+0.082199494$	94.715004	265.28500
$g_2^{ii}(t)$	$+4.6787653$	279.18288	$+0.15958626$	80.817118	279.18288

Table 3.11: G, ϕ_g, $\lVert g_1/G \rVert$, α_1, and α_2 for the type (ii) ACFs for $f_{8,2}^c$

where

$$\begin{aligned}
\mathbf{e}_{8,1}^a &= \langle \mathbf{e}_8^a(0), \mathbf{u}_\omega \rangle &&= [\ -0.10\overline{6} \quad -0.5\overline{3} \quad +0.74\overline{6}\ ]^T \\
\mathbf{e}_{8,2}^a &= \mathbf{e}_8^a(0) - \mathbf{e}_{8,1}^a &&= [\ -7.89\overline{3} \quad +0.5\overline{3} \quad -0.74\overline{6}\ ]^T \\
\mathbf{e}_{8,3}^a &= \mathbf{u}_\omega \times \mathbf{e}_8^a(0) &&= [\ 0 \quad -6.4663230 \quad -4.6188022\ ]^T
\end{aligned}$$

Substituting the equation for $\mathbf{e}_8^a(t)$ into the definitions of the two type (ii) ACFs gives the constant coefficients shown in Table 3.10. Each type (ii) ACF is sinusoidal, so the sine and cosine terms can be combined into a single cosine term. The values of G, ϕ_g, $\lVert g_1/G \rVert$, α_1, and α_2 for $g_1^{ii}(t)$ and $g_2^{ii}(t)$ are listed in Table 3.11. For both of these ACFs, $\phi_g = \alpha_2$ and the single sign change is from positive to negative at $\omega t = 360^\circ + \alpha_1 - \alpha_2$. The timeline representing $g_1^{ii}(t)$ is $\{\pm(360^\circ + 94.715004^\circ - 265.28500^\circ)\} = \{\pm 189.43000^\circ\}$. The timeline representing $g_2^{ii}(t)$ is $\{\pm(360^\circ + 80.817118^\circ - 279.18288^\circ)\} = \{\pm 161.63424^\circ\}$.

Another criterion for applicability of a type (c) C-function is that the two type (ii) ACFs be either both nonnegative or both nonpositive. The timeline for $g_1^{ii}(t)$ shows that it is nonnegative in the interval $[0^\circ, 189.43000^\circ]$. Similarly, the timeline representing $g_2^{ii}(t)$ shows that it is nonnegative for

$[0°, 161.63424°]$. Thus, both type (ii) ACFs for $f^c_{8,2}$ are greater than or equal to zero in the interval $[0°, 161.63424°]$. Conversely, $g^{ii}_1(t)$ is nonpositive in the interval $[189.4300°, 360°]$ while $g^{ii}_2(t)$ is nonpositive in the interval $[161.63424°, 360°]$. The interval where both of these ACFs are nonpositive is $[189.43000°, 360°]$. The values of the two type (ii) ACFs have opposite signs in the interval $(161.63424°, 189.43000°)$.

The final criterion for applicability of a type (c) C-function states that both type (i) ACFs must be nonpositive if both type (ii) ACFs are nonnegative and vice versa. These two cases are distinguished by differing orientations for the type (c) C-function. From the above analysis, the interval where both of the type (i) ACFs for $f^c_{8,2}$ are nonnegative is $[198.36576°, 360°]$. The interval where the two type (ii) ACFs are nonpositive is $[189.43000°, 360°]$. In the region where these two intervals overlap, $[198.36576°, 360°]$, $f^c_{8,2}$ is applicable and has positive orientation. The interval where both type (i) ACFs are nonpositive is $[0°, 170.57000°]$ while the interval where both type (ii) ACFs are nonnegative is $[0°, 161.63424°]$. These intervals overlap in the interval $[0°, 161.63424°]$. In this interval, $f^c_{8,2}$ is applicable but has negative orientation. The resulting applicability timeline for $f^c_{8,2}$ is $\{\pm 161.63424, \mp 198.36576\}$ and the orientation timeline is $\{-1, +1\}$. Figures 3.13 and 3.14 show plots of the two type (i) ACFs and the two type (ii) ACFs for $f^c_{8,2}$, respectively. In these plots, the applicability interval where $f^c_{8,2}$ has positive orientation is shaded with plus signs and the applicability interval where it has negative orientation is shaded with minus signs.

A geometric interpretation of the behavior of the four ACFs for $f^c_{8,2}$ is presented in Figures 3.15 and 3.16. Let P^a_4 and P^a_5 be the rotating planes that pass through the origin and remain parallel to the planes containing faces 4 and 5 of $\mathcal{A}$, respectively. Since $-\mathbf{u}^a_4$ is used in the definition of g^i_1, the value of $g^i_1(t)$ is positive whenever $\mathbf{e}^b_2$ is in the negative half-space of P^a_4. In this case, the positive cone formed by P^a_4 as $\mathcal{A}$ rotates is taken to be the cone which shares its axis with the cone formed by $-\mathbf{u}^a_4$. The location of $\mathbf{e}^b_2$ relative to the positive and negative cones formed by the planes P^a_4 and P^a_5 is shown in Figure 3.15. The fact that both type (i) ACFs change sign in a cycle of ωt is confirmed by the location of $\mathbf{e}^b_2$ outside of either of the cones formed by both P^a_4 and P^a_5.

Let P^b_3 and P^b_6 be the two fixed planes that pass through the origin and are parallel to the planes containing faces 3 and 6 of $\mathcal{B}$, respectively. The locations of these planes relative to the cone formed by the rotating

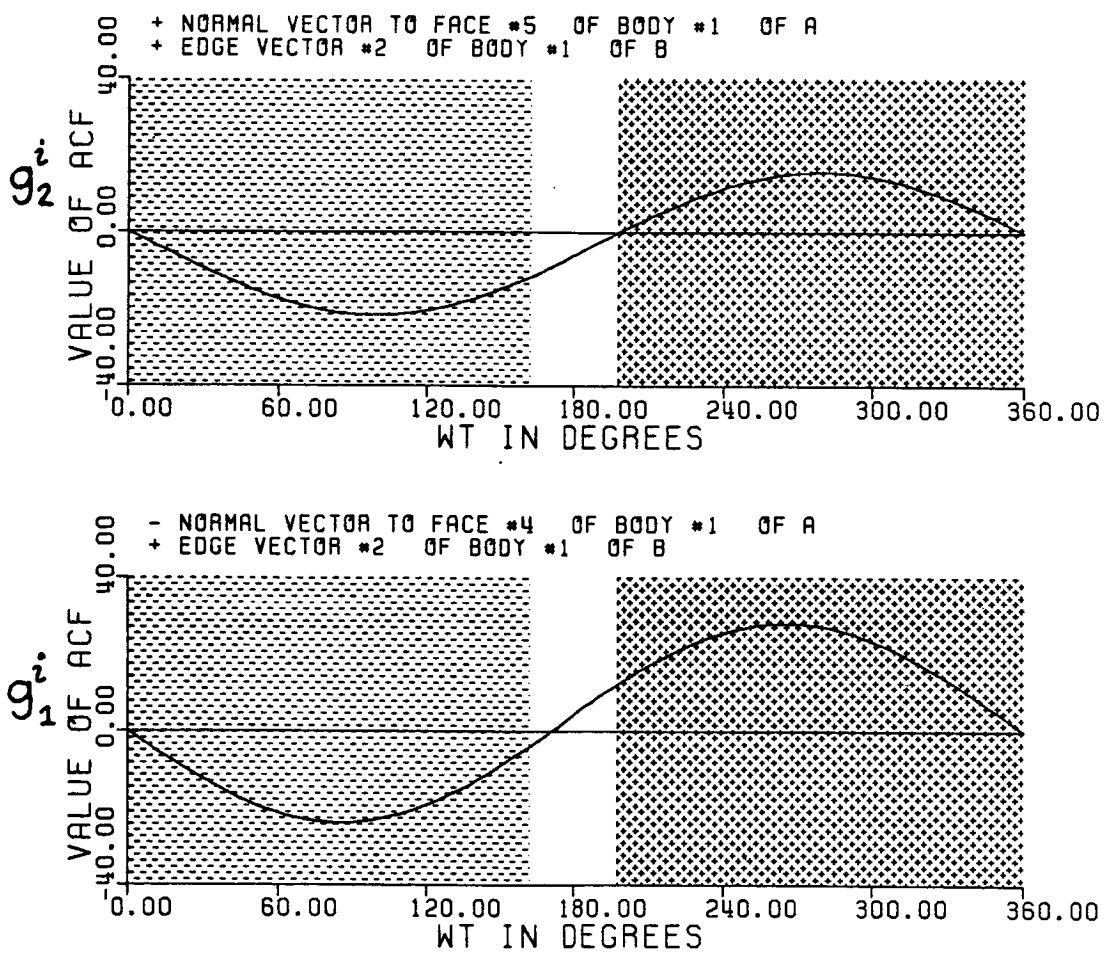
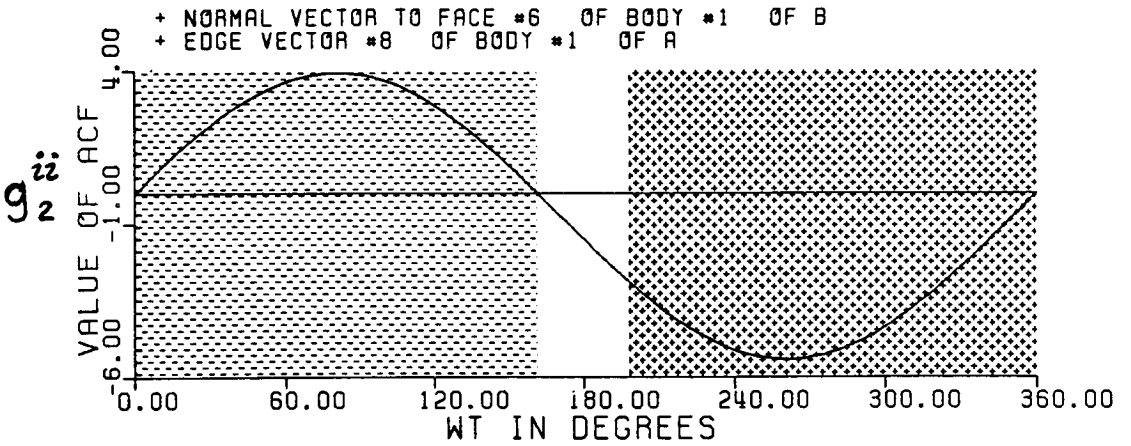


Figure 3.13: The two type (i) ACFs for $f^c_{\&2}$



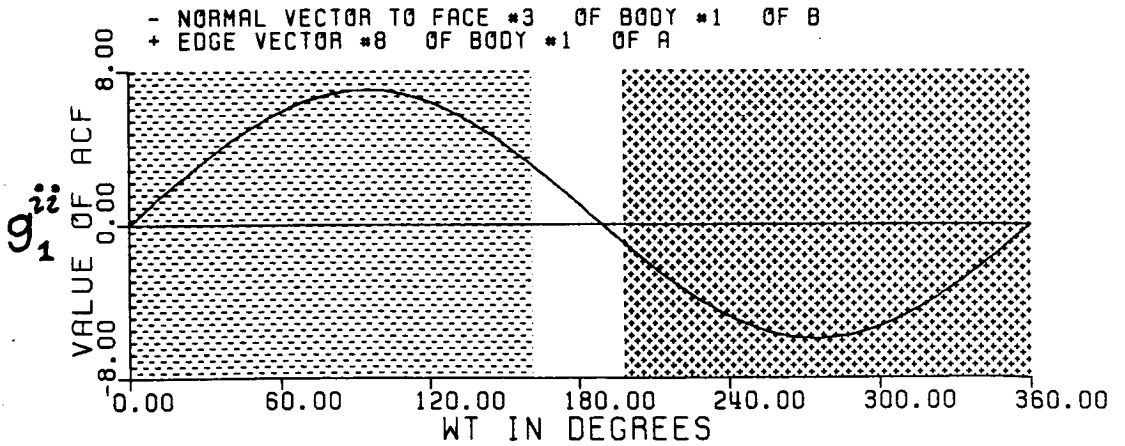
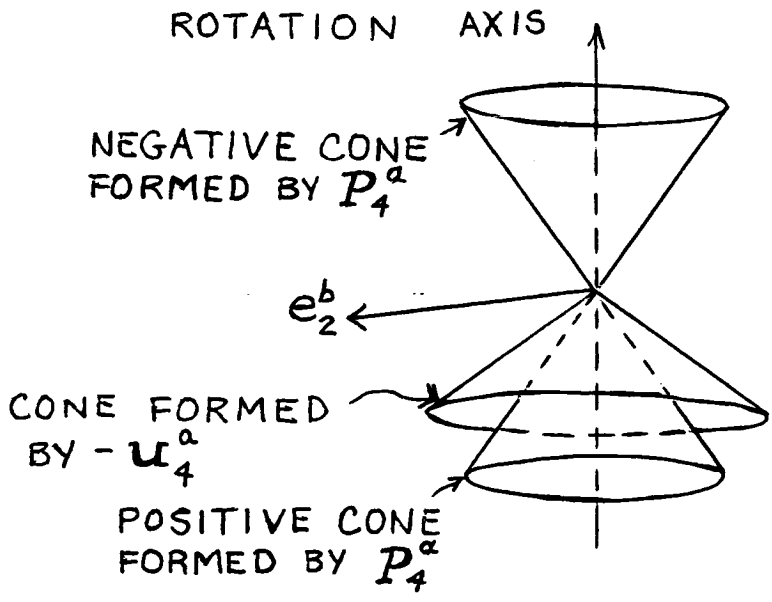
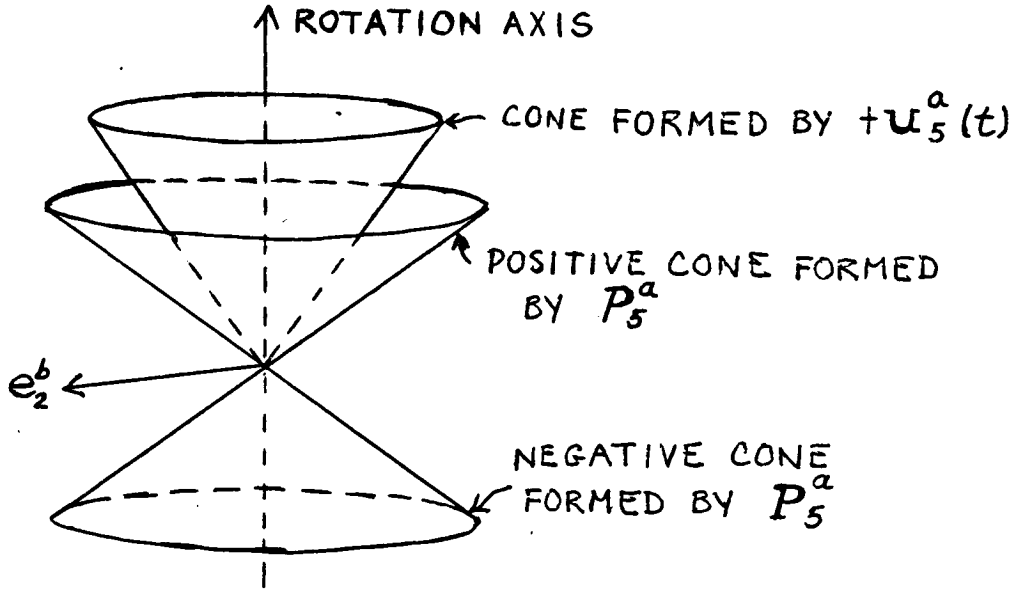


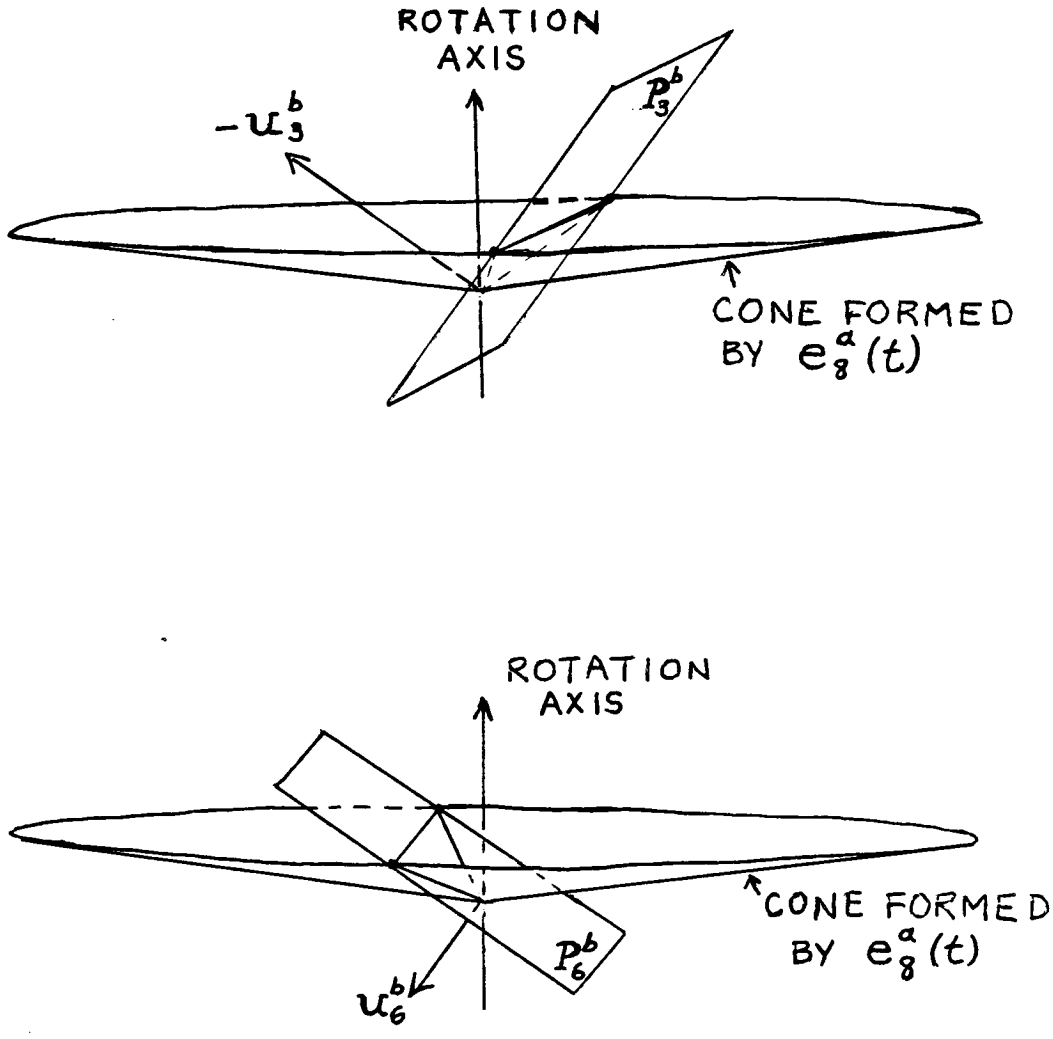
Figure 3.14: The two type (ii) ACFs for $f^c_{8,2}$





The edge vector e_2^b lie outside of both the positive and negative cones formed by P_4^a and P_5^a. The two corresponding ACFs, $g_1^i(t)$ and $g_2^i(t)$, both change sign in a cycle of ωt.

Figure 3.15: A graphical interpretation of the two type (i) ACFs for $f_{8,2}^{c\,i}$



Both $g_1^{ii}(t)$ and $g_2^{ii}(t)$ change sign in a cycle of ωt since P_3^b and P_6^b intersect the cone formed by $\mathbf{e}_8^a$.

Figure 3.16: A graphical interpretation of the two type (ii) ACFs for $f_{8,2}^c$

vector $\mathbf{e}_8^a$ are shown in Figure 3.16. Both P_3^b and P_6^b intersect the cone formed by $\mathbf{e}_8^a$. This agrees with the behavior of the type (ii) ACFs found in the preceding analysis. Both $g_1^{ii}(t)$ and $g_2^{ii}(t)$ take on positive and negative values in a single cycle of ωt.

The three examples presented above have shown how to find the applicability and orientation timelines for each of the three types of C-functions. The ability to find these timelines has two important consequences for the solution of the collision detection problem. First, the search for the allowable zeros of a particular C-function can be confined to its applicability intervals, if any, in each cycle of ωt, as stored in its applicability timeline. Any zeros in these intervals will automatically satisfy the first criterion for allowable zeros, which is also the second collision condition. The orientation of the C-function in its applicability intervals is available from its orientation timeline and is needed to eliminate those of its zeros that do not satisfy the second criterion for allowable zeros. Second, the test of the third collision condition at an allowable zero of a C-function can be simplified. The applicable C-functions at any value of time t are easily found by comparing the value of ωt modulo 2π at that point to the applicability intervals of the various C-functions. The applicability tests do not have to be repeated for all of the C-functions every time an allowable zero is found.

The problem of locating any zeros of a C-function in its applicability intervals still remains to be solved. This problem is addressed in the next two chapters, Chapters 4 and 5, for the case where $\mathcal{A}$ is moving with constant linear and angular velocities relative to $\mathcal{B}$.

Chapter 4

The C-functions for Constant Linear and Angular Velocities

The collision detection algorithm must search for the allowable zeros of the C-functions along the given trajectory of $\mathcal{A}$. In Chapter 3, it was shown that this zero search can be limited to the applicability intervals of each C-function. These applicability intervals were found by examining the form of the ACFs for the specified rotational motion of $\mathcal{A}$. For the case where $\mathcal{A}$ is rotating with constant angular velocity, the values of the ACFs were shown to depend only on the location of $\mathcal{A}$ in a rotation cycle as indicated by the value of ωt. The value of t measures the time elapsed since the initial configuration where $t = 0$. Once the translational motion of $\mathcal{A}$ is also specified, the values of the C-functions will depend on the location of $\mathcal{A}$ along its trajectory as measured by the value of t. In order to develop an appropriate strategy for finding the allowable zeros of the C-functions that occur in their applicability intervals, the behavior of the C-functions as functions of t must be considered.

This chapter will discuss the form of the C-functions for the particular case where $\mathcal{A}$ is moving with constant linear and angular velocity. The first section of this chapter presents a derivation of the three types of C-functions as functions of time and shows that they have a common form. The special cases of translation only and rotation only are also considered.

A practical C-function zero search strategy will require finite bounds on the parts of the trajectory of $\mathcal{A}$ which must be searched for zeros. This requirement is motivated by two different considerations. First, the C-functions are shown to be transcendental functions of t later in this chapter. In the most general case, no analytic formula exists for finding the zeros of such functions. A numerical procedure must be developed to iterate for individual zeros of a C-function. This numerical zero search procedure is the subject of Chapter 5. Second, any numerical search procedure for the zeros of a C-function will only be feasible if the search range can be bounded. However, the applicability intervals of the C-functions as defined in Chapter 3 occur cyclicly with ωt along the trajectory of $\mathcal{A}$. Therefore, a bounded

zero search interval does not result from applicability considerations.

The second through fourth sections of this chapter develop three further methods for bounding the intervals of t that are searched for zeros of the C-functions. The first method, presented in section 4.2, uses the general form of the C-functions to derive mathematical limits on the values of t where a zero can occur. However, this approach does not always yield finite bounds for the search intervals. For this reason, two other methods, referred to as the dynamic spherical approximation and the physical constraints, respectively, are developed in sections 4.3 and 4.4. These methods use only the translational motion of $\mathcal{A}$ and the relative sizes of $\mathcal{A}$ and $\mathcal{B}$ to find bounds for the zero search intervals.

The dynamic spherical approximation and the physical constraints use the third collision condition to limit the zero search interval. This parallels the development in Chapter 3, which described how the second collision condition of applicability can be used to limit the zero search. The new constraints eliminate values of t where the third collision condition cannot be satisfied because of the finite size of the features of $\mathcal{A}$ and $\mathcal{B}$. Any zeros the C-functions may have at values of t that do not satisfy these constraints cannot represent valid collisions of $\mathcal{A}$ and $\mathcal{B}$.

As will be seen in sections 4.3 and 4.4, the dynamic spherical approximation and the physical constraints are approximate and, therefore, do not exactly define the intervals where a zero will satisfy the third collision condition. Consequently, a C-function may have a zero at a value of t that satisfies the approximate constraints but does not satisfy the third collision condition. This contrasts with the applicability conditions, which exactly define the applicability intervals of a C-function. These conditions provide exact bounds on the intervals where the second collision condition is satisfied.

The final section of this chapter gives a short discussion of how the information from the various methods for limiting the zero search intervals is integrated into the collision detection procedure.

4.1 Derivation of the C-functions as Functions of Time

The quaternion representing the orientation of $\mathcal{A}$ as a function of time for constant angular velocity was given in chapter 3. This was used to derive the equation for a vector representing a feature of $\mathcal{A}$ as a function of time. The equation expressing the position of $\mathcal{A}$ as a function of time for constant linear velocity is simply

$$\mathbf{x}(t) = \mathbf{x}_o + \mathbf{v}t$$

where $\mathbf{x}_o$ is the initial position of C^a at time $t = 0$ and $\mathbf{v}$ is the constant linear velocity vector, both expressed in reference coordinates. The above equation for $\mathbf{x}(t)$ along with the equation for the vector representing the appropriate feature of $\mathcal{A}$ as a function of time are substituted into the definitions of the C-functions to obtain their forms as functions of time. Detailed derivations for each of the three types of C-functions are presented in the next three sections.

4.1.1 The Type (a) C-functions

Recall from section 2.2.1 that the definition of the type (a) C-function $f^a_{i,j}$ modelling potential contact of face i of $\mathcal{A}$ and vertex j of $\mathcal{B}$ is

$$f^a_{i,j}\left(\mathbf{x}, \vec{\Theta}\right) = \left\langle \mathbf{u}^a_i\left(\vec{\Theta}\right), \mathbf{b}_j \right\rangle - \left\langle \mathbf{u}^a_i\left(\vec{\Theta}\right), \mathbf{x} + \mathbf{a}_i\left(\vec{\Theta}\right) \right\rangle$$

The second inner product can be divided into two terms so that

$$f^a_{i,j}\left(\mathbf{x}, \vec{\Theta}\right) = \left\langle \mathbf{u}^a_i\left(\vec{\Theta}\right), \mathbf{b}_j \right\rangle - \left\langle \mathbf{u}^a_i\left(\vec{\Theta}\right), \mathbf{x} \right\rangle - \left\langle \mathbf{u}^a_i\left(\vec{\Theta}\right), \mathbf{a}_i\left(\vec{\Theta}\right) \right\rangle$$

The vertex $\mathbf{a}_i\left(\vec{\Theta}\right)$ was chosen to lie on the ith face of $\mathcal{A}$. Therefore, the third inner product measures the distance of the plane containing the ith face of $\mathcal{A}$ from C^a along the outward-pointing normal vector to face i, $\mathbf{u}^a_i\left(\vec{\Theta}\right)$. This distance is a constant which is independent of the coordinate system in which the vectors $\mathbf{u}^a_i$ and $\mathbf{a}_i$ are expressed. Let $d^a_i = \langle \mathbf{u}^a_i, \mathbf{a}_i \rangle$ denote this constant.

The first two inner products in the equation for $f^a_{i,j}$ change as $\mathcal{A}$ translates and rotates. Let $\mathbf{u}^a_i(0) = \mathbf{Q}_o \mathbf{u}^a_i \mathbf{Q}_o{}^{-1}$ be the face normal vector $\mathbf{u}^a_i$

expressed in reference coordinates at time $t = 0$. Then the equation for $\mathbf{u}_i^a$ as a function of time is

$$\mathbf{u}_i^a(t) = \mathbf{u}_1^a + \mathbf{u}_2^a \cos(\omega t) + \mathbf{u}_3^a \sin(\omega t)$$

where

$$\begin{aligned} \mathbf{u}_1^a &= \langle \mathbf{u}_i^a(0), \mathbf{u}_\omega \rangle \, \mathbf{u}_\omega \\ \mathbf{u}_2^a &= \mathbf{u}_i^a(0) - \mathbf{u}_1^a \\ \mathbf{u}_3^a &= \mathbf{u}_\omega \times \mathbf{u}_i^a(0) \end{aligned}$$

Consider the first inner product in the definition of $f_{i,j}^a$, $\langle \mathbf{u}_i^a\left(\vec{\Theta}\right), \mathbf{b}_j \rangle$. Substituting the expression for $\mathbf{u}_i^a$ as a function of time into this term gives

$$\langle \mathbf{u}_i^a(t), \mathbf{b}_j \rangle = \langle \mathbf{u}_1^a, \mathbf{b}_j \rangle + \langle \mathbf{u}_2^a, \mathbf{b}_j \rangle \cos(\omega t) + \langle \mathbf{u}_3^a, \mathbf{b}_j \rangle \sin(\omega t)$$

The second inner product $\langle \mathbf{u}_i^a\left(\vec{\Theta}\right), \mathbf{x} \rangle$ can be further divided into the two terms shown below

$$\langle \mathbf{u}_i^a(t), \mathbf{x}(t) \rangle = \langle \mathbf{u}_i^a(t), \mathbf{x}_o \rangle + \langle \mathbf{u}_i^a(t), \mathbf{v} \rangle \, t$$

Expanding the two inner products in the above equation using the equation for $\mathbf{u}_i^a(t)$ yields

$$\langle \mathbf{u}_i^a(t), \mathbf{x}_o \rangle = \langle \mathbf{u}_1^a, \mathbf{x}_o \rangle + \langle \mathbf{u}_2^a, \mathbf{x}_o \rangle \cos(\omega t) + \langle \mathbf{u}_3^a, \mathbf{x}_o \rangle \sin(\omega t)$$

and

$$\langle \mathbf{u}_i^a(t), \mathbf{v} \rangle = \langle \mathbf{u}_1^a, \mathbf{v} \rangle + \langle \mathbf{u}_2^a, \mathbf{v} \rangle \cos(\omega t) + \langle \mathbf{u}_3^a, \mathbf{v} \rangle \sin(\omega t)$$

When these expressions are substituted into the equation for the $\langle \mathbf{u}_i^a(t), \mathbf{x}(t) \rangle$, the result is

$$\begin{aligned} &\langle \mathbf{u}_i^a(t), \mathbf{x}(t) \rangle = \\ &\quad \{\langle \mathbf{u}_1^a, \mathbf{x}_o \rangle + \langle \mathbf{u}_1^a, \mathbf{v} \rangle \, t\} + \\ &\quad \{\langle \mathbf{u}_2^a, \mathbf{x}_o \rangle + \langle \mathbf{u}_2^a, \mathbf{v} \rangle \, t\} \cos(\omega t) + \{\langle \mathbf{u}_3^a, \mathbf{x}_o \rangle + \langle \mathbf{u}_3^a, \mathbf{v} \rangle \, t\} \sin(\omega t) \end{aligned}$$

The final step in the derivation of $f_{i,j}^a(t)$ is to substitute the expressions for its three terms back into its definition. After combining similar terms, the type (a) C-function as a function of time is

$$\begin{aligned} &f_{i,j}^a(t) = \\ &\quad \{[\langle \mathbf{u}_1^a, \mathbf{b}_j - \mathbf{x}_o \rangle - d_i^a] - \langle \mathbf{u}_1^a, \mathbf{v} \rangle \, t\} + \\ &\quad \{\langle \mathbf{u}_2^a, \mathbf{b}_j - \mathbf{x}_o \rangle - \langle \mathbf{u}_2^a, \mathbf{v} \rangle \, t\} \cos(\omega t) + \{\langle \mathbf{u}_3^a, \mathbf{b}_j - \mathbf{x}_o \rangle - \langle \mathbf{u}_3^a, \mathbf{v} \rangle \, t\} \sin(\omega t) \end{aligned}$$

All of the inner products appearing in this equation are constants. The resulting form of the type (a) C-function is similar to a shifted sinusoid. However, the constant coefficients are replaced by the linear functions of time shown in braces in the above equation for $f^a_{i,j}$.

4.1.2 The Type (b) C-functions

The definition of the type (b) C-function $f^b_{i,j}$ modelling potential contact between vertex i of $\mathcal{A}$ and face j of $\mathcal{B}$ is

$$f^b_{i,j}\left(\mathbf{x}, \vec{\Theta}\right) = \left\langle \mathbf{u}^b_j, \mathbf{x} + \mathbf{a}_i\left(\vec{\Theta}\right)\right\rangle - \left\langle \mathbf{u}^b_j, \mathbf{b}_j\right\rangle$$

When the first inner product is separated into two terms, the type (b) C-function becomes

$$f^b_{i,j}\left(\mathbf{x}, \vec{\Theta}\right) = \left\langle \mathbf{u}^b_j, \mathbf{x}\right\rangle + \left\langle \mathbf{u}^b_j, \mathbf{a}_i\left(\vec{\Theta}\right)\right\rangle - \left\langle \mathbf{u}^b_j, \mathbf{b}_j\right\rangle$$

Because the vertex $\mathbf{b}_j$ was chosen to lie on the jth face of $\mathcal{B}$, the third inner product measures the distance of the plane containing the jth face of $\mathcal{B}$ from the origin of reference coordinates along its outward-pointing normal vector $\mathbf{u}^b_j$. This distance is a constant which depends only on the geometry of $\mathcal{B}$. Let $d^b_j = \langle \mathbf{u}^b_j, \mathbf{b}_j\rangle$ denote this constant.

The time dependence of $f^b_{i,j}$ comes from the vectors $\mathbf{x}$ and $\mathbf{a}_i\left(\vec{\Theta}\right)$ appearing in the first two inner products. Consider the first inner product, $\langle \mathbf{u}^b_j, \mathbf{x}\rangle$. Substituting the expression for $\mathbf{x}(t)$ into this term gives

$$\left\langle \mathbf{u}^b_j, \mathbf{x}(t)\right\rangle = \left\langle \mathbf{u}^b_j, \mathbf{x}_o\right\rangle + \left\langle \mathbf{u}^b_j, \mathbf{v}\right\rangle t$$

Let $\mathbf{a}_i(0) = \mathbf{Q}_o\,\mathbf{a}_i\,\mathbf{Q}_o{}^{-1}$ be the vertex vector $\mathbf{a}_i$ expressed in reference coordinates at the initial time. The equation for $\mathbf{a}_i$ as a function of time for constant angular velocity is

$$\mathbf{a}_i(t) = \mathbf{a}_1 + \mathbf{a}_2\cos(\omega t) + \mathbf{a}_3\sin(\omega t)$$

where

$$\begin{aligned}
\mathbf{a}_1 &= \langle \mathbf{a}_i(0), \mathbf{u}_\omega\rangle\,\mathbf{u}_\omega \\
\mathbf{a}_2 &= \mathbf{a}_i(0) - \mathbf{a}_1 \\
\mathbf{a}_3 &= \mathbf{u}_\omega \times \mathbf{a}_i(0)
\end{aligned}$$

Substituting this into the second inner product of $f^b_{i,j}$, $\langle \mathbf{u}^b_j, \mathbf{a}_i\left(\vec{\Theta}\right)\rangle$, gives

$$\left\langle \mathbf{u}^b_j, \mathbf{a}_i(t)\right\rangle = \left\langle \mathbf{u}^b_j, \mathbf{a}_1\right\rangle + \left\langle \mathbf{u}^b_j, \mathbf{a}_2\right\rangle \cos(\omega t) + \left\langle \mathbf{u}^b_j, \mathbf{a}_3\right\rangle \sin(\omega t)$$

All three inner products in the definition of $f^b_{i,j}$ as functions of time can be substituted back into the equation for $f^b_{i,j}$. The result of this substitution after similar terms have been combined is

$$f^b_{i,j}(t) = \left\{\left[\left\langle \mathbf{u}^b_j, \mathbf{x}_o + \mathbf{a}_1\right\rangle - d^b_j\right] + \left\langle \mathbf{u}^b_j, \mathbf{v}\right\rangle t\right\} + \left\langle \mathbf{u}^b_j, \mathbf{a}_2\right\rangle \cos(\omega t) + \left\langle \mathbf{u}^b_j, \mathbf{a}_3\right\rangle \sin(\omega t)$$

All of the vectors appearing in the inner products of this equation are constant. The inner products are therefore constant scalar coefficients for $f^b_{i,j}$. The above equation for $f^b_{i,j}$ shows the type (b) C-function to be the sum of two pure sinusoidal terms and a term that is linear in time.

4.1.3 The Type (c) C-functions

The final type of C-function to be derived as a function of time is the type (c) C-function. By definition, the type (c) C-function modelling potential contact of edge i of $\mathcal{A}$ and edge j of $\mathcal{B}$ is

$$f^c_{i,j}\left(\mathbf{x}, \vec{\Theta}\right) = \left\langle \mathbf{e}^a_i\left(\vec{\Theta}\right) \times \mathbf{e}^b_j, \mathbf{x} + \mathbf{a}_{i_1}\left(\vec{\Theta}\right)\right\rangle - \left\langle \mathbf{e}^a_i\left(\vec{\Theta}\right) \times \mathbf{e}^b_j, \mathbf{b}_{j_1}\right\rangle$$

This C-function can be written as

$$f^c_{i,j}\left(\mathbf{x}, \vec{\Theta}\right) = \left\langle \mathbf{e}^a_i\left(\vec{\Theta}\right) \times \mathbf{e}^b_j, \mathbf{x}\right\rangle + \left\langle \mathbf{e}^a_i\left(\vec{\Theta}\right) \times \mathbf{e}^b_j, \mathbf{a}_{i_1}\left(\vec{\Theta}\right)\right\rangle - \left\langle \mathbf{e}^a_i\left(\vec{\Theta}\right) \times \mathbf{e}^b_j, \mathbf{b}_{j_1}\right\rangle$$

where the first scalar triple product has been split into two scalar triple products. Each of the three scalar triple products in the definition of $f^c_{i,j}$ will be considered separately. The identities

$$\langle \mathbf{a}, \mathbf{b} \times \mathbf{c}\rangle = \langle \mathbf{b}, \mathbf{c} \times \mathbf{a}\rangle = \langle \mathbf{c}, \mathbf{a} \times \mathbf{b}\rangle$$

will be used to find these scalar triple products as functions of time.

The vectors in the first scalar triple product, $\langle \mathbf{e}^a_i\left(\vec{\Theta}\right) \times \mathbf{e}^b_j, \mathbf{x}\rangle$, can be rearranged as:

$$\left\langle \mathbf{e}^a_i\left(\vec{\Theta}\right) \times \mathbf{e}^b_j, \mathbf{x}\right\rangle = \left\langle \mathbf{e}^a_i\left(\vec{\Theta}\right), \mathbf{e}^b_j \times \mathbf{x}\right\rangle$$

The cross product $\mathbf{e}_j^b \times \mathbf{x}$ as a function of time is

$$\mathbf{e}_j^b \times \mathbf{x}(t) = \left(\mathbf{e}_j^b \times \mathbf{x}_o\right) + \left(\mathbf{e}_j^b \times \mathbf{v}\right) t$$

The two cross products in parentheses in this equation are constant vectors. The first triple product can now be further divided into

$$\left\langle \mathbf{e}_i^a(t) \times \mathbf{e}_j^b, \mathbf{x}(t)\right\rangle = \left\langle \mathbf{e}_i^a(t), \mathbf{e}_j^b \times \mathbf{x}_o\right\rangle + \left\langle \mathbf{e}_i^a(t), \mathbf{e}_j^b \times \mathbf{v}\right\rangle t$$

The expression for $\mathbf{e}_i^a$ as a function of time is

$$\mathbf{e}_i^a(t) = \mathbf{e}_1^a + \mathbf{e}_2^a \cos(\omega t) + \mathbf{e}_3^a \sin(\omega t)$$

where $\mathbf{e}_i^a(0) = \mathbf{Q}_o \mathbf{e}_i^a \mathbf{Q}_o{}^{-1}$ is the edge vector expressed in reference coordinates at the initial time and

$$\begin{aligned} \mathbf{e}_1^a &= \langle \mathbf{e}_i^a(0), \mathbf{u}_\omega \rangle \, \mathbf{u}_\omega \\ \mathbf{e}_2^a &= \mathbf{e}_i^a(0) - \mathbf{e}_1^a \\ \mathbf{e}_3^a &= \mathbf{u}_\omega \times \mathbf{e}_i^a(0) \end{aligned}$$

When this equation for $\mathbf{e}_i^a(t)$ is substituted into the first scalar triple product for $f_{i,j}^c$ the result is

$$\begin{aligned} \left\langle \mathbf{e}_i^a(t) \times \mathbf{e}_j^b, \mathbf{x}(t)\right\rangle &= \left\{\left\langle \mathbf{e}_1^a, \mathbf{e}_j^b \times \mathbf{x}_o\right\rangle + \left\langle \mathbf{e}_1^a, \mathbf{e}_j^b \times \mathbf{v}\right\rangle t\right\} + \\ &\quad \left\{\left\langle \mathbf{e}_2^a, \mathbf{e}_j^b \times \mathbf{x}_o\right\rangle + \left\langle \mathbf{e}_2^a, \mathbf{e}_j^b \times \mathbf{v}\right\rangle t\right\} \cos(\omega t) + \\ &\quad \left\{\left\langle \mathbf{e}_3^a, \mathbf{e}_j^b \times \mathbf{x}_o\right\rangle + \left\langle \mathbf{e}_3^a, \mathbf{e}_j^b \times \mathbf{v}\right\rangle t\right\} \sin(\omega t) \end{aligned}$$

This has the form of a shifted sinusoid with coefficients that are linear functions of time instead of constants. This is the same general form as that of the type (a) C-function $f_{i,j}^a(t)$.

The second scalar triple product, $\langle \mathbf{e}_i^a\left(\vec{\Theta}\right) \times \mathbf{e}_j^b, \mathbf{a}_{i_1}\left(\vec{\Theta}\right)\rangle$, can be rewritten as follows

$$\left\langle \mathbf{e}_i^a\left(\vec{\Theta}\right) \times \mathbf{e}_j^b, \mathbf{a}_{i_1}\left(\vec{\Theta}\right)\right\rangle = \left\langle \mathbf{e}_j^b, \mathbf{a}_{i_1}\left(\vec{\Theta}\right) \times \mathbf{e}_i^a\left(\vec{\Theta}\right)\right\rangle$$

The relative configuration of the two vectors $\mathbf{a}_{i_1}$ and $\mathbf{e}_i^a$ is fixed because $\mathcal{A}$ is a rigid object. Hence, the cross product $\mathbf{a}_{i_1} \times \mathbf{e}_i^a$ is a constant vector when expressed in the body coordinates of $\mathcal{A}$. For convenience in the subsequent

derivation, let $\mathbf{c}_i^a(\vec{\Theta}) = (\mathbf{a}_{i_1} \times \mathbf{e}_i^a)(\vec{\Theta})$ represent this cross product as a function of the orientation of $\mathcal{A}$. Then the second inner product for $f_{i,j}^c$ becomes

$$\left\langle \mathbf{e}_i^a\left(\vec{\Theta}\right) \times \mathbf{e}_j^b, \mathbf{a}_{i_1}\left(\vec{\Theta}\right)\right\rangle = \left\langle \mathbf{e}_j^b, \mathbf{c}_i^a\left(\vec{\Theta}\right)\right\rangle$$

The value of the cross product $\mathbf{c}_i^a = \mathbf{a}_{i_1} \times \mathbf{e}_i^a$ can first be computed in the body coordinates of $\mathcal{A}$. The time variation of this vector can then be treated in the same manner as a vector representing a feature of $\mathcal{A}$. In other words, its expression as a function of time can be found according to the analysis of section 3.1.2. Let $\mathbf{c}_i^a(0) = (\mathbf{a}_{i_1} \times \mathbf{e}_i^a)(0) = \mathbf{Q}_o\,(\mathbf{a}_{i_1} \times \mathbf{e}_i^a)\,\mathbf{Q}_o^{-1}$ be the cross product $\mathbf{c}_i^a$ expressed in reference coordinates at $t = 0$. Then $\mathbf{c}_i^a$ as a function of time is

$$\mathbf{c}_i^a(t) = \mathbf{c}_1^a + \mathbf{c}_2^a \cos(\omega t) + \mathbf{c}_3^a \sin(\omega t)$$

where

$$\begin{aligned}
\mathbf{c}_1^a &= \langle \mathbf{c}_i^a(0), \mathbf{u}_\omega\rangle\, \mathbf{u}_\omega &&= \langle (\mathbf{a}_{i_1} \times \mathbf{e}_i^a)(0), \mathbf{u}_\omega\rangle\, \mathbf{u}_\omega \\
\mathbf{c}_2^a &= \mathbf{c}_i^a(0) - \mathbf{c}_1^a \\
\mathbf{c}_3^a &= \mathbf{u}_\omega \times \mathbf{c}_i^a(0) &&= \mathbf{u}_\omega \times (\mathbf{a}_{i_1} \times \mathbf{e}_i^a)(0)
\end{aligned}$$

Substituting this equation for $\mathbf{c}_i^a(t)$ into the second scalar triple product for $f_{i,j}^c$ gives

$$\begin{aligned}
\left\langle \mathbf{e}_i^a(t) \times \mathbf{e}_j^b, \mathbf{a}_{i_1}(t)\right\rangle &= \left\langle \mathbf{e}_j^b, \mathbf{c}_i^a(t)\right\rangle \\
&= \left\langle \mathbf{e}_j^b, \mathbf{c}_1^a\right\rangle + \left\langle \mathbf{e}_j^b, \mathbf{c}_2^a\right\rangle \cos(\omega t) + \left\langle \mathbf{e}_j^b, \mathbf{c}_3^a\right\rangle \sin(\omega t)
\end{aligned}$$

The three inner products in the above expression are all constants. Thus, the second scalar triple product for $f_{i,j}^c$ is simply a shifted sinusoid with constant coefficients when expressed as a function of time.

The last scalar triple product in the definition of $f_{i,j}^c$, $\langle \mathbf{e}_i^a\left(\vec{\Theta}\right) \times \mathbf{e}_j^b, \mathbf{b}_{j_1}\rangle$, can be rearranged to obtain

$$\left\langle \mathbf{e}_i^a\left(\vec{\Theta}\right) \times \mathbf{e}_j^b, \mathbf{b}_{j_1}\right\rangle = \left\langle \mathbf{e}_i^a\left(\vec{\Theta}\right), \mathbf{e}_j^b \times \mathbf{b}_{j_1}\right\rangle$$

The cross product $\mathbf{e}_j^b \times \mathbf{b}_{j_1}$ will be denoted $\mathbf{c}_j^b$. This vector is similar to the cross product $\mathbf{c}_i^a = \mathbf{a}_{i_1} \times \mathbf{e}_i^a$ introduced in the preceding paragraph. Since $\mathcal{B}$ is stationary, $\mathbf{c}_j^b$ is a constant vector. The equation for $\mathbf{e}_i^a$ as a function of time has already been found in deriving the expression for the first scalar

triple product of $f^c_{i,j}$. The result of substituting $\mathbf{e}^a_i(t)$ into the third scalar triple product is

$$\begin{aligned}\left\langle \mathbf{e}^a_i(t) \times \mathbf{e}^b_j, \mathbf{b}_{j_1} \right\rangle &= \left\langle \mathbf{e}^a_i(t), \mathbf{c}^b_j \right\rangle \\ &= \left\langle \mathbf{e}^a_1, \mathbf{c}^b_j \right\rangle + \left\langle \mathbf{e}^a_2, \mathbf{c}^b_j \right\rangle \cos(\omega t) + \left\langle \mathbf{e}^a_3, \mathbf{c}^b_j \right\rangle \sin(\omega t)\end{aligned}$$

As was the case for the second triple product, the third triple product is a shifted sinusoidal function of time because the inner products in the expression above are all constants.

The final step in deriving the type (c) C-function as a function of time is to combine the three scalar triple products as functions of time. This yields the equation below where similar terms have been combined

$$\begin{aligned}&f^c_{i,j}(t) = \\ &\quad \left\{\left[\left\langle \mathbf{e}^a_1, \mathbf{e}^b_j \times \mathbf{x}_o \right\rangle + \left\langle \mathbf{e}^b_j, \mathbf{c}^a_1 \right\rangle - \left\langle \mathbf{e}^a_1, \mathbf{c}^b_j \right\rangle\right] + \left\langle \mathbf{e}^a_1, \mathbf{e}^b_j \times \mathbf{v} \right\rangle t\right\} + \\ &\quad \left\{\left[\left\langle \mathbf{e}^a_2, \mathbf{e}^b_j \times \mathbf{x}_o \right\rangle + \left\langle \mathbf{e}^b_j, \mathbf{c}^a_2 \right\rangle - \left\langle \mathbf{e}^a_2, \mathbf{c}^b_j \right\rangle\right] + \left\langle \mathbf{e}^a_2, \mathbf{e}^b_j \times \mathbf{v} \right\rangle t\right\} \cos(\omega t) + \\ &\quad \left\{\left[\left\langle \mathbf{e}^a_3, \mathbf{e}^b_j \times \mathbf{x}_o \right\rangle + \left\langle \mathbf{e}^b_j, \mathbf{c}^a_3 \right\rangle - \left\langle \mathbf{e}^a_3, \mathbf{c}^b_j \right\rangle\right] + \left\langle \mathbf{e}^a_3, \mathbf{e}^b_j \times \mathbf{v} \right\rangle t\right\} \sin(\omega t)\end{aligned}$$

The inner products in this equation are all constant. As expected from the forms of its three triple products, the type (c) C-function as a function of time has the same form as the type (a) C-function. This general form is that of a shifted sinusoid having coefficients that are linear terms in time.

4.1.4 The C-functions for Translation Only

For the case where $\mathcal{A}$ is translating but not rotating, $\vec{\omega} = \mathbf{0}$ and the vectors representing the features of $\mathcal{A}$ are constant. Therefore, their values in reference coordinates depend only on the initial orientation of $\mathcal{A}$. Hence, the ith face normal, edge, and vertex vectors are equal to the constant vectors $\mathbf{u}^a_i(0)$, $\mathbf{e}^a_i(0)$, and $\mathbf{a}_i(0)$, respectively. The time dependence of the C-functions in this case is due solely to the presence of $\mathbf{x}$ in their definitions.

When $\mathcal{A}$ is not rotating, the type (a) C-function $f^a_{i,j}$ becomes

$$f^a_{i,j}(t) = \langle \mathbf{u}^a_i(0), \mathbf{b}_j \rangle - \langle \mathbf{u}^a_i(0), \mathbf{x}(t) \rangle - d^a_i$$

Similarly, the type (b) C-function $f^b_{i,j}$ becomes

$$f^b_{i,j}(t) = \left\langle \mathbf{u}^b_j, \mathbf{x}(t) + \mathbf{a}_i(0) \right\rangle - d^b_j$$

and the type (c) C-function $f^c_{i,j}$ becomes

$$f^c_{i,j}(t) = \left\langle \mathbf{e}^a_i(0) \times \mathbf{e}^b_j, \mathbf{x}(t) + \mathbf{a}_i(0) \right\rangle - \left\langle \mathbf{e}^a_i(0) \times \mathbf{e}^b_j, \mathbf{b}_j \right\rangle$$

Substituting for $\mathbf{x}(t)$ into each of these three equation gives the following equations for $f^a_{i,j}(t)$, $f^b_{i,j}(t)$, and $f^c_{i,j}(t)$ when $\vec{\omega} = \mathbf{0}$:

$$\begin{aligned}
f^a_{i,j}(t) &= [\langle \mathbf{u}^a_i(0), \mathbf{b}_j - \mathbf{x}_o \rangle - d^a_i] - \langle \mathbf{u}^a_i(0), \mathbf{v} \rangle t \\
f^b_{i,j}(t) &= \left[\left\langle \mathbf{u}^b_j, \mathbf{x}_o + \mathbf{a}_i(0) \right\rangle - d^b_j\right] + \left\langle \mathbf{u}^b_j, \mathbf{v} \right\rangle t \\
f^c_{i,j}(t) &= \left\langle \mathbf{e}^a_i(0) \times \mathbf{e}^b_j, \mathbf{x}_o + \mathbf{a}_i(0) - \mathbf{b}_j \right\rangle + \left\langle \mathbf{e}^a_i(0) \times \mathbf{e}^b_j, \mathbf{v} \right\rangle t
\end{aligned}$$

Therefore, all three types of C-functions are linear in time and have no sinusoidal terms for the case of translation only.

4.1.5 The C-functions for Rotation Only

Now consider the special case where $\mathbf{v} = \mathbf{0}$ and $\mathcal{A}$ is rotating but not translating. In this case, the position vector $\mathbf{x}$ is equal to the constant initial position vector, $\mathbf{x} = \mathbf{x}_o$. The face normal, edge, and vertex vectors for $\mathcal{A}$ are functions of time because $\mathcal{A}$ is rotating. The equations for $\mathbf{u}^a_i(t)$, $\mathbf{e}^a_i(t)$, and $\mathbf{a}_i(t)$ were given in the preceding sections.

Using the equation for $\mathbf{u}^a_i(t)$, the type (a) C-function $f^a_{i,j}$ becomes

$$\begin{aligned}
f^a_{i,j}(t) &= \langle \mathbf{u}^a_i(t), \mathbf{b}_j - \mathbf{x}_o \rangle - d^a_i \\
&= [\langle \mathbf{u}^a_1, \mathbf{b}_j - \mathbf{x}_o \rangle - d^a_i] + \\
&\quad \langle \mathbf{u}^a_2, \mathbf{b}_j - \mathbf{x}_o \rangle \cos(\omega t) + \langle \mathbf{u}^a_3, \mathbf{b}_j - \mathbf{x}_o \rangle \sin(\omega t)
\end{aligned}$$

when $\mathcal{A}$ is not translating. In this case, the type (b) C-function $f^b_{i,j}$ becomes

$$\begin{aligned}
f^b_{i,j}(t) &= \left\langle \mathbf{u}^b_j, \mathbf{x}_o \right\rangle + \left\langle \mathbf{u}^b_j, \mathbf{a}_i(t) \right\rangle - d^b_j \\
&= \left[\left\langle \mathbf{u}^b_j, \mathbf{x}_o + \mathbf{a}_1 \right\rangle - d^b_j\right] + \left\langle \mathbf{u}^b_j, \mathbf{a}_2 \right\rangle \cos(\omega t) + \left\langle \mathbf{u}^b_j, \mathbf{a}_3 \right\rangle \sin(\omega t)
\end{aligned}$$

using the equation for $\mathbf{a}_i(t)$. Thus, the form of both the type (a) and (b) C-functions for the case of rotation only is a shifted sinusoid with constant coefficients.

The type (c) C-function can also be shown to have a shifted sinusoidal form for the case where $\mathcal{A}$ is not translating. When $\mathbf{v} = \mathbf{0}$, only the terms

involving $\mathbf{x}_o$ remain in the expression for the first scalar triple product as a function of time. The derivation of the other two scalar triple products as functions of time is identical to that given in section 4.1.3. Combining the three scalar triple products as functions of time and omitting the terms containing $\mathbf{v}$ gives

$$\begin{aligned} f^c_{i,j}(t) &= \left[\left\langle \mathbf{e}^a_1, \mathbf{e}^b_j \times \mathbf{x}_o \right\rangle + \left\langle \mathbf{e}^b_j, \mathbf{c}^a_1 \right\rangle - \left\langle \mathbf{e}^a_1, \mathbf{c}^b_j \right\rangle\right] + \\ &\quad \left[\left\langle \mathbf{e}^a_2, \mathbf{e}^b_j \times \mathbf{x}_o \right\rangle + \left\langle \mathbf{e}^b_j, \mathbf{c}^a_2 \right\rangle - \left\langle \mathbf{e}^a_2, \mathbf{c}^b_j \right\rangle\right] \cos(\omega t) + \\ &\quad \left[\left\langle \mathbf{e}^a_3, \mathbf{e}^b_j \times \mathbf{x}_o \right\rangle + \left\langle \mathbf{e}^b_j, \mathbf{c}^a_3 \right\rangle - \left\langle \mathbf{e}^a_3, \mathbf{c}^b_j \right\rangle\right] \sin(\omega t) \end{aligned}$$

Therefore, all three types of C-functions have the form of a shifted sinusoid as functions of time when $\mathcal{A}$ is rotating but not translating.

4.1.6 The Generalized C-function

Examining the equations for $f^a_{i,j}(t)$, $f^b_{i,j}(t)$, and $f^c_{i,j}(t)$, it is clear that they share a common form as functions of time. This form will be referred to as the generalized C-function $f(t)$ where

$$f(t) = (m_1 t + l_1) + (m_2 t + l_2)\cos(\omega t) + (m_3 t + l_3)\sin(\omega t)$$

The six coefficients m_1, m_2, m_3, l_1, l_2, and l_3 are scalar constants which depend on the type of the C-function and the vectors representing the particular features of $\mathcal{A}$ and $\mathcal{B}$ which define it. The values of these coefficients derived in sections 4.1.1, 4.1.2, and 4.1.3 for each of the three types of C-functions for the case of combined translation and rotation of $\mathcal{A}$ are listed in Table 4.1. For the type (a) and (c) C-functions, all six coefficients can be nonzero. This results from the derivations in sections 4.1.1 and 4.1.3 which shows these two types of C-functions to be shifted sinusoids having three coefficients that are linear in time instead of constant. However, both m_2 and m_3 are zero for a type (b) C-function so that terms of the form $t\cos(\omega t)$ and $t\sin(\omega t)$ are absent from type (b) C-functions. This stems from the analysis of section 4.1.2 which shows the type (b) C-functions to be sums of a linear term and two pure sinusoidal terms in time.

For the special case of translation only, the four coefficients m_2, m_3, l_2, and l_3 are all zero and all three types of C-functions have the linear form

$f(t) = (m_1 t + l_1) + (m_2 t + l_2)\cos(\omega t) + (m_3 t + l_3)\sin(\omega t)$			
	Type (a)	Type (b)	Type (c)
m_1	$-\langle \mathbf{u}_1^a, \mathbf{v}\rangle$	$\left\langle \mathbf{u}_j^b, \mathbf{v}\right\rangle$	$\left\langle \mathbf{e}_1^a, \mathbf{e}_j^b \times \mathbf{v}\right\rangle$
m_2	$-\langle \mathbf{u}_2^a, \mathbf{v}\rangle$	0	$\left\langle \mathbf{e}_2^a, \mathbf{e}_j^b \times \mathbf{v}\right\rangle$
m_3	$-\langle \mathbf{u}_3^a, \mathbf{v}\rangle$	0	$\left\langle \mathbf{e}_3^a, \mathbf{e}_j^b \times \mathbf{v}\right\rangle$
l_1	$\langle \mathbf{u}_1^a, \mathbf{b}_j - \mathbf{x}_o\rangle - d_i^a$	$\left\langle \mathbf{u}_j^b, \mathbf{x}_o + \mathbf{a}_1\right\rangle - d_j^b$	$\left\langle \mathbf{e}_1^a, \mathbf{e}_j^b \times \mathbf{x}_o\right\rangle + \left\langle \mathbf{e}_j^b, \mathbf{c}_1^a\right\rangle - \left\langle \mathbf{e}_1^a, \mathbf{c}_j^b\right\rangle$
l_2	$\langle \mathbf{u}_2^a, \mathbf{b}_j - \mathbf{x}_o\rangle$	$\left\langle \mathbf{u}_j^b, \mathbf{a}_2\right\rangle$	$\left\langle \mathbf{e}_2^a, \mathbf{e}_j^b \times \mathbf{x}_o\right\rangle + \left\langle \mathbf{e}_j^b, \mathbf{c}_2^a\right\rangle - \left\langle \mathbf{e}_2^a, \mathbf{c}_j^b\right\rangle$
l_3	$\langle \mathbf{u}_3^a, \mathbf{b}_j - \mathbf{x}_o\rangle$	$\left\langle \mathbf{u}_j^b, \mathbf{a}_3\right\rangle$	$\left\langle \mathbf{e}_3^a, \mathbf{e}_j^b \times \mathbf{x}_o\right\rangle + \left\langle \mathbf{e}_j^b, \mathbf{c}_3^a\right\rangle - \left\langle \mathbf{e}_3^a, \mathbf{c}_j^b\right\rangle$

Table 4.1: The six constant coefficients for the generalized C-function

$m_1 t + l_1$. The values of m_1 and l_1 found in section 4.1.4 for the different types of C-functions when $\vec{\omega} = \mathbf{0}$ are shown in Table 4.2.

For the special case of rotation only, the three coefficients m_1, m_2, and m_3 are all zero. In this case, all three types of C-functions have the shifted sinusoidal form $l_1 + l_2\cos(\omega t) + l_3\sin(\omega t)$. The values of l_1, l_2, and l_3 derived in section 4.1.5 for the different types of C-functions when $\mathbf{v} = \mathbf{0}$ are shown in Table 4.3.

In summary, this section has shown that all three types of C-functions share a common form, $f(t)$, as functions of time. This is true even for the special cases where $\mathcal{A}$ is translating but not rotating and when $\mathcal{A}$ is rotating but not translating. Therefore, the form of the generalized C-function $f(t)$ will be used for the analysis of the C-functions in the remainder of this thesis. This is especially important in Chapter 5 where the zero search procedure for a C-function will be shown to depend on the values of the coefficients of $f(t)$.

$\vec{\omega} = \mathbf{0} \Rightarrow f(t) = m_1 t + l_1$			
	Type (a)	Type (b)	Type (c)
m_1	$-\langle \mathbf{u}_i^a(0), \mathbf{v} \rangle$	$\left\langle \mathbf{u}_j^b, \mathbf{v} \right\rangle$	$\left\langle \mathbf{e}_i^a(0) \times \mathbf{e}_j^b, \mathbf{v} \right\rangle$
l_1	$\langle \mathbf{u}_i^a(0), \mathbf{b}_j - \mathbf{x}_o \rangle - d_i^a$	$\left\langle \mathbf{u}_j^b, \mathbf{x}_o + \mathbf{a}_i(0) \right\rangle - d_j^b$	$\left\langle \mathbf{e}_i^a(0) \times \mathbf{e}_j^b, \mathbf{x}_o + \mathbf{a}_i(0) - \mathbf{b}_j \right\rangle$

Table 4.2: The coefficients m_1 and l_1 of the generalized C-function for the case of translation only

$\mathbf{v} = \mathbf{0} \Rightarrow f(t) = l_1 + l_2 \cos(\omega t) + l_3 \sin(\omega t)$			
	Type (a)	Type (b)	Type (c)
l_1	$\langle \mathbf{u}_1^a, \mathbf{b}_j - \mathbf{x}_o \rangle - d_i^a$	$\left\langle \mathbf{u}_j^b, \mathbf{x}_o + \mathbf{a}_1 \right\rangle - d_j^b$	$\left\langle \mathbf{e}_1^a, \mathbf{e}_j^b \times \mathbf{x}_o \right\rangle + \left\langle \mathbf{e}_j^b, \mathbf{c}_1^a \right\rangle - \left\langle \mathbf{e}_1^a, \mathbf{c}_j^b \right\rangle$
l_2	$\langle \mathbf{u}_2^a, \mathbf{b}_j - \mathbf{x}_o \rangle$	$\left\langle \mathbf{u}_j^b, \mathbf{a}_2 \right\rangle$	$\left\langle \mathbf{e}_2^a, \mathbf{e}_j^b \times \mathbf{x}_o \right\rangle + \left\langle \mathbf{e}_j^b, \mathbf{c}_2^a \right\rangle - \left\langle \mathbf{e}_2^a, \mathbf{c}_j^b \right\rangle$
l_3	$\langle \mathbf{u}_3^a, \mathbf{b}_j - \mathbf{x}_o \rangle$	$\left\langle \mathbf{u}_j^b, \mathbf{a}_3 \right\rangle$	$\left\langle \mathbf{e}_3^a, \mathbf{e}_j^b \times \mathbf{x}_o \right\rangle + \left\langle \mathbf{e}_j^b, \mathbf{c}_3^a \right\rangle - \left\langle \mathbf{e}_3^a, \mathbf{c}_j^b \right\rangle$

Table 4.3: The coefficients l_1, l_2 and l_3 for the generalized C-function for the case of rotation only

4.2 Analytic Root Intervals for the Generalized C-function

Recall that the applicability intervals of a C-function do not provide finite bounds for the search for its zeros. Another way to bound the zero search intervals is to examine the form of the generalized C-function to determine the regions where it is possible for it to have zero value. The intervals of time where it is mathematically possible for $f(t)$ to be zero are called the *analytic root intervals.*[1]

Consider the two sinusoidal terms in the definition of $f(t)$. They can be combined into a single cosine term with a time-varying amplitude $F(t)$ and phase angle $\phi_f(t)$. The resulting expression for $f(t)$ is

$$f(t) = (m_1 t + l_1) + F(t)\cos(\omega t + \phi_f(t))$$

where

$$F(t) = \sqrt{(m_2 t + l_2)^2 + (m_3 t + l_3)^2} \geq 0$$

and

$$\begin{aligned} F(t)\cos\phi_f(t) &= m_2 t + l_2 \\ -F(t)\sin\phi_f(t) &= m_3 t + l_3 \end{aligned}$$

For any value of t where f is zero,

$$F(t)\cos(\omega t + \phi_f(t)) = -(m_1 t + l_1)$$

Taking the absolute value of each side of this equation gives

$$F(t)\,|\cos(\omega t + \phi_f(t))| = |m_1 t + l_1|$$

Since the magnitude of the cosine is always less than or equal to one, it is only possible for $f(t)$ to equal zero in an interval of time where

$$F(t) \geq |m_1 t + l_1|$$

Thus, the phase angle ϕ_f is eliminated by converting the original equality to this inequality. The analytic root intervals of $f(t)$ are defined to be the regions where

$$\sqrt{(m_2 t + l_2)^2 + (m_3 t + l_3)^2} \geq |m_1 t + l_1|$$

[1]The derivation of the analytic root intervals for $f(t)$ presented in this section was suggested to the author by reference [22].

Clearly, the locations of these regions will depend on the values of the six coefficients m_1, m_2, m_3, l_1, l_2, and l_3.

The inequality above can be rearranged to obtain a requirement that the value of a polynomial in t of degree no greater than 2 be greater than or equal to zero. Squaring both sides and subtracting the right-hand term from both sides of this inequality gives

$$(m_2t + l_2)^2 + (m_3t + l_3)^2 - (m_1t + l_1)^2 \geq 0$$

When the squared terms are expanded and terms of the same degree in t are combined, the result is

$$(m_2^2 + m_3^2 - m_1^2)t^2 + 2(m_2l_2 + m_3l_3 - m_1l_1)t + (l_2^2 + l_3^2 - l_1^2) \geq 0$$

For convenience, the polynomial on the left-hand side of this inequality will be denoted $p(t)$ and its three constant coefficients will be denoted p_1, p_2, and p_3 where

$$\begin{aligned} p(t) &= p_1t^2 + 2p_2t + p_3 \\ p_1 &= (m_2^2 + m_3^2 - m_1^2) \\ p_2 &= (m_2l_2 + m_3l_3 - m_1l_1) \\ p_3 &= (l_2^2 + l_3^2 - l_1^2) \end{aligned}$$

The intervals of time where $p(t) \geq 0$ depend on the values of its three coefficients p_1, p_2, and p_3. The possible cases for the behavior of $p(t)$ are discussed in the following paragraphs.

$\mathbf{p_1 \neq 0}$ If $p_1 \neq 0$, then $p(t)$ has degree 2 and represents a parabola. When $p_1 > 0$, the parabola curves upward. The parabola curves downward when $p_1 < 0$. The regions of t, if any, where $p(t) \geq 0$ depend on the values of p_1 and the discriminant p_d where

$$\begin{aligned} p_d &= p_2^2 - p_1p_3 \\ &= (m_1l_2 - m_2l_1)^2 + (m_1l_3 - m_3l_1)^2 - (m_3l_2 - m_2l_3)^2 \end{aligned}$$

The three possibilities $p_d < 0$, $p_d = 0$, and $p_d > 0$ are discussed below.

If $p_d < 0$, then the two roots of $p(t)$ are complex and there are no values of t where the parabola crosses the zero axis. If in addition $p_1 > 0$, then $p(t)$ is always positive and it is possible for $f(t)$ to have zeros for all values of t. In this case, the analytic root interval for $f(t)$ is the entire time axis

$(-\infty, +\infty)$. On the other hand, if $p_1 < 0$ when $p_d < 0$, then $p(t)$ is always negative. If this were the case, it would not be possible for $f(t)$ to equal zero for any value of t. However, it can be shown that it is not possible for the conditions $p_d < 0$ and $p_1 < 0$ to be satisfied simultaneously for the given definitions of p_d and p_1.

Now suppose that $p_d = 0$. Then there is only one real value of t, $t_d = -p_2/p_1$, where $p(t) = 0$. If in addition $p_1 > 0$, then $p(t)$ is positive for all other values of t. Therefore, $p(t) \geq 0 \ \forall \ t$ and $f(t)$ can be zero for any value of t. The analytic root interval then is the entire time axis $(-\infty, +\infty)$. But if $p_1 < 0$ while $p_d = 0$, then $p(t)$ is negative at all values of t other than t_d. In this case, the analytic root interval is the single point t_d.

Finally, suppose that $p_d > 0$. The two real values of t where $p(t) = 0$ are

$$t = \frac{-p_2 \pm \sqrt{p_d}}{p_1}$$

Let t_1 be the root of $p(t)$ defined with the minus sign in front of the square root term and t_2 be the root defined with the plus sign. When $p_1 > 0$, $t_2 > t_1$ and $p(t) \geq 0$ for $t \leq t_1$ and $t \geq t_2$. Then, there are two analytic root intervals for $f(t)$, namely $(-\infty, t_1]$ and $[t_2, \infty)$. When $p_1 < 0$, $t_2 < t_1$ and $p(t) \geq 0$ for $t_2 \leq t \leq t_1$. The single analytic root interval in this case is $[t_2, t_1]$.

$p_1 = 0$ and $p_2 \neq 0$ If $p_1 = 0$ and $p_2 \neq 0$, then $p(t)$ has degree 1 and is the straight line $2p_2t + p_3$. The only value of t where $p(t) = 0$ in this case is $t_l = -p_3/2p_2$. For the case where $p_2 < 0$, $p(t)$ has negative slope and its value is nonnegative for $t \leq t_l$. The analytic root interval for $f(t)$ is then $(-\infty, t_l]$. When $p_2 > 0$, $p(t)$ has positive slope and its value is nonnegative for $t \geq t_l$. The analytic root interval for $f(t)$ in this case is $[t_l, \infty)$.

$p_1 = p_2 = 0$ The last case for $p(t)$ occurs when both p_1 and p_2 are zero. Then, $p(t)$ has degree 0 and is simply equal to the constant p_3. If $p_3 \geq 0$, then $p(t) \geq 0 \ \forall \ t$ and the analytic root interval for $f(t)$ is $(-\infty, \infty)$. If $p_3 < 0$, then $p(t) < 0 \ \forall \ t$ and it is not possible for $f(t)$ to equal zero for any value of t.

Intervals where $f(t)$ may have roots	
$p_1 > 0$	
$p_d \leq 0$	$-\infty < t < \infty$
$p_d > 0$	$-\infty < t \leq t_1$ and $t_2 \leq t < \infty$
$p_1 < 0$	
$p_d < 0$	not possible
$p_d = 0$	$t = t_d$
$p_d > 0$	$t_2 \leq t \leq t_1$
$p_1 = 0$ and $p_2 \neq 0$	
$p_2 < 0$	$-\infty < t \leq t_l$
$p_2 > 0$	$t_l \leq t < \infty$
$p_1 = 0$ and $p_2 = 0$	
$p_3 < 0$	$\emptyset$ (no roots possible)
$p_3 \geq 0$	$-\infty < t < \infty$

Table 4.4: The analytic root intervals for the generalized C-function

The analytic root intervals of $f(t)$ for the cases outlined above are summarized in Table 4.4. The cases where the analytic root interval degenerates to a single point or to the empty set can be shown to correspond to special values of the coefficients of $f(t)$. The behavior of $f(t)$ in these cases will be discussed in detail in the next chapter. For most of the remaining cases, the analytic root intervals are unbounded. This means that additional techniques for bounding the zero search interval(s) are required. Two such techniques, the dynamic spherical approximation and the physical constraints, are developed in the following sections.

4.3 The Dynamic Spherical Approximation

A simple way to obtain finite bounds for the zero search of all C-functions defined by $\mathcal{A}$ and $\mathcal{B}$ is to completely enclose $\mathcal{A}$ and $\mathcal{B}$ by spheres. Any zeros of the C-functions which occur at values of t where the spheres are disjoint

cannot be valid collisions. Thus, the zero search can be confined to those intervals of time, if any, when the sphere enclosing $\mathcal{A}$ overlaps the sphere enclosing $\mathcal{B}$.

In order to find the tightest bounds for the zero search interval, the location of the centers of the two spheres and their radii should be chosen such that the spheres have the minimum radii necessary to contain $\mathcal{A}$ and $\mathcal{B}$. Let the center of the sphere enclosing $\mathcal{B}$ be denoted C^b and let $\mathbf{x}^b$ be the position vector from the origin of the reference coordinate system to C^b. Also, let r_b denote the radius of the sphere enclosing $\mathcal{B}$ centered at C^b. The problem of finding the values of $\mathbf{x}^b$ and r_b for the smallest sphere which completely encloses the stationary (nonconvex) polyhedron $\mathcal{B}$ is a nonlinear optimization problem. Unfortunately, the optimal solution cannot be found analytically. Iterative techniques exist which will converge to the optimal answer in computation time on the order of the number of vertices on the convex hull of $\mathcal{B}$. See, for example, reference [23].

Since this problem is auxiliary to the main goal of developing a collision detection algorithm, these iterative techniques are not used in this thesis. Instead, a simpler method is used to generate a suboptimal solution for $\mathbf{x}^b$ and r_b. The first step for this method is to find the maximum and minimum values of the X, Y, and Z coordinates for the vertices of $\mathcal{B}$. Then, the corresponding components of the vector $\mathbf{x}^b$ are set equal to the arithmetic mean of these maximum and minimum values. This value of $\mathbf{x}^b$ is the center of the smallest rectangular box which contains $\mathcal{B}$ having edges aligned with the reference coordinate axes. This does not necessarily coincide with the center of the smallest sphere which completely encloses $\mathcal{B}$. The minimum radius of the sphere enclosing $\mathcal{B}$ with center at the chosen value for $\mathbf{x}^b$ is the maximum distance from $\mathbf{x}^b$ to a vertex of $\mathcal{B}$. If $\mathbf{b}_j$ is the jth vertex vector of $\mathcal{B}$, then

$$r_b = \max_j \|\mathbf{b}_j - \mathbf{x}^b\|$$

The sphere enclosing $\mathcal{A}$ is chosen so that its center follows the trajectory described by $\mathbf{x}(t)$ in addition to having the minimum radius necessary to completely enclose $\mathcal{A}$ regardless of its orientation. The motion of the center of the sphere enclosing $\mathcal{A}$ is restricted in order to simplify the test determining whether the two spheres overlap by eliminating the need to consider the rotational motion of $\mathcal{A}$. Clearly, the center of the sphere enclosing $\mathcal{A}$ that satisfies the above criteria is located at its center of rotation,

C^a, and the corresponding *minimum* radius is just the maximum distance of a vertex of $\mathcal{A}$ from C^a. Let r_a denote this minimum radius of the sphere enclosing $\mathcal{A}$ with center at C^a. Then,

$$r_a = \max_i \| \mathbf{a}_i \|$$

where by definition the vertex vector $\mathbf{a}_i$ points from C^a to its ith vertex. Any sphere not centered at C^a but moving parallel to C^a must have a radius greater than r_a to completely enclose $\mathcal{A}$.

The quantities r_a, r_b, and $\mathbf{x}^b$ are independent of the motion of $\mathcal{A}$ and depend only on the locations of the vertices of $\mathcal{A}$ and $\mathcal{B}$. Thus, the problem of selecting their values should not be regarded as part of the collision detection process. Rather, it should be considered a part of the process of modelling the shapes of $\mathcal{A}$ and $\mathcal{B}$. The values of these three quantities can be precomputed and stored as part of the data representing a particular $\mathcal{A}$ or $\mathcal{B}$, along with its face normal, edge, and vertex vectors.

The problem now is to determine if the sphere of radius r_a with center at C^a moving along the linear trajectory $\mathbf{x}(t) = \mathbf{x}_o + \mathbf{v}t$ overlaps the stationary sphere of radius r_b with center at C^b and, if so, when. The velocity vector $\mathbf{v}$ is assumed to be nonzero. (The case of static spherical approximation where $\mathbf{v} = \mathbf{0}$ and $\mathbf{x}(t) = \mathbf{x}_o$ will be discussed in Chapter 6.)

The magnitude of the vector $\mathbf{x}(t) - \mathbf{x}^b$ is the distance between the centers of the two spheres at time t. Therefore, the two spheres enclosing $\mathcal{A}$ and $\mathcal{B}$ will overlap whenever

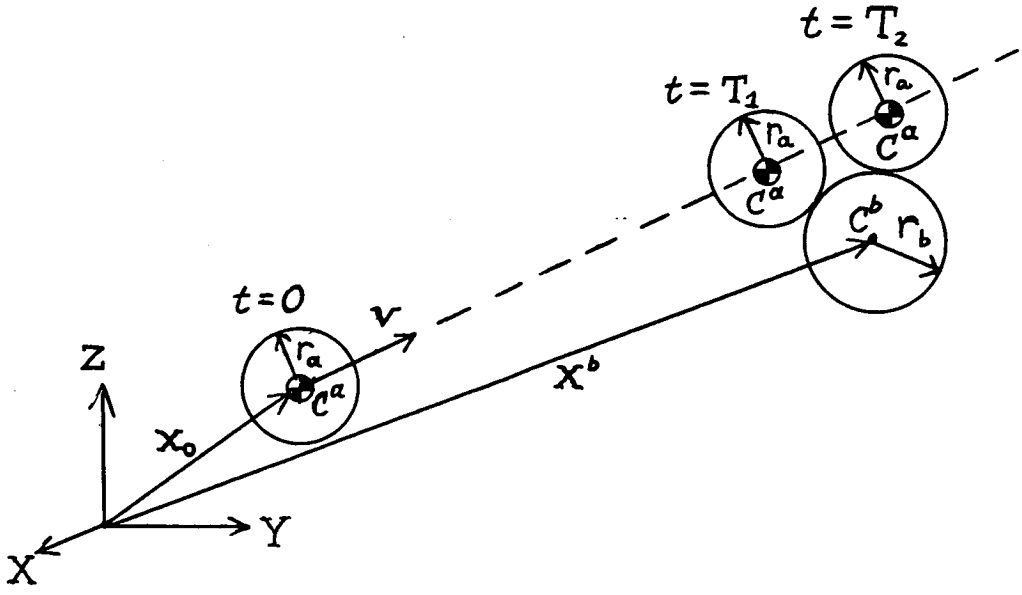
$$\| \mathbf{x}(t) - \mathbf{x}^b \| \leq (r_a + r_b)$$

Substituting the equation for $\mathbf{x}(t)$ into this inequality gives

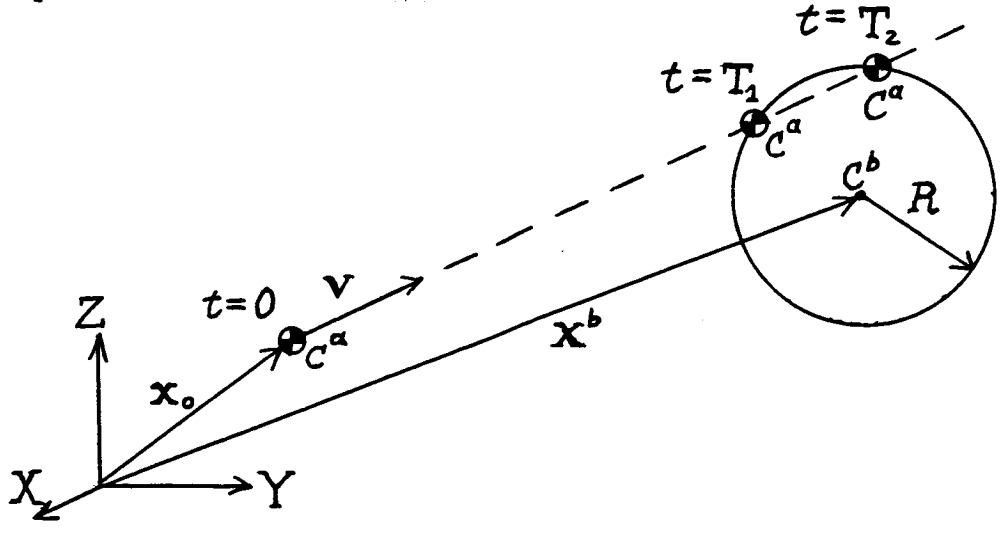
$$\| \mathbf{d}_o + \mathbf{v}t \| \leq (r_a + r_b)$$

where $\mathbf{d}_o = \mathbf{x}_o - \mathbf{x}^b$. The vector $\mathbf{d}_o$ is constant and its magnitude represents the initial distance between the centers of the two spheres. This inequality is also the criterion which determines when C^a moving along the given linear trajectory will lie inside or on the surface of the stationary sphere of radius $R = r_a + r_b$ with center at C^b. These two interpretations of dynamic spherical approximation are illustrated in Figure 4.1.

The inequality above can be transformed into a quadratic inequality. This is a useful form because the objective here is to find the values of t



The sphere enclosing $\mathcal{A}$ overlaps the sphere enclosing $\mathcal{B}$ between $t = T_1$ and $t = T_2$.



C^a lies inside the sphere of radius $R = r_a + r_b$ centered at C^b between $t = T_1$ and $t = T_2$.

Figure 4.1: Two depictions of dynamic spherical approximation

for which overlap of the two spheres occurs. Squaring both sides of the inequality gives

$$\|\mathbf{d}_o + \mathbf{v}t\|^2 = \langle \mathbf{d}_o + \mathbf{v}t, \mathbf{d}_o + \mathbf{v}t \rangle \leq R^2$$

Expanding the squared magnitude of $\mathbf{d}_o + \mathbf{v}t$, subtracting R^2 from both sides, and combining terms of the same degree in t gives

$$\langle \mathbf{v}, \mathbf{v} \rangle t^2 + 2 \langle \mathbf{d}_o, \mathbf{v} \rangle t + \langle \mathbf{d}_o, \mathbf{d}_o \rangle - R^2 \leq 0$$

The graph of the quadratic on the left-hand side of the inequality above is a parabola which curves upward because $\langle \mathbf{v}, \mathbf{v} \rangle > 0$. The three possibilities for the behavior of this quadratic that depend on the value of its discriminant

$$d_s = \langle \mathbf{d}_o, \mathbf{v} \rangle^2 - \langle \mathbf{v}, \mathbf{v} \rangle \left[\langle \mathbf{d}_o, \mathbf{d}_o \rangle - R^2 \right]$$

are presented below. Graphically, these possibilities correspond to the parabola lying entirely above the zero axis, touching the zero axis at its vertex, and intersecting the zero axis at two real values of t.

$\mathbf{d_s < 0}$ First if $d_s < 0$, then there are no real values of t where the quadratic equals zero. The parabola then lies entirely above the zero axis and the quadratic is positive for all values of t. This means that the sphere enclosing $\mathcal{A}$ never overlaps the sphere enclosing $\mathcal{B}$, or equivalently that C^a always lies outside the stationary sphere of radius R centered at C^b. Hence, $\mathcal{A}$ and $\mathcal{B}$ are never close enough to come into contact. No zero search of the C-functions is required in this case because the dynamic spherical approximation has determined that a collision between $\mathcal{A}$ and $\mathcal{B}$ is not possible. This does not mean that there are no values of t where the C-functions equal zero. However, these zeros will not represent a collision and are not of interest.

$\mathbf{d_s = 0}$ Second, if $d_s = 0$, then the quadratic equals zero only when

$$t_v = -\frac{\langle \mathbf{d}_o, \mathbf{v} \rangle}{\langle \mathbf{v}, \mathbf{v} \rangle}$$

In this case, the parabola touches the zero axis only at its vertex t_v. The sphere enclosing $\mathcal{A}$ is tangent to the sphere enclosing $\mathcal{B}$ when $t = t_v$ and

is outside of the sphere enclosing $\mathcal{B}$ for all other values of t. Equivalently, C^a lies on the surface of the stationary sphere of radius R centered at C^b for $t = t_v$ and outside of this sphere for all other values of t. Thus, t_v is the only value of t where $\mathcal{A}$ and $\mathcal{B}$ can possibly come into contact. No zero search of the C-functions is required in this case. Whether or not $\mathcal{A}$ and $\mathcal{B}$ are actually in contact at t_v depends on the orientation of $\mathcal{A}$. A static check of the three collision conditions for the configuration of $\mathcal{A}$ at t_v will determine if t_v is a valid collision time.

$\mathbf{d_s > 0}$ The third and final case is $d_s > 0$. In this case, there are two real values of t where the quadratic equals zero at

$$t = \frac{-\langle \mathbf{d_o}, \mathbf{v} \rangle \pm \sqrt{d_s}}{\langle \mathbf{v}, \mathbf{v} \rangle}$$

Let T_1 be the root of the quadratic defined using the minus sign in the above equation and T_2 be the root defined using the plus sign so that $T_2 > T_1$. The parabola crosses the zero axis at the two points T_1 and T_2. The part of the parabola between these two values of t lies below the zero axis. Thus, the value of the quadratic is nonpositive for $T_1 \leq t \leq T_2$. In this interval, the sphere enclosing $\mathcal{A}$ overlaps the sphere enclosing $\mathcal{B}$. In other words, C^a lies on or inside of the stationary sphere with radius R centered at C^b in this interval. The only physically possible contacts between $\mathcal{A}$ and $\mathcal{B}$ must occur between T_1 and T_2. The zero search of the C-functions is then restricted to the finite interval $[T_1, T_2]$.

4.4 The Physical Constraints

The search for the zeros of the C-functions will only be performed for the case when dynamic spherical approximation gives a nondegenerate search interval $[T_1, T_2]$. This bounded interval applies to the zero searches for all C-functions defined by $\mathcal{A}$ and $\mathcal{B}$. Even though the spheres enclosing $\mathcal{A}$ and $\mathcal{B}$ overlap in this finite interval, it is possible that the two features of $\mathcal{A}$ and $\mathcal{B}$ which define a specific C-function never come into contact for the given trajectory of $\mathcal{A}$. This results from the approximation of $\mathcal{A}$ and $\mathcal{B}$ by spheres where the sizes of the spheres are determined by the maximum "size" of the features of $\mathcal{A}$ or $\mathcal{B}$. Another possibility is that the features defining a

C-function are close enough for contact to occur only within some smaller subinterval of the overall search interval $[T_1, T_2]$.

The physical constraints are introduced to find bounds on the zero search interval for a particular C-function. The results of the physical constraints apply *only* to this C-function, unlike the results of dynamic spherical approximation which apply to *all* C-functions. They are used in conjunction with the dynamic spherical approximation to determine if and when it is possible for the two features defining this C-function to come into contact. Like the dynamic spherical approximation, only the translation of $\mathcal{A}$ is used to check the physical constraints. Whereas the dynamic spherical approximation considers the overall sizes of $\mathcal{A}$ and $\mathcal{B}$, the physical constraints depend only on the "sizes" of the features of $\mathcal{A}$ and $\mathcal{B}$ which define the given C-function.

The procedure for deriving the physical constraints is to find the region surrounding the stationary feature of $\mathcal{B}$ such that a value of $\mathbf{x}(t)$ on or inside of this region represents a position for $\mathcal{A}$ where it is possible for the feature of $\mathcal{A}$ to touch the feature of $\mathcal{B}$. The presence of the other features of $\mathcal{A}$ and $\mathcal{B}$ is ignored in finding this region. The "size" of a feature of $\mathcal{A}$ is represented by the maximum distance from C^a to any point on that feature. The physical constraint region is then defined as the set of points whose minimum distance to points on the feature of $\mathcal{B}$ is less than or equal to this "size" of the feature of $\mathcal{A}$. The finite "size" of the feature of $\mathcal{B}$ insures that this region is bounded.

The physical constraint region is completely contained in the sphere of radius R centered at C^b used for the dynamic spherical approximation. This is because R is defined as the sum of the maximum distances from C^a to any point on the surface of $\mathcal{A}$ and from $\mathbf{x}^b$ to any point on the surface of $\mathcal{B}$.

When C^a is located on or inside the physical constraint region, orientations of $\mathcal{A}$ exist such that the feature of $\mathcal{A}$ touches the feature of $\mathcal{B}$ defining the C-function. Conversely, when C^a is located outside of this region, there is no orientation of $\mathcal{A}$ which can bring these two features into contact. A zero of the C-function for a value t where $\mathbf{x}(t)$ is outside of this region obviously cannot satisfy the third collision condition.

The shape of the region representing the physical constraint depends on the type of the C-function. Once this shape is known, conditions can be formulated that determine if and when it is possible for C^a to lie on or in

the region for its linear trajectory $\mathbf{x}_o + \mathbf{v}t$. The next three sections derive the physical constraint conditions for type (a), (b), and (c) C-functions, respectively.

4.4.1 The Type (a) Physical Constraints

A type (a) contact, represented by a zero of the C-function $f^a_{i,j}$, occurs when some part of face i of $\mathcal{A}$ touches vertex j of $\mathcal{B}$. The physical constraint for this type of C-function determines if and when it is possible for the moving polygon representing face i of $\mathcal{A}$ to touch the stationary point which is the jth vertex of $\mathcal{B}$ for some orientation of $\mathcal{A}$. Any zero of $f^a_{i,j}$ which occurs at a value of t that does not satisfy the physical constraint must occur when vertex j of $\mathcal{B}$ lies in a part of the plane P^a_i containing face i of $\mathcal{A}$ that is outside of the polygon representing the face itself. Such a zero of $f^a_{i,j}$ will not satisfy the third collision condition and, therefore, cannot represent a valid collision.

The "size" of face i of $\mathcal{A}$ is the maximum distance of any point on face i of $\mathcal{A}$ from C^a. This is just the maximum of the magnitudes of the vectors representing the vertices of face i of $\mathcal{A}$. Let this size be denoted r^a_i where

$$r^a_i = \max_k \|\mathbf{a}_k\| \ , \quad k \in \{\text{vertices of } \mathcal{A} \text{ bounding face } i\}$$

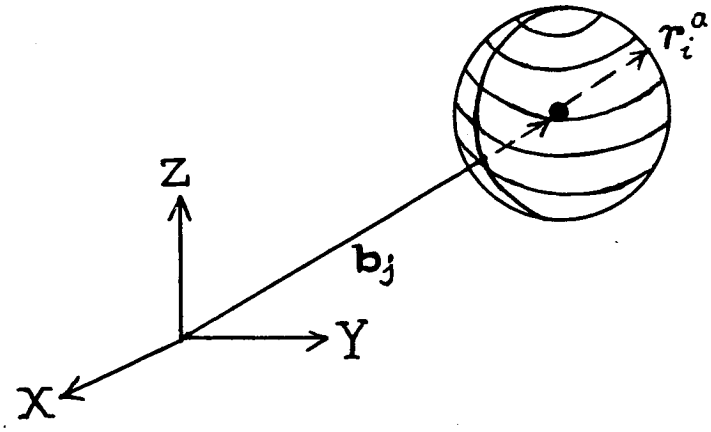
Clearly, it is possible for face i of $\mathcal{A}$ to touch vertex j of $\mathcal{B}$ only if the distance between C^a and vertex j of $\mathcal{B}$ is less than or equal to r^a_i. The set of all points at a distance less than or equal to r^a_i from vertex j of $\mathcal{B}$ is simply the region bounded by the sphere of radius r^a_i centered at vertex j of $\mathcal{B}$. This spherical region representing the type (a) physical constraint is shown in Figure 4.2.

The exact physical constraint inequality for $f^a_{i,j}$ is

$$\|\mathbf{x}(t) - \mathbf{b}_j\| \leq r^a_i$$

where the magnitude of $\mathbf{x}(t) - \mathbf{b}_j$ is obviously the distance of C^a from vertex j of $\mathcal{B}$ at time t. This condition is similar to the inequality used in the dynamic spherical approximation with $\mathbf{x}^b$ replaced by $\mathbf{b}_j$ and R replaced by r^a_i. Substituting the assumed linear trajectory for C^a into this inequality yields

$$\|\mathbf{d}_j + \mathbf{v}t\| \leq r^a_i$$



• Vertex j of $\mathcal{B}$

Figure 4.2: The exact physical constraint region for $f_{i,j}^a$

where $\mathbf{d}_j = \mathbf{x}_o - \mathbf{b}_j$. The vector $\mathbf{d}_j$ is analogous to the vector $\mathbf{d}_o$ defined for the dynamic spherical approximation. The magnitude of $\mathbf{d}_j$ represents the initial distance of C^a from vertex j of $\mathcal{B}$.

As was the case for the dynamic spherical approximation, the type (a) physical constraint can be reduced to the requirement that t be such that the value of a quadratic in t is less than or equal to zero. Following the development for the dynamic spherical approximation, the resulting inequality is

$$\langle \mathbf{v}, \mathbf{v} \rangle t^2 + 2 \langle \mathbf{d}_j, \mathbf{v} \rangle t + \langle \mathbf{d}_j, \mathbf{d}_j \rangle - (r_i^a)^2 \leq 0$$

The same three possibilities exist for the behavior of this quadratic as were discussed for the dynamic spherical approximation. These possibilities are distinguished by the value of the discriminant

$$d_{ij}^a = \langle \mathbf{d}_j, \mathbf{v} \rangle^2 - \langle \mathbf{v}, \mathbf{v} \rangle \left[\langle \mathbf{d}_j, \mathbf{d}_j \rangle - (r_i^a)^2 \right]$$

and are summarized below.

$\mathbf{d_{ij}^a < 0}$ If $d_{ij}^a < 0$, the quadratic is positive for all values of t. Therefore, C^a always lies outside the stationary sphere of radius r_i^a centered at vertex j of $\mathcal{B}$. Note that this case can occur even though C^a passes through the larger sphere of radius R centered at C^b. In this case, face i of $\mathcal{A}$ is never

close enough to vertex j of $\mathcal{B}$ for contact to be possible. Hence, the type (a) C-function $f^a_{i,j}$ can be eliminated from the zero search.

$\mathbf{d}^a_{ij} = 0$ When $d^a_{ij} = 0$, then the quadratic equals zero only at time

$$t_a = -\frac{\langle \mathbf{d}_j, \mathbf{v} \rangle}{\langle \mathbf{v}, \mathbf{v} \rangle}$$

C^a then lies on the surface of the sphere of radius r^a_i centered at $\mathbf{b}_j$ for $t = t_a$ and is outside of this sphere for all other values of t. Thus, t_a is the only value of t where it is possible for face i of $\mathcal{A}$ to touch vertex j of $\mathcal{B}$. Note that t_a will either lie within the overall zero search interval $[T_1, T_2]$ or be equal to t_v, depending on the results from the dynamic spherical approximation. In this case, the value of $f^a_{i,j}$ need only be checked at $t = t_a$. If $f^a_{i,j}(t_a) = 0$, then the second and third collision conditions must be checked for the configuration of $\mathcal{A}$ at t_a to determine if t_a is a valid collision of $\mathcal{A}$ and $\mathcal{B}$.

$\mathbf{d}^a_{ij} > 0$ When $d^a_{ij} > 0$, the two real values of t where the quadratic equals zero are

$$t = \frac{-\langle \mathbf{d}_j, \mathbf{v} \rangle \pm \sqrt{d^a_{ij}}}{\langle \mathbf{v}, \mathbf{v} \rangle}$$

Let t_i be the root of the quadratic defined with the minus sign in the above equation and t_f be the root defined with the plus sign so that $t_f > t_i$. The value of the quadratic is nonpositive for $t_i \leq t \leq t_f$. In this interval, C^a lies on or inside the sphere with radius r^a_i centered at $\mathbf{b}_j$. Any contact between face i of $\mathcal{A}$ and vertex j of $\mathcal{B}$ must occur between t_i and t_f. Therefore, the type (a) C-function $f^a_{i,j}$ must be searched for allowable zeros in the finite interval $[t_i, t_f]$. This interval is a subset of the overall zero search interval $[T_1, T_2]$ determined by the dynamic spherical approximation.

4.4.2 The Type (b) Physical Constraints

A zero of the C-function $f^b_{i,j}$ represents a potential type (b) contact between vertex i of $\mathcal{A}$ and some part of face j of $\mathcal{B}$. The physical constraint for this type of C-function determines if and when it is possible for the moving vertex of $\mathcal{A}$ to touch the stationary polygon which is the face of $\mathcal{B}$. Any

zero of $f^b_{i,j}$ which occurs at a value of t that does not satisfy the physical constraint will occur when vertex i of $\mathcal{A}$ passes through a part of the plane P^b_j containing face j of $\mathcal{B}$ that is outside of the polygon representing the face itself. Such a zero of $f^b_{i,j}$ cannot satisfy the third collision condition and cannot represent a valid collision.

Clearly, there is only one possible choice for the "size" of vertex i of $\mathcal{A}$ namely its distance from C^a. Let this distance be denoted r^b_i, so that $r^b_i = \|\mathbf{a}_i\|$. Contact between vertex i of $\mathcal{A}$ and face j of $\mathcal{B}$ is possible only when the distance between C^a and the point on face j of $\mathcal{B}$ closest to C^a is less than r^b_i. The region that represents this constraint consists of all points whose minimum distance from the polygon representing face j of $\mathcal{B}$ is less than or equal to r^b_i.

The shape of this region is more complicated than the sphere that represented the physical constraint for a type (a) C-function. It can be visualized as the union of disjoint regions of three different kinds. The first region is formed by extending face j of $\mathcal{B}$ for a distance of r^b_i along and opposite to its outward-pointing normal. This results in a solid volume whose cross section is the polygon representing face j of $\mathcal{B}$ and whose length is $2r^b_i$. Face j of $\mathcal{B}$ divides this solid into two equal sections of length r^b_i. The second type of region has the form of half of a right circular cylinder with radius r^b_i and length equal to the length of one of the edges of $\mathcal{B}$ bounding face j. These half-cylinders have their axes along the appropriate edges of face j of $\mathcal{B}$. The third type of region is a section of a sphere having radius r^b_i. These spherical sections are located with their centers at each of the vertices bounding face j of $\mathcal{B}$. The union of these three types of regions forms the type (b) physical constraint region. This construction is illustrated in Figure 4.3 for the case where face j of $\mathcal{B}$ is a square.

Ideally, the physical constraint for $f^b_{i,j}$ would be a set of conditions which determine if and when C^a lies within the region described above for the assumed linear trajectory of $\mathcal{A}$. Because of the shape of this region, these conditions are quite complex. An implementation of these exact constraints could use more time than is saved by further bounding the zero search interval for $f^b_{i,j}$. For this reason, an approximate volume will be suggested that leads to a single condition for the type (b) physical constraint.

Consider the volume containing all points located at a distance less than or equal to r^b_i from plane P^b_j. This volume is bounded by two planes which are parallel to P^b_j and located at a distance r^b_i on either side of it.

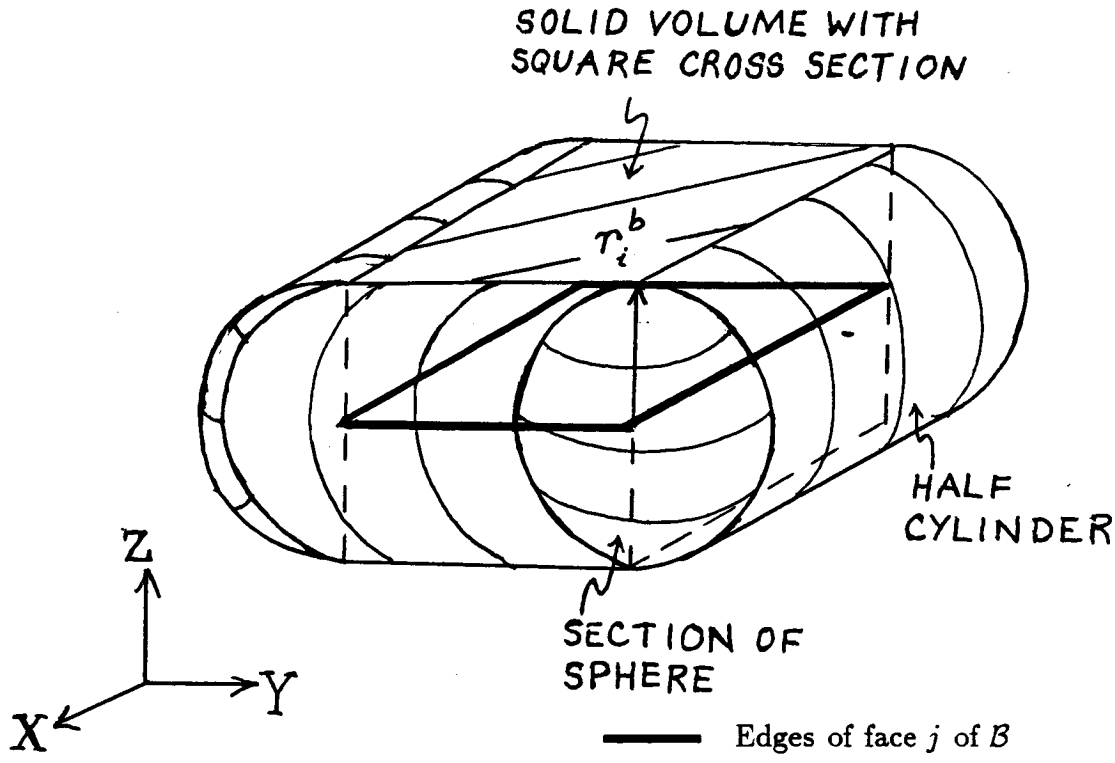


Figure 4.3: The exact physical constraint region for $f_{i,j}^b$

The thickness of this volume, $2r_i^b$, is the same as the length of the first region discussed above. This volume extends indefinitely in all directions parallel to plane P_j^b. In order to obtain a bounded region for the volume approximating the exact type (b) physical constraint region, this infinite volume is intersected with the sphere of radius R centered at C^b. The intersection of these two volumes is shown in Figure 4.4. The resulting approximate region completely contains the region representing the exact physical constraint for $f_{i,j}^b$.

Because of the definition of the new region, there are two constraints that determine if and when C^a lies within this region for its assumed trajectory $\mathbf{x}(t)$. The first constraint determines if and when C^a will lie within the sphere of radius R centered at C^b. This is just the quadratic constraint derived in section 4.3 for the dynamic spherical approximation.

The second constraint determines if and when C^a will lie within the volume surrounding P_j^b constructed above. The inequality that represents this constraint when $\mathcal{A}$ is translating with constant linear velocity will now be derived. Because the volume is bounded by planes parallel to P_j^b, only

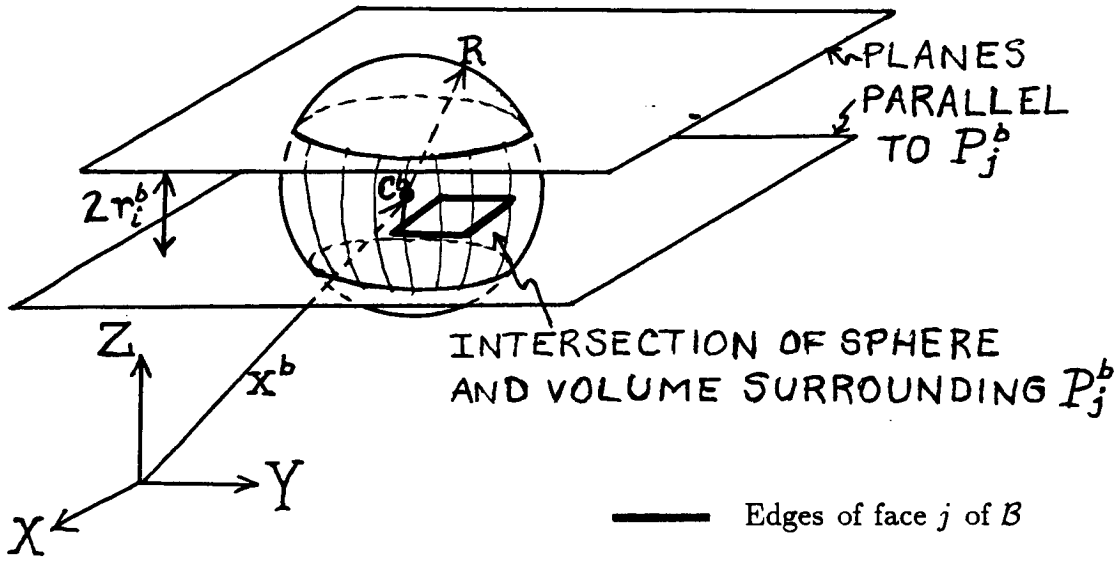


Figure 4.4: The approximate physical constraint region for $f_{i,j}^b$

the motion of C^a perpendicular to P_j^b need be considered. C^a will lie between these planes whenever

$$\left|\left\langle \mathbf{x}(t),\, \mathbf{u}_j^b \right\rangle - d_j^b\right| \leq r_i^b$$

where d_j^b is the distance of P_j^b from the origin of the reference coordinate system as measured along $\mathbf{u}_j^b$. The inner product of $\mathbf{x}(t)$ with $\mathbf{u}_j^b$ represents the distance of C^a from the origin of the reference coordinate system measured along $\mathbf{u}_j^b$. When d_j^b is subtracted from this inner product, the result is the distance of C^a from P_j^b itself measured along $\mathbf{u}_j^b$.

Substituting the equation for $\mathbf{x}(t)$ into the above inequality gives

$$\left|\left\langle \mathbf{x}_o,\, \mathbf{u}_j^b \right\rangle - d_j^b + \left\langle \mathbf{v},\, \mathbf{u}_j^b \right\rangle t\right| \leq r_i^b$$

The first inner product represents the initial distance of C^a from the origin of the reference coordinate system. The second inner product represents the component of the velocity of C^a normal to P_j^b. Its sign determines whether C^a is moving in a direction along $\mathbf{u}_j^b$ or opposite to $\mathbf{u}_j^b$.

When the component of the velocity normal to P_j^b is nonzero, that is when $\langle \mathbf{v}, \mathbf{u}_j^b \rangle \neq 0$, C^a has a component of translation in the direction normal to plane P_j^b. Because of this motion, it must pass through the volume surrounding P_j^b in some finite time interval. In this case, the linear inequality above is satisfied for the finite interval denoted by $[t_i, t_f]$. If $\langle \mathbf{v}, \mathbf{u}_j^b \rangle$ is positive, then

$$t_i = \frac{d_j^b - r_i^b - \left\langle \mathbf{x}_o, \mathbf{u}_j^b \right\rangle}{\left\langle \mathbf{v}, \mathbf{u}_j^b \right\rangle} \quad \text{and} \quad t_f = \frac{d_j^b + r_i^b - \left\langle \mathbf{x}_o, \mathbf{u}_j^b \right\rangle}{\left\langle \mathbf{v}, \mathbf{u}_j^b \right\rangle}$$

On the other hand, if $\langle \mathbf{v}, \mathbf{u}_j^b \rangle$ is negative, then

$$t_i = \frac{d_j^b + r_i^b - \left\langle \mathbf{x}_o, \mathbf{u}_j^b \right\rangle}{\left\langle \mathbf{v}, \mathbf{u}_j^b \right\rangle} \quad \text{and} \quad t_f = \frac{d_j^b - r_i^b - \left\langle \mathbf{x}_o, \mathbf{u}_j^b \right\rangle}{\left\langle \mathbf{v}, \mathbf{u}_j^b \right\rangle}$$

When $\langle \mathbf{v}, \mathbf{u}_j^b \rangle = 0$, C^a does not move in the direction normal to P_j^b. Any motion of C^a is in a direction parallel to P_j^b. Therefore, the distance of C^a from P_j^b measured along $\mathbf{u}_j^b$ remains constant at its initial value. The linear inequality reduces to the requirement that this initial value have magnitude less than or equal to r_i^b:

$$\left| \left\langle \mathbf{x}_o, \mathbf{u}_j^b \right\rangle - d_j^b \right| \leq r_i^b$$

The above constraint will either be satisfied for all values of t or it will not be satisfied for any values of t, depending on the value of $\mathbf{x}_o$. If $\mathbf{x}_o$ satisfies this requirement, then C^a always lies inside the volume surrounding P_j^b. If $\mathbf{x}_o$ does not satisfy this requirement, then C^a never lies within the volume surrounding P_j^b.

The quadratic constraint for the dynamic spherical approximation and the linear constraint for the volume surrounding P_j^b must be satisfied simultaneously for C^a to lie in the region approximating the type (b) physical constraint. The quadratic constraint will be checked first in the collision detection procedure because the results of the dynamic spherical approximation apply to all of the C-functions. The linear constraint that determines if and when C^a lies within the volume surrounding P_j^b depends on the particular C-function $f_{i,j}^b$ and will not be checked until this C-function is examined by the collision detection procedure. Because its results can

only be applied to a single type (b) C-function, this linear constraint will be referred to as the type (b) physical constraint for the remainder of this thesis.

In the discussion that follows, it will be assumed that the result of checking the dynamic spherical approximation constraint is a finite interval $[T_1, T_2]$ restricting the search for the zeros of all the C-functions. If this is not the case, then there is no need for the collision detection procedure to check the physical constraints for any of the C-functions because there is no interval of time where $\mathcal{A}$ and $\mathcal{B}$ are close enough to come into contact.

Suppose that C^a is moving normal to plane P_j^b so that $\langle \mathbf{v}, \mathbf{u}_j^b \rangle \neq 0$. It was previously shown that in this case the type (b) physical constraint is satisfied in a finite interval $[t_i, t_f]$. The interval $[t_i, t_f]$ does not necessarily overlap with the interval determined from the dynamic spherical approximation. If $[t_i, t_f]$ and $[T_1, T_2]$ do overlap, then C^a lies in the approximate type (b) physical constraint region for t in the interval of overlap. Therefore, the zero search for $f_{i,j}^b$ is confined to this interval of overlap. If the two intervals $[t_i, t_f]$ and $[T_1, T_2]$ do not overlap, then C^a never lies in the approximate region representing the type (b) physical constraint and no zero search is necessary for $f_{i,j}^b$.

Now suppose that C^a is moving parallel to plane P_j^b. In this case, the result of the type (b) physical constraint depends on the value of $\mathbf{x}_o$ as explained above. If the type (b) physical constraint is always satisfied, it provides no further bounds on the zero search interval beyond the bounds found from the dynamic spherical approximation. On the other hand, if the type (b) physical constraint is never satisfied for any t, then C^a never lies in the approximate type (b) physical constraint region and $f_{i,j}^b$ need not be searched for its allowable zeros.

4.4.3 The Type (c) Physical Constraints

The type (c) C-function $f_{i,j}^c$ models contact between edge i of $\mathcal{A}$ and edge j of $\mathcal{B}$. The physical constraint for this type of C-function determines if and when it is possible for the moving edge of $\mathcal{A}$ to touch the stationary edge of $\mathcal{B}$. A value of t where $f_{i,j}^c(t) = 0$ that does not satisfy the physical constraint occurs when the intersection point of the lines containing the two edges lies outside of the segments representing edge j of $\mathcal{B}$ and edge i of $\mathcal{A}$. Therefore, the third collision condition cannot be satisfied for such a

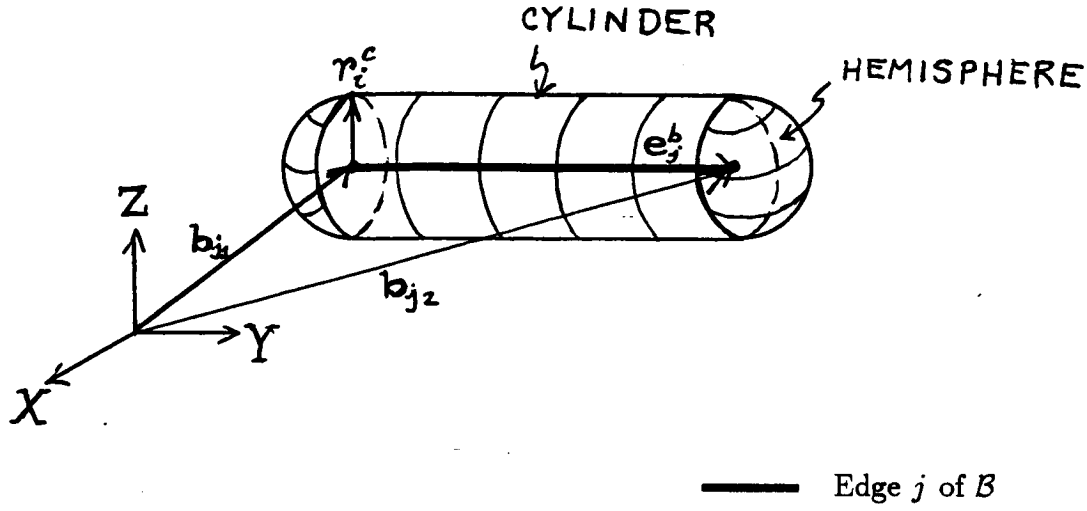


Figure 4.5: The exact physical constraint region for $f_{i,j}^c$

zero of $f_{i,j}^c$.

The "size" of edge i of $\mathcal{A}$ is the maximum distance of any point along edge i of $\mathcal{A}$ from C^a. This is just the larger of the magnitudes of the vectors representing the vertices at the endpoints of edge i. Let this distance be called r_i^c where

$$r_i^c = \max\{\|\mathbf{a}_{i_1}\|, \|\mathbf{a}_{i_2}\|\}$$

In this equation, $\mathbf{a}_{i_1}$ and $\mathbf{a}_{i_2}$ are the vectors representing the vertices of $\mathcal{A}$ at the endpoints of $\mathbf{e}_i^a$. Contact between edge i of $\mathcal{A}$ and edge j of $\mathcal{B}$ is possible only when the distance between C^a and the point on edge j of $\mathcal{B}$ closest to C^a is less than or equal to r_i^c. The region representing this constraint consists of all points whose minimum distance from edge j of $\mathcal{B}$ is less than or equal to r_i^c. This region is the union of three disjoint regions. The first region is a right circular cylinder of radius r_i^c and length equal to the length of edge j of $\mathcal{B}$ whose axis is edge j of $\mathcal{B}$. The remaining two regions are hemispheres with radius r_i^c whose centers are at the two vertices at the endpoints of edge j. The complete region resulting from the union of the cylinder and the two hemispheres is illustrated in Figure 4.5.

As was the case for the exact type (b) physical constraint, the conditions that determine when C^a lies within the region shown in Figure 4.5 are somewhat complicated. Analogous to the type (b) situation, the exact region is approximated by a simpler volume that results in a single inequal-

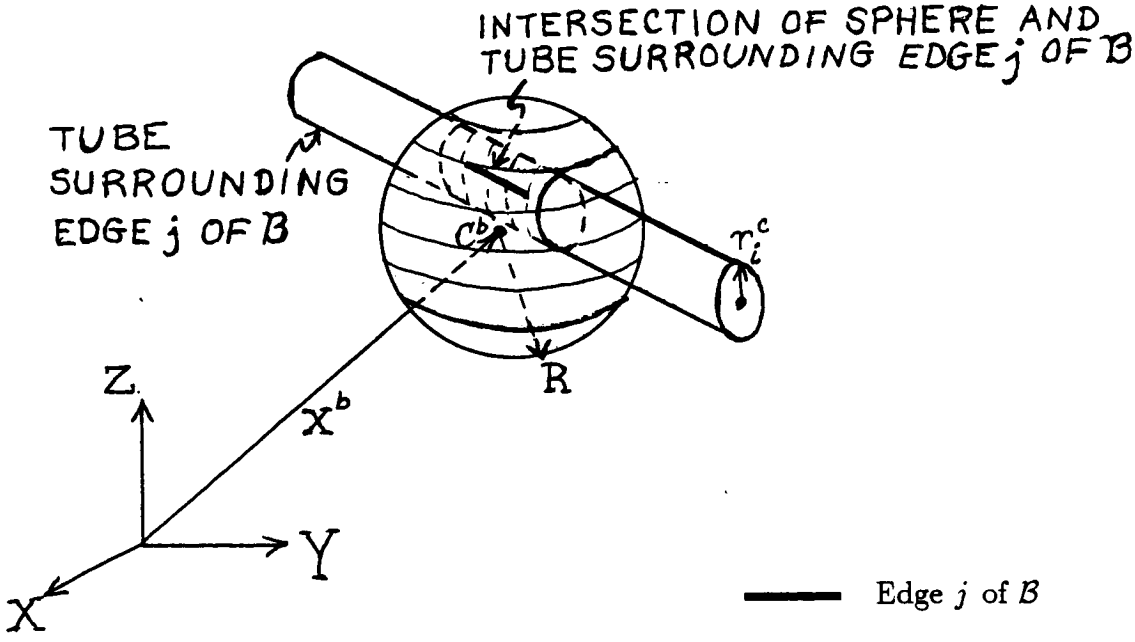


Figure 4.6: The approximate physical constraint region for $f_{i,j}^c$

ity condition for the type (c) physical constraint. The approximate volume for the type (c) physical constraint consists of the intersection of an infinite "tube" with radius r_i^c and axis along the line containing edge j of $\mathcal{B}$ with the sphere of radius R centered at C^b. This intersection is shown in Figure 4.6. The region representing the exact physical constraint for $f_{i,j}^c$ is completely enclosed by this new volume.

There are two separate conditions that must be satisfied simultaneously for C^a to lie within the regions intersecting to form this new volume. As before, the first condition determines if and when C^a lies within the sphere and is represented by the quadratic inequality derived for the dynamic spherical approximation. In this case, the second condition determines if and when C^a lies within the tube around $\mathbf{e}_j^b$. Because the tube has infinite length, this condition depends only on the motion of C^a normal to edge j of $\mathcal{B}$.

Let $\mathbf{u}_{e_j}^b$ be the unit vector pointing along edge j of $\mathcal{B}$ where $\mathbf{u}_{e_j}^b = \mathbf{e}_j^b / \| \mathbf{e}_j^b \|$. The component of any vector $\mathbf{p}$ that is normal to edge j of $\mathcal{B}$ is found by simply subtracting its component along $\mathbf{u}_{e_j}^b$ from the vector itself.

Let this component be denoted by a prime superscript. Then

$$\mathbf{p}' = \mathbf{p} - \left\langle \mathbf{p}, \mathbf{u}^b_{e_j} \right\rangle \mathbf{u}^b_{e_j}$$

Define $\mathbf{x}'$ to be the component of the position vector of C^a normal to $\mathbf{e}^b_j$. Then

$$\mathbf{x}'(t) = \mathbf{x}_o' + \mathbf{v}'t$$

where $\mathbf{x}_o'$ is the component of the initial position of C^a normal to edge j and $\mathbf{v}'$ is the component of its velocity vector normal to edge j.

C^a lies within the tube surrounding edge j of $\mathcal{B}$ whenever the perpendicular distance between it and the line containing edge j of $\mathcal{B}$ is less than or equal to r_i^c. The perpendicular distance from C^a to the line containing edge j is $\|\mathbf{x}'(t) - \mathbf{b}_{j_1}'\|$ where $\mathbf{b}_{j_1}'$ is the component of the vector representing the tail vertex of $\mathbf{e}^b_j$, $\mathbf{b}_{j_1}$, normal to $\mathbf{e}^b_j$. The choice of the tail vertex vector in the above equation is arbitrary. It can be replaced by any vector that terminates on the line containing edge j of $\mathcal{B}$ because all such vectors have the same component normal to $\mathbf{u}^b_{e_j}$.

Now, the constraint for C^a to lie in the tube around $\mathbf{e}^b_j$ becomes

$$\|\mathbf{x}'(t) - \mathbf{b}_{j_1}'\| \leq r_i^c$$

Substituting the expression for $\mathbf{x}'(t)$ into this inequality gives

$$\|\mathbf{d}_{j_1} + \mathbf{v}'t\| \leq r_i^c$$

where $\mathbf{d}_{j_1} = \mathbf{x}_o' - \mathbf{b}_{j_1}'$. This constraint is analogous to the physical constraint for a type (a) C-function with $\mathbf{d}_j$ replaced by $\mathbf{d}_{j_1}$, $\mathbf{v}$ replaced by $\mathbf{v}'$, and r_i^a replaced by r_i^c. It reduces to a requirement on the value of a quadratic in t as shown below:

$$\langle \mathbf{v}', \mathbf{v}' \rangle t^2 + 2 \langle \mathbf{d}_{j_1}, \mathbf{v}' \rangle t + \langle \mathbf{d}_{j_1}, \mathbf{d}_{j_1} \rangle - (r_i^c)^2 \leq 0$$

There are three possible results of this constraint depending on the discriminant of the quadratic on the left-hand side of the inequality above. Let d_{ij}^c be this discriminant where

$$d_{ij}^c = \langle \mathbf{d}_{j_1}, \mathbf{v} \rangle^2 - \langle \mathbf{v}', \mathbf{v}' \rangle \left[\langle \mathbf{d}_{j_1}, \mathbf{d}_{j_1} \rangle - (r_i^c)^2 \right]$$

The three cases distinguished by the value of d_{ij}^c are briefly discussed in the following paragraphs. They correspond to the three cases discussed in section 4.4.1 for the type (a) physical constraint. It is assumed that the dynamic spherical approximation has yielded the finite interval $[T_1, T_2]$ before this constraint is checked.

$\mathbf{d}_{ij}^c < 0$ When $d_{ij}^c < 0$, the quadratic is positive for all values of t. Therefore, C^a always lies outside the tube around edge j of $\mathcal{B}$. Note that this case can occur even though C^a passes through the larger sphere of radius R centered at C^b. Because C^a never lies in the region approximating the type (c) physical constraint, $f_{i,j}^c$ can be eliminated from the zero search.

$\mathbf{d}_{ij}^c = 0$ When $d_{ij}^c = 0$, C^a lies on the surface of the tube only for

$$t_c = -\frac{\langle \mathbf{d}_{j_1}, \mathbf{v}' \rangle}{\langle \mathbf{v}', \mathbf{v}' \rangle}$$

and outside of this tube for all other values of t. The value of t_c is checked to see whether it lies between T_1 and T_2. If t_c does lies within the overall zero search interval $[T_1, T_2]$, C^a lies in the approximate type (c) physical constraint region at $t = t_c$ and the value of $f_{i,j}^c$ need only be checked at this point. If $f_{i,j}^c(t_c) = 0$, then the second and third collision conditions are checked for the configuration of $\mathcal{A}$ at t_c to determine if t_c is a valid collision of $\mathcal{A}$ and $\mathcal{B}$. If t_c does not lie between T_1 and T_2, then C^a never lies in the approximate physical constraint region and $f_{i,j}^c$ can be eliminated from the zero search.

$\mathbf{d}_{ij}^c > 0$ When $d_{ij}^c > 0$, C^a lies on or inside the tube in the finite interval $[t_i, t_f]$ where

$$t_i = \frac{-\langle \mathbf{d}_{j_1}, \mathbf{v}' \rangle - \sqrt{d_{ij}^c}}{\langle \mathbf{v}', \mathbf{v}' \rangle} \quad \text{and} \quad t_f = \frac{-\langle \mathbf{d}_{j_1}, \mathbf{v}' \rangle + \sqrt{d_{ij}^c}}{\langle \mathbf{v}', \mathbf{v}' \rangle}$$

This interval must be checked to determine if it overlaps with the interval $[T_1, T_2]$. If $[t_i, t_f]$ and $[T_1, T_2]$ do overlap, then C^a lies in the approximate type (c) physical constraint region in the interval of overlap and the zero search for $f_{i,j}^c$ is confined to this interval of overlap. If the two intervals do not overlap, then C^a never lies in the approximate physical constraint region and no zero search is necessary for $f_{i,j}^c$.

Recall that the quadratic constraint for the dynamic spherical approximation will be checked first by the collision detection procedure and that its results apply to all of the C-functions. The quadratic constraint that determines if and when C^a will lie within the infinite tube around $\mathbf{e}_j^b$ applies

only to the particular C-function $f^c_{i,j}$ and will be referred to as the type (c) physical constraint for the remainder of this thesis. This constraint for the C-function $f^c_{i,j}$ is only checked if the result of the dynamic spherical approximation is a finite interval $[T_1, T_2]$ in which to search for the zeros of the C-functions.

4.5 Summary of Methods for Bounding the Zero Search

In section 4.1, it was shown that the generalized form of the C-functions, $f(t)$, includes terms that are transcendental in t. The zeros of such a function must be found using a numerical search procedure. Sections 4.2 through 4.4 have presented three methods for bounding the interval of time over which this numerical procedure must be executed. This section will briefly discuss the order in which these various methods will be used by the collision detection procedure.

The dynamic spherical approximation is the first method of bounding the zero search region to be used by the collision detection procedure. The results of this approximation apply to *all* of the C-functions defined by the features of $\mathcal{A}$ and $\mathcal{B}$. When the quadratic inequality for the dynamic spherical approximation is satisfied for a finite interval $[T_1, T_2]$, contact between $\mathcal{A}$ and $\mathcal{B}$ is possible only for values of t in this interval. Therefore, the only zeros of any C-function that can represent valid collisions must occur in this interval. Hence, the zero search for all of the C-functions can be restricted to this interval.

Once an overall search interval $[T_1, T_2]$ has been found from the dynamic spherical approximation, the C-functions are examined individually. The physical constraint and analytic root intervals are checked for each C-function. The physical constraint for a C-function is checked before its analytic root intervals are determined. There are three reasons for this. First, it is easier to check the physical constraint than to determine the analytic root intervals because the physical constraint depends only on the translational motion of $\mathcal{A}$. The analytic root intervals can only be determined after the six coefficients of $f(t)$ for this C-function are found. The values of these coefficients depend on both the translational and rotational motion of $\mathcal{A}$. Second, the analytic root intervals of a C-function are not al-

ways bounded. However, any interval of time where the dynamic spherical approximation constraint and the physical constraint are both satisfied is bounded. The bounds from these two constraints must be available to restrict the zero search to finite subsets of any infinite analytic root intervals. Finally, there is no need to find the analytic root intervals if the physical constraint is not satisfied because any zeros of the C-function cannot represent valid collisions.

A more detailed description of the hierarchy of the methods for bounding the zero search will be given in Chapter 6. This includes a discussion of how the applicability intervals are used in conjunction with the methods presented in this chapter. The procedures used to find the zeros of the C-functions in the resulting finite intervals will be described in the next chapter, Chapter 5.

Chapter 5

Finding the Zeros of the C-functions

The second and third collision conditions have been used in Chapters 3 and 4 to develop techniques for bounding the intervals of time that must be searched for the zeros of the C-functions. The first collision condition requires that the collision detection algorithm locate the zeros of the C-functions in any bounded intervals that result from applying these techniques. All of the C-functions were shown to have the form of the generalized C-function $f(t)$ introduced in section 4.1.6. This chapter presents methods for locating the zeros of $f(t)$.

Twelve cases for the generalized C-function are identified depending on the values of its six coefficients. In some of these cases, $f(t)$ has no zeros and in some other cases, its zeros can be found analytically. However, in general, an analytic solution for the zeros cannot be found. Numerical procedures are developed for these cases to iterate for the zeros of $f(t)$ in a bounded search interval.

The twelve cases for $f(t)$ can be divided into two classes, referred to as classes *I* and *II*, respectively. A C-function falls into class *I* if it satisfies the condition that its coefficients m_2 and m_3 are both zero. All type (b) C-functions are included in class *I*. Class *I* can be subdivided into seven cases. These cases are distinguished by conditions on the values of the four nonzero coefficients m_1, l_1, l_2, and l_3. The seven cases of class *I* C-functions are discussed in section 5.1.

Class *II* C-functions are distinguished by the property that at least one of the coefficients m_2 and m_3 is nonzero. Only type (a) or type (c) C-functions are included in class *II*. Class *II* can be subdivided into five cases. These are discussed in section 5.2.

5.1 Class *I*

The two coefficients m_2 and m_3 are both zero for $f(t)$ in class *I*. This means that terms of the form $t\cos(\omega t)$ and $t\sin(\omega t)$ are absent from such $f(t)$. There are seven possibilities for the behavior of $f(t)$ in this class.

These range from the trivial case where $f(t) \equiv 0$ to the most general case where

$$f(t) = (m_1 t + l_1) + l_2 \cos(\omega t) + l_3 \sin(\omega t)$$

Four of these cases occur when m_1 is zero as well as m_2 and m_3. In these cases, the linear term $m_1 t$ is also missing from $f(t)$. The coefficient m_1 is nonzero for the last three cases so that the linear term $m_1 t$ is present.

The seven cases for $f(t)$ are discussed in detail in the sections below. This includes a confirmation of the analytic root intervals predicted by the criteria of section 4.2. Also, either a formula or a numerical search procedure is given for finding the zeros of $f(t)$ in those cases where zeros are possible. A brief summary of the seven class I cases is given in section 5.1.6.

5.1.1 Case 1

Case 1 of class I for the generalized C-function occurs when all six coefficients of $f(t)$ are zero. In this case, $f(t) = 0 \; \forall \; t$. The analytic root interval for $f(t)$ in this case is the entire time axis because $p_1 = p_2 = p_3 = p(t) = 0$.

When $f(t) \equiv 0$, the configuration of $\mathcal{A}$ is moving along the C-surface defined by $f(t)$ in $C_{space}(\mathcal{A})$. If the initial configuration is located in free space, then it is originally located in a part of the C-surface that is not on the boundary of $\mathbf{CO}_{\mathcal{B}}^{\mathcal{A}}$. Suppose that the trajectory of the configuration of $\mathcal{A}$ moves toward that part of the C-surface which is part of the boundary of $\mathbf{CO}_{\mathcal{B}}^{\mathcal{A}}$. Consider the point where the trajectory of the configuration of $\mathcal{A}$ touches the edge of that part of the C-surface which is on the boundary of $\mathbf{CO}_{\mathcal{B}}^{\mathcal{A}}$. By definition, this edge represents the intersection of the C-surface defined by the C-function that is always zero with one or more C-surfaces defined by other C-functions whose value is not always zero. Therefore, when the trajectory touches this edge, there will be at least one other C-function whose value equals zero at that configuration.

As an example, consider the type (b) C-function $f_{i,j}^b$. If $f_{i,j}^b(t) \equiv 0$, then the trajectory of $\mathcal{A}$ is such that vertex i of $\mathcal{A}$ moves in the plane P_j^b that contains face j of $\mathcal{B}$. This means that the configuration of $\mathcal{A}$ moves along the C-surface defined by $f_{i,j}^b$. If the initial configuration is in free space, then vertex i of $\mathcal{A}$ lies in that part of plane P_j^b that is outside of the polygon representing face j of $\mathcal{B}$. Suppose that the rotational motion of $\mathcal{A}$ is such that $f_{i,j}^b$ is applicable as vertex i moves toward the polygon representing face j of $\mathcal{B}$. Then, when vertex i touches an edge of face j

of $\mathcal{B}$, the configuration of $\mathcal{A}$ touches the edge of that part of the C-surface defined by $f^b_{i,j}$ which lies on the boundary of $\mathbf{CO}^{\mathcal{A}}_{\mathcal{B}}$. But there are several other contacts which occur at this point besides the type (b) contact defined by $f^b_{i,j}$. There is another type (b) contact between vertex i of $\mathcal{A}$ and the other face of $\mathcal{B}$ which meets at the edge of face j of $\mathcal{B}$. There are also type (c) contacts between all of the edges that meet at vertex i of $\mathcal{A}$ and the edge of face j of $\mathcal{B}$. The C-functions representing these other contacts are all equal to zero as well as $f^b_{i,j}$ when vertex i touches an edge of face j of $\mathcal{B}$.

If a C-function whose value is always zero represents a valid collision of $\mathcal{A}$ and $\mathcal{B}$ for some t, the trajectory of the configuration of $\mathcal{A}$ must intersect the edge of that part of the C-surface defined by the C-function that is on the boundary of $\mathbf{CO}^{\mathcal{A}}_{\mathcal{B}}$. The collision that takes place at this intersection point can be detected by checking other C-functions whose values are not always zero. If the trajectory of the configuration of $\mathcal{A}$ always stays in that part of the C-surface defined by a zero-valued C-function that is not part of the boundary of $\mathbf{CO}^{\mathcal{A}}_{\mathcal{B}}$, that C-function does not represent a valid collision of $\mathcal{A}$ and $\mathcal{B}$ for any t.

Thus, a C-function that is always zero can be ignored by the collision detection algorithm.

5.1.2 Case 2

For the second case of class I C-functions, only the coefficient l_1 is nonzero. Both l_2 and l_3 as well as m_1, m_2, and m_3 are zero. In this case, $f(t)$ is a nonzero constant, that is $f(t) = l_1 \neq 0 \; \forall \; t$. Thus, there are no values of t where $f(t)$ equals zero. Clearly, there is no need to search for zeros of $f(t)$. A C-function with constant, nonzero value cannot represent a potential collision of the features of $\mathcal{A}$ and $\mathcal{B}$ which define it.

5.1.3 Cases 3 and 4

At least one of the coefficients l_2 or l_3 is nonzero for the third and fourth cases of class I C-functions. The constant term l_1 can be either zero or nonzero. In general, $f(t)$ has the shifted sinusoidal form

$$f(t) = l_1 + l_2 \cos(\omega t) + l_3 \sin(\omega t)$$

The two sinusoidal terms can be combined into a single term with constant amplitude F and phase angle ϕ_f. The C-function becomes

$$f(t) = l_1 + F\cos(\omega t + \phi_f)$$

where

$$F = \sqrt{l_2^2 + l_3^2} > 0$$

and

$$\begin{array}{rcl} F\cos\phi_f & = & l_2 \\ -F\sin\phi_f & = & l_3 \end{array} \quad \Rightarrow \quad \phi_f = \tan^{-1}\left(\frac{-l_3}{l_2}\right) \in [0, 2\pi)$$

The zeros of $f(t)$ occur at the values of t, if any, where

$$\cos(\omega t + \phi_f) = -\frac{l_1}{F}$$

Whether there are any values of t which satisfy this equation depends on the value of the vertical shift l_1 relative to the amplitude F of the sinusoidal term. The value of $|l_1/F|$ distinguishes the third and fourth cases for class *I*.

For case 3, $|l_1/F| > 1$ or $l_2^2 + l_3^2 - l_1^2 < 0$. In this case, there are no values of t where the shifted sinusoid equals zero. This is because the vertical shift due to l_1 is larger than the amplitude of the sinusoidal component of $f(t)$. No zero search is necessary in this case. A C-function satisfying these conditions cannot represent a valid collision of the features of $\mathcal{A}$ and $\mathcal{B}$ that define it.

For case 4, $|l_1/F| \leq 1$ or $l_2^2 + l_3^3 - l_1^2 \geq 0$. Thus, the vertical shift due to l_1 is smaller than the amplitude of the sinusoidal component of $f(t)$. The zeros of $f(t)$ occur periodically where

$$\omega t = \left\{ \begin{array}{l} \cos^{-1}\left(-\frac{l_1}{F}\right) \\ 2\pi - \cos^{-1}\left(-\frac{l_1}{F}\right) \end{array} \right\} - \phi_f + 2n\pi$$

where n is an integer. Since the C-function has periodic zeros, its analytic root interval is $(-\infty, \infty)$. This can be verified by noting that $p_1 = p_2 = 0$ and $p_3 = l_2^2 + l_3^2 - l_1^2 \geq 0$ in this case. Only those of the periodic zeros that lie in the bounded intervals found from the techniques discussed in Chapters 3 and 4 need be considered by the collision detection algorithm.

Note that the special case where $l_1 = 0$ is included in case 4. In this case, $f(t)$ is the pure sinusoid

$$\begin{aligned} f(t) &= l_2 \cos(\omega t) + l_3 \sin(\omega t) \\ &= F \cos(\omega t + \phi_f) \end{aligned}$$

and its zeros occur periodically where

$$\omega t = \left\{ \begin{array}{c} \pi/2 \\ 3\pi/2 \end{array} \right\} - \phi_f + 2n\pi$$

for n an integer.

5.1.4 Case 5

The fifth case of class I C-functions occurs when the two coefficients l_2 and l_3 are zero but the coefficient m_1 is nonzero. In this case, $f(t)$ is simply the straight line $m_1 t + l_1$. It has one zero at $t = -l_1/m_1$. The values of the coefficients of $p(t)$ are $p_1 = -m_1^2 < 0$, $p_2 = -m_1 l_1$, and $p_3 = -l_1^2$. Therefore, $p_d = p_2^2 - p_1 p_3 = 0$ and the analytic root interval for $f(t) = m_1 t + l_1$ is the single point $t_d = -p_2/p_1 = -l_1/m_1$.

5.1.5 Cases 6 and 7

For the two final cases of class I C-functions, the coefficient m_1 is nonzero and at least one of the two coefficients l_2 and l_3 is also nonzero. The form of the generalized C-function is then:

$$f(t) = (m_1 t + l_1) + l_2 \cos(\omega t) + l_3 \sin(\omega t)$$

Again, the two sinusoidal terms can be combined to obtain

$$f(t) = (m_1 t + l_1) + F \cos(\omega t + \phi_f)$$

where F and ϕ_f are defined in section 5.1.3.

In these cases, $p_1 = -m_1^2 < 0$, $p_2 = -m_1 l_1$, and $p_3 = l_2^2 + l_3^2 - l_1^2$. The value of the discriminant of $p(t)$ is $p_d = m_1^2(l_2^2 + l_3^2) > 0$. Therefore, there is

a finite analytic root interval for $f(t)$, $[t_2, t_1]$. The bounds of this interval are the two roots of $p(t)$:

$$
\begin{aligned}
t_1 &= \frac{m_1 l_1 - |m_1|\sqrt{l_2^2 + l_3^2}}{-m_1^2} = \frac{m_1 l_1 - |m_1| F}{-m_1^2} \\
t_2 &= \frac{m_1 l_1 + |m_1|\sqrt{l_2^2 + l_3^2}}{-m_1^2} = \frac{m_1 l_1 + |m_1| F}{-m_1^2}
\end{aligned}
$$

These values are

$$
\begin{aligned}
t_1 &= \frac{l_1 - F}{-m_1}, \quad t_2 = \frac{l_1 + F}{-m_1} \quad \text{for} \quad m_1 > 0 \\
t_1 &= \frac{l_1 + F}{-m_1}, \quad t_2 = \frac{l_1 - F}{-m_1} \quad \text{for} \quad m_1 < 0
\end{aligned}
$$

The bounds for the analytic root interval can be confirmed by examining the equation for $f(t)$. Since the value of the cosine term is constrained to lie between $-F$ and F, $f(t)$ must be positive whenever $m_1 t + l_1 > F$ and it must be negative whenever $m_1 t + l_1 < -F$. Consider the regions where $t < t_2$ and $t > t_1$. Suppose that m_1 is positive. Then for $t < t_2$, $m_1 t + l_1 < -F$ and $f(t) < 0$ and for $t > t_1$, $m_1 t + l_1 > F$ and $f(t) > 0$. Now suppose that m_1 is negative. Then for $t < t_2$, $m_1 t + l_1 > F$ and $f(t) > 0$ and for $t > t_1$, $m_1 t + l_1 < -F$ and $f(t) < 0$. Therefore, t_1 and t_2 are the bounds of the interval defined by the condition $-F \leq m_1 t + l_1 \leq F$. There must be at least one zero of $f(t)$ in this interval because $f(t)$ has opposite signs for values of t to the left and right of this interval.

5.1.5.1 The First Two Derivatives of $f(t)$ for Cases 6 & 7

The presence of the linear term $m_1 t$ in $f(t)$ for cases 6 and 7 above means that there is no analytic formula for the zeros of such $f(t)$. However, useful information for developing a numerical zero search procedure can be gained by taking the first two derivatives of $f(t)$. In fact, the criterion that distinguishes case 6 from case 7 for the class I C-functions comes from examining the first derivative of $f(t)$.

The first derivative of f with respect to t is

$$
\begin{aligned}
\frac{df}{dt} &= m_1 - \omega l_2 \sin(\omega t) + \omega l_3 \cos(\omega t) \\
&= m_1 - \omega F \sin(\omega t + \phi_f)
\end{aligned}
$$

By taking one derivative of $f(t)$, the linear term in t is eliminated and the form of the first derivative is a shifted sinusoid. The second derivative of f with respect to t is

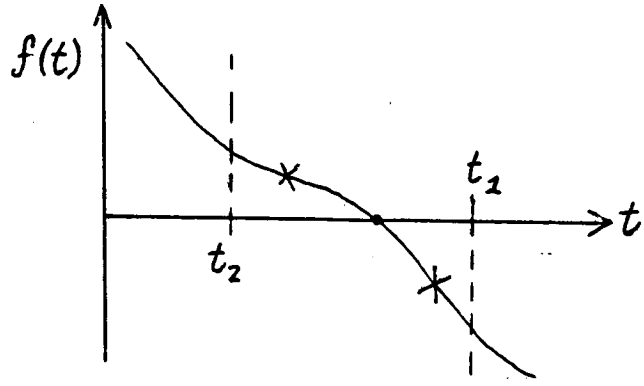
$$\begin{aligned}\frac{d^2f}{dt^2} &= -\omega^2 l_2 \cos(\omega t) - \omega^2 l_3 \sin(\omega t) \\ &= -\omega^2 F \cos(\omega t + \phi_f)\end{aligned}$$

Differentiating a second time has eliminated the constant term so that d^2f/dt^2 is a pure sinusoid.

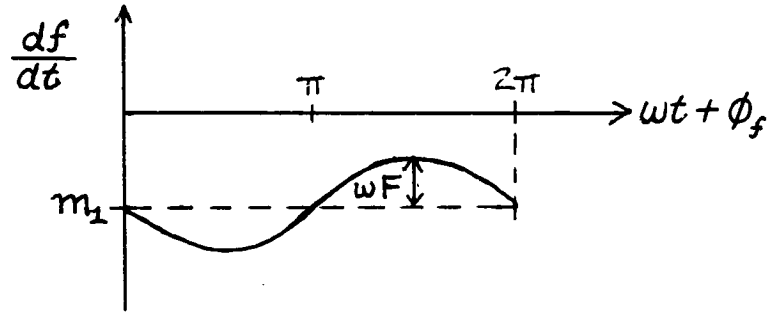
The second derivative of $f(t)$ measures its curvature. The points where the second derivative is zero are the values of t where the curvature of $f(t)$ changes sign. These points occur whenever $\cos(\omega t + \phi_f) = 0$. There are two such points in every cycle of $\omega t + \phi_f$, at $\omega t + \phi_f = \pi/2 + 2n\pi$ and at $\omega t + \phi_f = 3\pi/2 + 2n\pi$. These two points represent points where changes in the sign of the curvature of $f(t)$ from negative to positive and from positive to negative occur, respectively.

The values of t, if any, where the first derivative of $f(t)$ is zero are potential relative maxima and minima of $f(t)$. The form of df/dt is a shifted sinusoid and it is easy to derive the condition under which df/dt can equal zero. For $|m_1/(\omega F)| > 1$, there are no values of t where the first derivative is zero and $f(t)$ cannot have any relative maxima or minima. For $|m_1/(\omega F)| = 1$, the first derivative of $f(t)$ equals zero once in each cycle of $\omega t + \phi_f$ when $\sin(\omega t + \phi_f) = \pm 1$. These are at $\omega t + \phi_f = \pi/2 + 2n\pi$ when m_1 is positive and $\omega t + \phi_f = 3\pi/2 + 2n\pi$ when m_1 is negative. The value of the second derivative of $f(t)$ is zero at these values of $\omega t + \phi_f$. By definition, the single point in a period of $\omega t + \phi_f$ where both the first and second derivatives of $f(t)$ are zero is an inflection point for the C-function. Thus, there are no relative maxima or minima for $f(t)$ in this case.

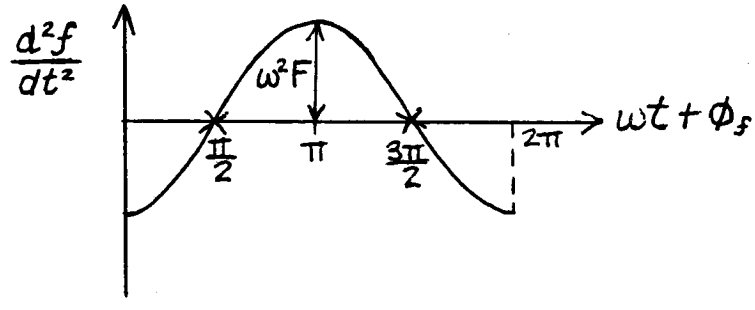
The two cases described above are taken together as case 6 for class I generalized C-functions. The condition that distinguishes this case is $|m_1/(\omega F)| \geq 1$. Since $f(t)$ has no relative maxima or minima, there can only be one zero of $f(t)$ between t_2 and t_1. If $m_1 > 0$, this zero must occur when the value of $f(t)$ changes from negative to positive because $f(t)$ is constantly increasing. On the other hand, if $m_1 < 0$, the single zero must occur when the value of $f(t)$ changes from positive to negative since $f(t)$ is constantly decreasing. The behavior of $f(t)$ in this case is illustrated in Figure 5.1. This figure shows plots of the first and second derivatives of



The generalized C-function $f(t)$ has only one zero between t_2 and t_1.



The first derivative of $f(t)$ for a single period of $\omega t+\phi_f$ when $|m_1/(\omega F)| > 1$ and $m_1 < 0$.



The second derivative of $f(t)$ for a single period of $\omega t + \phi_f$.

$\times$ Change in sign of the curvature of $f(t)$ ($d^2f/dt^2 = 0$)

Figure 5.1: The generalized C-function $f(t)$ for case 6 of class I

$f(t)$ for one period of $\omega t + \phi_f$ along with a sketch of $f(t)$ itself.

For $|m_1/(\omega F)| < 1$, the first derivative of f is zero whenever $\sin(\omega t + \phi_f) = m_1/(\omega F)$. There are two such points in every cycle of $\omega t + \phi_f$ at

$$\delta_1 = \sin^{-1}\left(\frac{m_1}{\omega F}\right) \quad , \quad \delta_2 = \pi - \sin^{-1}\left(\frac{m_1}{\omega F}\right)$$

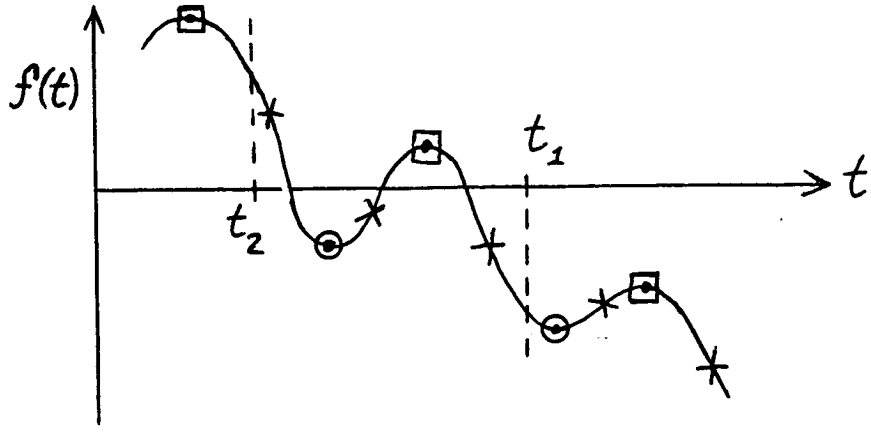
First consider the values of δ_1 and δ_2 when m_1 is positive. In this case, δ_1 lies between 0 and $\pi/2$ while δ_2 lies between $\pi/2$ and π. The value of the second derivative is negative at δ_1 and positive at δ_2. By definition, $f(t)$ has a relative maximum at δ_1 and a relative minimum at δ_2. Now consider the case when m_1 is negative. The value of δ_1 lies in the interval $(3\pi/2, 2\pi)$. This represents a relative maximum of $f(t)$ since d^2f/dt^2 is negative in this interval. The value of δ_2 lies in the interval $(\pi, 3\pi/2)$. The second derivative is positive in this interval and δ_2 is a relative minimum of $f(t)$. Since δ_1 and δ_2 are never exactly equal to $\pi/2$ or $3\pi/2$, $f(t)$ has no inflection points.

The condition $|m_1/(\omega F)| < 1$ distinguishes case 7 for class I generalized C-functions. There can be multiple zeros of $f(t)$ between t_2 and t_1 in this case because $f(t)$ has periodic relative maxima and minima. An example of the behavior of $f(t)$ in this case is illustrated in Figure 5.2. This figure shows plots of $f(t)$ and its first two derivatives. The relative maxima and minima and the points where the curvature changes sign are indicated on these plots.

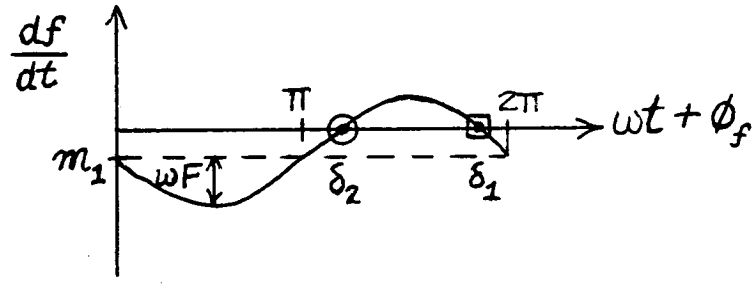
5.1.5.2 The Zero Search Procedure for Cases 6 & 7

One way of finding the zeros of a function in a bounded interval is to sample the function at the bounds of the interval and at selected points within the interval. The values of t at which the function is to be sampled form an ordered sequence of test points beginning with the lower bound of the search interval and ending with the upper bound of the search interval. The values of the function are checked at each pair of consecutive points in this sequence. If the value of the function has opposite sign at two consecutive points, then there is at least one zero of the function located between them. Some iteration method is then used to locate the zeros between these two points.

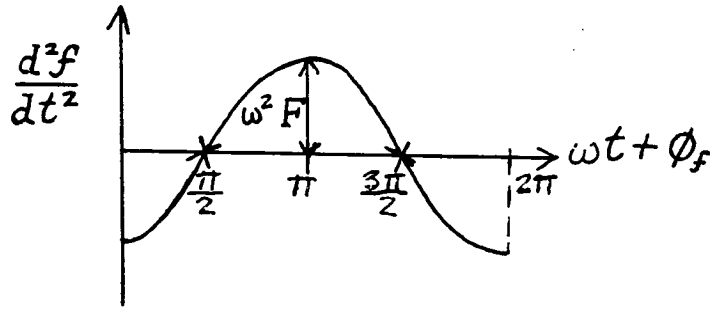
The problem with the method described above is the selection of the test points within the search interval. Ideally, these points should be chosen so that the search procedure does not miss any zeros of the function located



The generalized C-function $f(t)$ can have multiple zeros between t_2 and t_1.



The first derivative of $f(t)$ for a single period of $\omega t + \phi_f$ when $|m_1/(\omega F)| < 1$ and $m_1 < 0$.



The second derivative of $f(t)$ for a single period of $\omega t + \phi_f$.

- $\times$ Change in sign of the curvature of $f(t)$ ($d^2f/dt^2 = 0$)
- $\square$ Relative maximum ($df/dt = 0$, $d^2f/dt^2 < 0$)
- $\odot$ Relative minimum ($df/dt = 0$, $d^2f/dt^2 > 0$)

Figure 5.2: The generalized C-function $f(t)$ for case 7 of class I

in that interval. This could be achieved by finding test points that insure that there is no more than *one* zero of the function in the interval defined by two consecutive points. An iteration method could then be chosen which would be guaranteed to converge to the single zero of the function between two test points that yield values of the function having opposite signs. The ideal sequence of test points would also have the property that a zero of the function occurring exactly at one of the test points would be the only zero of the function in the larger interval defined by its two neighboring points in the sequence.

These conditions are satisfied by the set of test points containing those points in the search interval where the first derivative of the function is zero. All zeros must be located either at a test point or between two consecutive test points because any relative maxima and minima of the function are included in this sequence. The convergence of certain methods chosen to iterate for the zeros is improved by including the points where the second derivative of the function is zero in the set of test points. An explanation for this will be given later when the choice of the iteration method is discussed. Fortunately, it is possible to generate the points where the first or second derivative is zero analytically for cases 6 and 7 of the class I C-functions.

Because the points where the first and second derivatives of $f(t)$ are zero occur periodically with $\omega t + \phi_f$, the zero search interval is divided into subintervals at the points separating cycles of $\omega t + \phi_f$. The bounds of a subinterval will normally be points where $\omega t + \phi_f$ is equal to consecutive multiples of 2π. The upper or lower bound of a subinterval is replaced by the upper or lower bound of the overall search interval when the overall bound lies in the period defined by the current subinterval.

The basic pattern of test points for a subinterval that is a full cycle of $\omega t + \phi_f$ differs for cases 6 and 7. For case 6, only the points where the second derivative is zero are required because either the first derivative cannot equal zero or the points where it equals zero are coincident with points where the second derivative is also zero. The pattern then consists of the two points listed below

$\omega t + \phi_f$	Description
$\pi/2$	curvature sign change from $-$ to $+$
$3\pi/2$	curvature sign change from $+$ to $-$

For case 7, the points where either the first or second derivative is zero are

used. These points are distinct for this case. When m_1 is positive, the basic pattern is

$\omega t + \phi_f$	Description
δ_1	relative maximum
$\pi/2$	curvature sign change from $-$ to $+$
δ_2	relative minimum
$3\pi/2$	curvature sign change from $+$ to $-$

The basic pattern for negative m_1 is

$\omega t + \phi_f$	Description
$\pi/2$	curvature sign change from $-$ to $+$
δ_2	relative minimum
$3\pi/2$	curvature sign change from $+$ to $-$
δ_1	relative maximum

To convert these basic pattern points to the relevant values of t in the period defined by the current subinterval, the appropriate multiple of 2π is added, the phase angle ϕ_f is subtracted, and the result is divided by ω. If the subinterval defines only a partial cycle of $\omega t + \phi_f$, then only those points in the pattern which fall within its bounds are used.

To summarize, the zero search procedure for cases 6 and 7 first divides the overall search interval into whole or partial cycles of $\omega t + \phi_f$. It proceeds as follows in a single subinterval. First, an ordered sequence of test points is generated that includes the bounds of the subinterval and those points in the subinterval where the first or second derivative of $f(t)$ is zero. Next, it examines the values of $f(t)$ at consecutive pairs of test points. If the value of $f(t)$ is zero at a point, then it retains this point as a zero of $f(t)$ and proceeds to the next point. If the value of $f(t)$ changes sign between points, then a single zero of $f(t)$ must be between them. The search procedure then invokes an iteration method for finding this zero of $f(t)$. This process is repeated for each subinterval within the overall search interval.

This search procedure is complete except for the choice of the iteration method for finding a zero of $f(t)$ between two consecutive test points. The two methods considered for this thesis are linear interpolation and its variant, the secant method. For these methods, only values of the function itself are used to find a zero between two starting points where the function values have opposite sign. No evaluation of any derivatives of the function

is required. A brief summary of linear interpolation and the secant method is presented in Appendix B. More detailed descriptions can be found in references [26,27,28].

Linear interpolation is guaranteed to converge to a zero that lies between its starting points if the function is convex between them. A function is convex in a given region if the sign of its second derivative is the same at all points in the region. This is obviously the case for the sequence of test points described above. By including the points where the second derivative is zero among the test points, these points separate each subinterval into regions where $f(t)$ is convex. This is the primary reason for including these points in the sequence for cases 6 and 7. As noted previously, only the relative maxima and minima of $f(t)$ where its first derivative is zero are necessary to insure that no zero of $f(t)$ is missed in case 7. The points where the second derivative is zero are not absolutely necessary in case 6 because it is known that there can be only one zero of $f(t)$ between t_2 and t_1. Linear interpolation could be applied directly with the starting values of t_2 and t_1. However, including the points where the second derivative equals zero insures convergence of the linear interpolation.

Tests using linear interpolation to locate the zeros of sample C-functions in cases 6 and 7 of class I showed that as many as ten or fifteen steps were necessary for convergence in some cases. In other cases, the interpolated approximation to the zero oscillated between two values differing in the least significant digit for the last several iteration steps. When the secant method was used to locate the zeros for these same sample cases, it was found to converge in fewer steps than linear interpolation. Also, no oscillatory behavior was observed using this method. For these reasons, the secant method was chosen as the iteration method for the numerical zero search procedures implemented for this thesis.

Although there is no guarantee of convergence for the secant method in a region where the function is convex, the points where the second derivative is zero are retained in the set of test points. The reason for retaining these points is that the iteration will converge more rapidly when its starting values are closer to the zero. The points where the second derivative equals zero can be included with little additional expense in terms of computation time. Adding these points to the test sequence makes the intervals between consecutive search points smaller and any zero of $f(t)$ is more closely bound.

5.1.6 Summary of Class I

The seven class I cases for $f(t)$ are summarized in Table 5.1. This table gives the conditions holding for each case as well as the special forms of $f(t)$ for each case. Case 1 is the trivial case where $f(t) \equiv 0$ and can be ignored by the collision detection algorithm. There are no zeros of $f(t)$ for cases 2 and 3. There are analytic formulas for finding the zero(s) of $f(t)$ for cases 4 and 5.

All of the C-functions satisfy the conditions for cases 1, 2, and 5 if $\mathcal{A}$ is translating but not rotating. If $\mathcal{A}$ is rotating but not translating, then all of the C-functions qualify for cases 1, 2, 3, or 4. The collision detection algorithm is greatly simplified for rotation only or translation only because the C-functions either have no zeros or they have zeros which can be found analytically.

The zeros of $f(t)$ are found by a numerical search procedure in cases 6 and 7. The search procedure creates a sequence of test points consisting of the bounds of the search interval and any values of t within the interval where the first or second derivative of $f(t)$ equals zero. The value of $f(t)$ is computed at each of the test points. If $f(t)$ changes sign between any two consecutive test points, the secant method is used to iterate for the zero of $f(t)$ located between the two points.

5.2 Class II

The second class for the generalized C-function occurs when $f(t)$ satisfies the condition that either m_2 or m_3 is nonzero. A term of the form of $t\cos(\omega t)$ or $t\sin(\omega t)$ is always present in $f(t)$ for this class.

There are five cases of $f(t)$ in class II. Cases 1 and 2 depend on a special proportionality between the three coefficients m_1, m_2, and m_3 and the three coefficients l_1, l_2, and l_3. This leads to an analytic expression for the zeros of $f(t)$ in these two cases. The proportionality is not satisifed for the other three cases in class II. A numerical search procedure is required to find the zeros of $f(t)$ in these cases.

Cases 3, 4, and 5 are distinguished by the values of m_1 and l_1. The most general case is case 5 where both m_1 and l_1 are nonzero. The second derivative of $f(t)$ for case 5 has a special form that is exploited to develop the numerical zero search procedure. For cases 3 and 4, m_1 is zero. In case

The Seven Class I Cases	
Case	Distinguishing Conditions
1	$m_1 = l_1 = l_2 = l_3 = 0$
2	$m_1 = l_2 = l_3 = 0$, $l_1 \neq 0$
3	$m_1 = 0$, $l_2 \neq 0$ or $l_3 \neq 0$ $\left\lvert l_1/\sqrt{l_2^2 + l_3^2} \right\rvert > 1$
4	$m_1 = 0$, $l_2 \neq 0$ or $l_3 \neq 0$ $\left\lvert l_1/\sqrt{l_2^2 + l_3^2} \right\rvert \leq 1$
5	$m_1 \neq 0$, $l_2 = l_3 = 0$
6	$m_1 \neq 0$, $l_2 \neq 0$ or $l_3 \neq 0$ $\left\lvert m_1/\left(\omega\sqrt{l_2^2 + l_3^2}\right) \right\rvert > 1$
7	$m_1 \neq 0$, $l_2 \neq 0$ or $l_3 \neq 0$ $\left\lvert m_1/\left(\omega\sqrt{l_2^2 + l_3^2}\right) \right\rvert \leq 1$

Class I Forms of $f(t)$	
Case(s)	Form of $f(t)$
1	$f(t) \equiv 0$
2	$f(t) = l_1 \neq 0$
3 & 4	$f(t) = l_1 + l_2 \cos(\omega t) + l_3 \sin(\omega t)$
5	$f(t) = m_1 t + l_1$
6 & 7	$f(t) = (m_1 t + l_1) + l_2 \cos(\omega t) + l_3 \sin(\omega t)$

Table 5.1: The seven cases for class I of the generalized C-function

3, l_1 is also zero. The form of $f(t)$ or its first derivative for these special values of m_1 and l_1 is the same as the form of the second derivative of $f(t)$ in case 5. This leads to possible simplifications in the case 5 numerical zero search procedure when applied to cases 3 and 4.

More detailed descriptions of the five cases in class II are presented in the sections which follow. The last section gives a summary of the conditions defining the five cases and the form of $f(t)$ for each case.

5.2.1 Cases 1 and 2

For the first two cases of class II for the generalized C-function, a constant $c \in \Re$ exists such that

$$l_1 = cm_1 \quad l_2 = cm_2 \quad l_3 = cm_3$$

This includes the special case where $l_1 = l_2 = l_3 = c = 0$.

If the above condition is satisfied, then $f(t)$ can be factored into the product of a linear term and a shifted sinusoidal term. The factored form of $f(t)$ is

$$\begin{aligned} f(t) &= (t+c)\{m_1 + m_2\cos(\omega t) + m_3\sin(\omega t)\} \\ &= (t+c)\{m_1 + M\cos(\omega t + \phi_m)\} \end{aligned}$$

where

$$\begin{aligned} M &= \sqrt{m_2^2 + m_3^3} > 0 \\ \phi_m &= \tan^{-1}(\tfrac{-m_3}{m_2}) \in [0, 2\pi) \end{aligned}$$

Clearly, $f(t)$ has a zero at $t = -c$. The other zeros of $f(t)$, if any, are the zeros of the shifted sinusoidal term.

Cases 1 and 2 of class II are distinguished by the relative values of the vertical shift due to m_1 and the amplitude M of the shifted sinusoidal term. For case 1, $|m_1/M| > 1$ and the shifted sinusoidal component of $f(t)$ never equals zero. The zero at $t = -c$ is the only zero of $f(t)$ for this case. For case 2, $|m_1/M| \leq 1$ and the shifted sinusoidal component of $f(t)$ has periodic zeros. The zeros of the sinusoidal component occur at the values of t where $\cos(\omega t + \phi_m) = -m_1/M$. These points are

$$\omega t = \left\{ \begin{array}{c} \cos^{-1}\left(\frac{m_1}{M}\right) \\ 2\pi - \cos^{-1}\left(\frac{m_1}{M}\right) \end{array} \right\} - \phi_m + 2n\pi$$

where n is an integer. In this case, only those zeros that lie in the bounded search intervals found from the techniques discussed in Chapters 3 and 4 need be considered by the collision detection algorithm.

For case 1 and case 2, the values of p_2 and p_3, defined in section 4.2, become $c(m_2^2 + m_3^2 - m_1^2) = cp_1$ and $c^2(m_2^2 + m_3^2 - m_1^2) = c^2 p_1$, respectively. Therefore, the discriminant $p_d = p_2^2 - p_1 p_3$ is zero. Case 1 corresponds to $p_1 = m_2^2 + m_3^2 - m_1^2 < 0$ where the analytic root interval is the single point $t_d = -p_2/p_1 = -c$. Case 2 corresponds to $p_1 = m_2^2 + m_3^2 - m_1^2 \geq 0$ where the analytic root interval is the entire time axis. The function has zeros in each period of $\omega t + \phi_m$. Thus, the analytical prediction agrees with the conclusion above.

5.2.2 Cases 3, 4, & 5

The remaining three cases of class *II* C-functions do not have a special proportionality of the coefficients of $f(t)$ because there is no constant $c \in \Re$ for which $l_1 = cm_1$, $l_2 = cm_2$, and $l_3 = cm_3$. In these cases, $f(t)$ cannot be factored into a product of simpler terms. No analytic expression for the zeros of $f(t)$ can be found due to the presence of the t, $t\cos(\omega t)$, and/or $t\sin(\omega t)$ terms. As was true for cases 6 and 7 of class *I*, an examination of the first and second derivatives of $f(t)$ gives insight into constructing a numerical zero search procedure.[1] These derivatives are discussed in the following two sections.

5.2.2.1 The First Two Derivatives of $f(t)$ for Case 5

Consider the most general case for class *II* where $m_1 \neq 0$ and $l_1 \neq 0$. Both a constant term and a linear term are present in $f(t)$. These two terms are eliminated by differentiating $f(t)$ twice with respect to t. The first derivative of $f(t)$ with respect to t is

$$\frac{df}{dt} = m_1 + [\omega m_3 t + (m_2 + \omega l_3)]\cos(\omega t) + [-\omega m_2 t + (m_3 - \omega l_2)]\sin(\omega t)$$

[1] The analysis of the second derivative of $f(t)$ and the resulting zero search procedure for these cases were suggested to the author by reference [22].

The second derivative of $f(t)$ with respect to t is

$$\frac{d^2f}{dt^2} = \omega\left\{[-\omega m_2 t + (2m_3 - \omega l_2)]\cos(\omega t) - [\omega m_3 t + (2m_2 + \omega l_3)]\sin(\omega t)\right\}$$

The presence of terms of the form $t\cos(\omega t)$ and $t\sin(\omega t)$ in these two derivatives means that there is no analytic formula for the values of t where they are equal to zero. However, it is possible to analytically determine intervals of t which contain one and only one point where the second derivative is zero. This is explained in the next section.

5.2.2.2 The Second Derivative of $f(t)$ for Case 5

There is one special case for the coefficients of the second derivative of $f(t)$ which must be considered separately. In this case, d^2f/dt^2 can be factored into a linear term and a pure sinusoidal term and the values of t where it is zero can be found analytically. This special case occurs when there exists a constant $c_2 \in \Re$ for which

$$2m_3 - \omega l_2 = c_2(\omega m_2) \quad \text{and} \quad -(2m_2 + \omega l_3) = c_2(\omega m_3)$$

This includes the two cases $m_2 = 2m_3 - \omega l_2 = 0$ and $m_3 = 2m_2 + \omega l_3 = 0$. Multiplying the equation on the left by m_3 and the one on the right by m_2 and subtracting the two resulting equations gives

$$2\left(m_2^2 + m_3^2\right) + \omega\left(m_2 l_3 - m_3 l_2\right) = 0$$

For convenience in the subsequent discussion, let the quantity $2(m_2^2 + m_3^2) + \omega(m_2 l_3 - m_3 l_2)$ be denoted d_ϕ. Thus, the condition $d_\phi = 0$ is another way to define this special case for d^2f/dt^2.

Now, d^2f/dt^2 can be factored into the product of two terms

$$\begin{aligned}\frac{d^2f}{dt^2} &= -\omega^2(t - c_2)\{m_2\cos(\omega t) + m_3\sin(\omega t)\} \\ &= -\omega^2(t - c_2)\{M\cos(\omega t + \phi_m)\}\end{aligned}$$

where M and ϕ_m are defined in section 5.2.1. Therefore, one point where the second derivative is zero is $t = c_2$. The other zeros of the second

derivative occur periodically whenever $\cos(\omega t + \phi_m) = 0$. These zeros are $\omega t + \phi_m = \pi/2 + 2n\pi$ and $\omega t + \phi_m = 3\pi/2 + 2n\pi$ where n is an integer.

When $d_\phi \neq 0$, the expression for d^2f/dt^2 cannot be factored into a product of two simpler terms. However, the sine and cosine terms in the expression for d^2f/dt^2 can be combined into a single cosine term with a time-varying amplitude $F_2(t)$ and phase angle $\phi_2(t)$. The new equation for the second derivative is

$$\frac{d^2f}{dt^2} = \omega \left\{ F_2(t) \cos(\omega t + \phi_2(t)) \right\}$$

where

$$F_2(t) = \sqrt{\left[-\omega m_2 t + (2m_3 - \omega l_2)\right]^2 + \left[\omega m_3 t + (2m_2 + \omega l_3)\right]^2}$$

and

$$\begin{aligned} F_2(t) \cos \phi_2(t) &= -\omega m_2 t + (2m_3 - \omega l_2) \\ F_2(t) \sin \phi_2(t) &= +\omega m_3 t + (2m_2 + \omega l_3) \\ \tan \phi_2(t) &= \frac{+\omega m_3 t + (2m_2 + \omega l_3)}{-\omega m_2 t + (2m_3 - \omega l_2)} \end{aligned}$$

Because there is no proportionality of the coefficients of the second derivative, the two linear terms in the definition of $F_2(t)$ cannot both be zero simultaneously and $F_2(t) > 0 \; \forall \; t$. Now, since ω and $F_2(t)$ are both positive, the second derivative can only be zero at the points where $\cos(\omega t + \phi_2(t)) = 0$. These points are at $\omega t + \phi_2(t) = \pi/2 + 2n\pi$ and $\omega t + \phi_2(t) = 3\pi/2 + 2n\pi$ for n an integer. When $\omega t + \phi_2(t) = \pi/2 + 2n\pi$, the curvature of the function as measured by the value of its second derivative changes from positive to negative. When $\omega t + \phi_2 = 3\pi/2 + 2n\pi$, the curvature of the function changes from negative to positive.

The points where the second derivative of $f(t)$ is zero can also be defined as the points that satisfy the equation

$$\phi_2(t) = -\omega t + \frac{k'\pi}{2}$$

where k' is an odd integer. In other words, the points where $d^2f/dt^2 = 0$ are the points where the curve of $\phi_2(t)$ intersects one of the family of lines $y(t) = -\omega t + k'\pi/2$. An intersection of the $\phi_2(t)$ curve with one of the lines where $k' \in \{\ldots, -9, -5, -1, +1, +5, +9, \ldots\}$ represents a change

in curvature from positive to negative while an intersection with one of the lines $y(t)$ where $k' \in \{\ldots, -11, -7, -3, +3, +7, +11, \ldots\}$ represents a change in curvature from negative to positive. By examining the behavior of $\phi_2(t)$, the interval(s) of ωt where it will intersect a particular one of these lines $y(t)$ can be predicted.

First, consider the behavior of $\phi_2(t)$ as $t \to -\infty$ and $t \to +\infty$. As $t \to -\infty$, the value of the cosine of $\phi_2(t)$ approaches the constant value $m_2/\sqrt{m_2^2 + m_3^2}$ and the value of the sine of $\phi_2(t)$ approaches the constant value $-m_3/\sqrt{m_2^2 + m_3^2}$. Therefore, the value of $\phi_2(t)$ approaches a constant angle denoted β_1 where

$$\begin{aligned} \cos\beta_1 &= \frac{m_2}{\sqrt{m_2^2 + m_3^2}} \\ \sin\beta_1 &= \frac{-m_3}{\sqrt{m_2^2 + m_3^2}} \end{aligned} \quad \Rightarrow \quad \tan\beta_1 = \frac{-m_3}{m_2} \in [0, 2\pi)$$

As $t \to +\infty$, the cosine of the phase angle $\phi_2(t)$ approaches the constant value $-m_2/\sqrt{m_2^2 + m_3^2}$ and the sine of $\phi_2(t)$ approaches the constant value $m_3/\sqrt{m_2^2 + m_3^2}$. Therefore, the value of $\phi_2(t)$ approaches a constant angle denoted β_2 where

$$\begin{aligned} \cos\beta_2 &= \frac{-m_2}{\sqrt{m_2^2 + m_3^2}} \\ \sin\beta_2 &= \frac{m_3}{\sqrt{m_2^2 + m_3^2}} \end{aligned} \quad \Rightarrow \quad \tan\beta_2 = \frac{m_3}{-m_2} \in [0, 2\pi)$$

From the equations defining β_1 and β_2 it is clear that $\beta_2 = \beta_1 \pm \pi$.

The phase angle $\phi_2(t)$ is either increasing or decreasing between β_1 and β_2 as t goes from $-\infty$ to $+\infty$. To determine whether ϕ_2 is decreasing or increasing, consider its first derivative with respect to t. The derivative of $\phi_2(t)$ can be found implicitly from any of the three equations which define its sine, cosine, or tangent. The details of this calculation are omitted. The final result is

$$\frac{d\phi_2}{dt} = \frac{\omega\left[2\left(m_2^2 + m_3^2\right) + \omega\left(m_2 l_3 - m_3 l_2\right)\right]}{\left[-\omega m_2 t + \left(2m_3 - \omega l_2\right)\right]^2 + \left[\omega m_3 t + \left(2m_2 + \omega l_3\right)\right]^2} = \frac{\omega d_\phi}{F_2^2(t)}$$

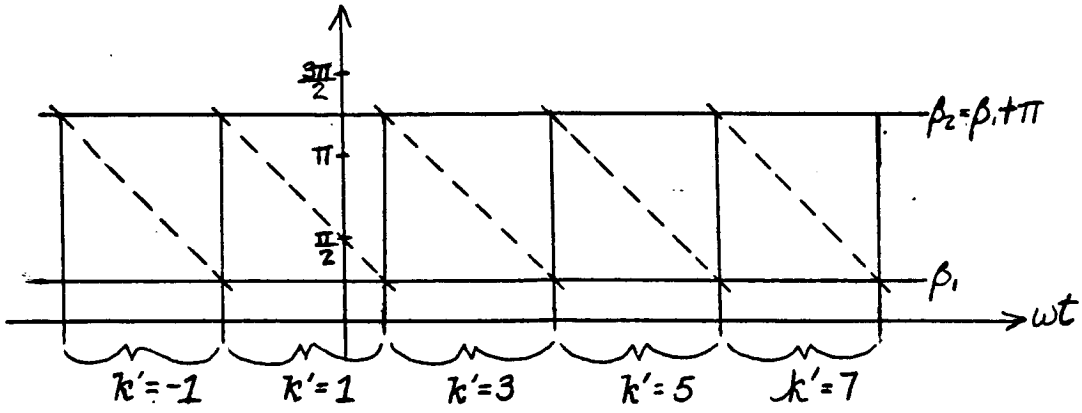
Note that t appears only in the denominator of the expression for $d\phi_2/dt$. Therefore, as $t \to \pm\infty$, $d\phi_2/dt \to 0$. This is expected since $\phi_2(t)$ approaches the constant value β_1 as $t \to -\infty$ and the constant value β_2 as $t \to +\infty$.

In the cases under consideration, $d_\phi \neq 0$ and the expression in the denominator, $F_2^2(t)$, is always positive. Therefore, the sign of d_ϕ determines the sign of $d\phi_2/dt$. When $d_\phi > 0$, the first derivative of ϕ_2 is always positive and $\phi_2(t)$ increases from β_1 to β_2. It is possible to choose the values of β_1 and β_2 so that the curve of $\phi_2(t)$ is continuously increasing from β_1 to $\beta_2 = \beta_1 + \pi$. When $d_\phi < 0$, the first derivative of ϕ_2 is always negative and $\phi_2(t)$ decreases from β_1 to β_2. It is possible to choose the values of β_1 and β_2 so that the curve of $\phi_2(t)$ is continuously decreasing from β_1 to $\beta_2 = \beta_1 - \pi$. The choice of the values of β_1 and β_2 for a continuous curve of $\phi_2(t)$ depends on the two coefficients m_2 and m_3 and the quantity d_ϕ. The sixteen possible cases are presented in Appendix A.

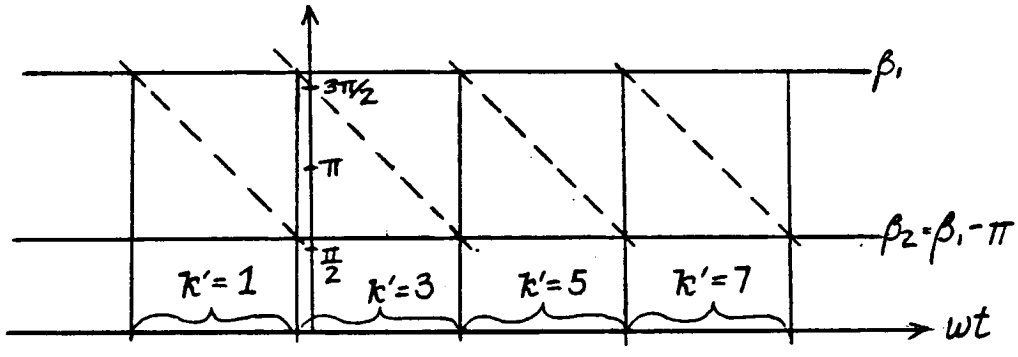
Whether $\phi_2(t)$ is increasing or decreasing, its range of values is constrained to lie in the interval of length π bounded by the values of β_1 and β_2. All intersections of the curve $\phi_2(t)$ with the lines $y(t) = -\omega t + k'\pi/2$ must occur along the segments of these lines which lie between $y(t) = \beta_1$ and $y(t) = \beta_2 = \beta_1 \pm \pi$. These points divide the ωt axis into intervals of length π as shown in Figure 5.3. When $\phi_2(t)$ is increasing, these intervals are bounded by the points $\omega t = -\beta_1 - \pi + k'\pi/2$ and $\omega t = -\beta_1 + k'\pi/2$. When $\phi_2(t)$ is decreasing, these intervals are bounded by $\omega t = -\beta_1 + k'\pi/2$ and $\omega t = -\beta_1 + \pi + k'\pi/2$. Because $\phi_2(t)$ is a single-valued function of t, its curve will pass through each of these intervals sequentially for $k' \in \{\ldots, -3, -1, +1, +3, \ldots\}$. That is, once the curve crosses into an interval defined by a particular value of k' it is not possible for it to return to an interval defined by a smaller value of k'.

When $\phi_2(t)$ is increasing, its slope as measured by the value of its first derivative always has opposite sign from the negative slope $-\omega$ of the lines $y(t) = -\omega t + k'\pi/2$. This means that one and only one intersection of the curve of $\phi_2(t)$ with the line $y(t) = -\omega t + k'\pi/2$ occurs in each interval $[-\beta_1 - \pi + k'\pi/2, -\beta_1 + k'\pi/2]$. The value of t where this intersection takes place is the only zero of the second derivative of $f(t)$ in that interval. This is illustrated in Figure 5.4.

When $\phi_2(t)$ is decreasing, its slope as measured by the value of its first derivative always has the same sign as the negative slope $-\omega$ of the lines $y(t) = -\omega t + k'\pi/2$. It is possible for the curve of $\phi_2(t)$ to intersect one of the lines more than once in an interval $[-\beta_1 + k'\pi/2, -\beta_1 + \pi + k'\pi/2]$. Whether or not multiple intersections will actually occur depends on the value of $d\phi_2/dt$ relative to $-\omega$. In particular, multiple intersections are



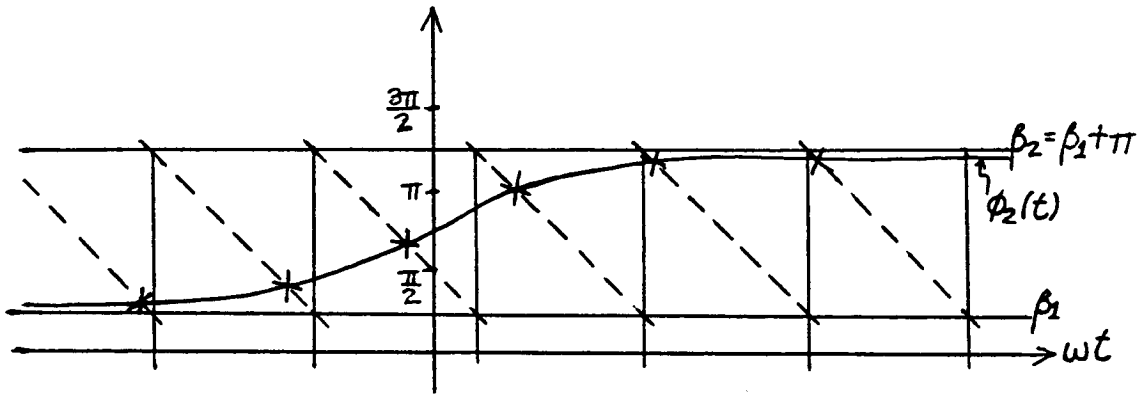
The intervals $[-\beta_1 - \pi + k'\pi/2, -\beta_1 + k'\pi/2]$ when $\phi_2(t)$ increases from β_1 to $\beta_2 = \beta_1 + \pi$.



The intervals $[-\beta_1 + k'\pi/2, -\beta_1 + \pi + k'\pi/2]$ when $\phi_2(t)$ decreases from β_1 to $\beta_2 = \beta_1 - \pi$.

— — — — $y(t) = -\omega t + k'\pi/2$ for k' odd

Figure 5.3: The intervals defined by $y(t) = \beta_1$ and $y(t) = \beta_2$



× Sign change for curvature of $f(t)$ ($d^2f/dt^2 = 0$)
– – – – $y(t) = -\omega t + k'\pi/2$ for k' odd

Figure 5.4: The intersections of $\phi_2(t)$ and the lines $y(t)$ for increasing $\phi_2(t)$

possible only when $d\phi_2/dt$ is less than or equal to $-\omega$ for some interval of t. From the equation for $d\phi_2/dt$ given above, this is possible only when

$$\frac{\omega\left[2\left(m_2^2+m_3^2\right)+\omega\left(m_2l_3-m_3l_2\right)\right]}{\left[-\omega m_2t+\left(2m_3-\omega l_2\right)\right]^2+\left[\omega m_3t+\left(2m_2+\omega l_3\right)\right]^2}\leq-\omega$$

Multiplying both sides by the denominator and bringing all terms to the left of the inequality gives

$$\begin{aligned}\left[-\omega m_2t+\left(2m_3-\omega l_2\right)\right]^2+\left[\omega m_3t+\left(2m_2+\omega l_3\right)\right]^2+\\ 2\left(m_2^2+m_3^2\right)+\omega\left(m_2l_3-m_3l_2\right)\leq 0\end{aligned}$$

Expanding the squared terms and combining terms of the same degree in t results in

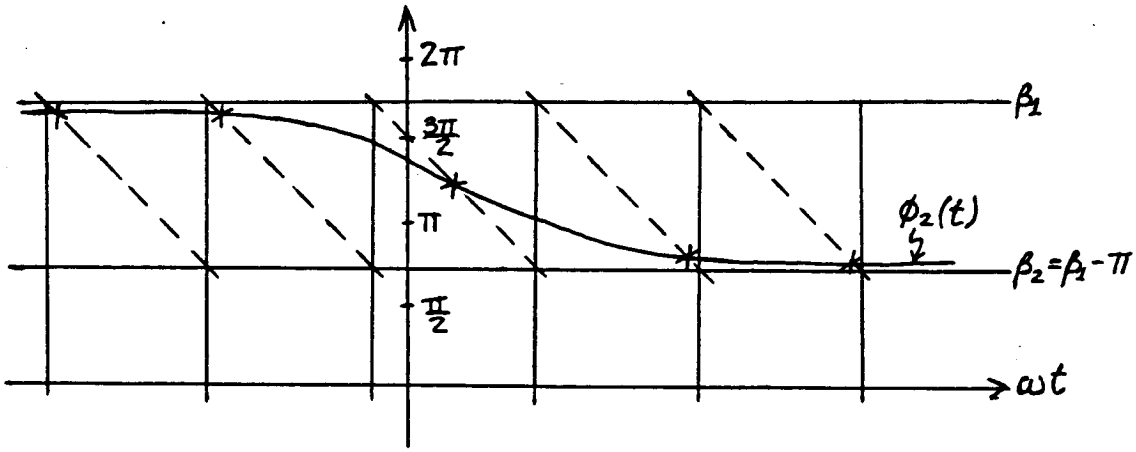
$$\begin{aligned}\omega^2\left(m_2^2+m_3^2\right)t^2+2\omega^2\left(m_2l_2+m_3l_3\right)t+\\ \left[6\left(m_2^2+m_3^2\right)+5\omega\left(m_2l_3-m_3l_2\right)+\omega^2\left(l_2^2+l_3^2\right)\right]\leq 0\end{aligned}$$

The quadratic on the left-hand side of this inequality represents a parabola which curves upward because the coefficient of the t^2 term is positive. Again, there are three possibilities depending on the discriminant of the quadratic. The value of this discriminant is easily computed from the coefficients of the quadratic and an expression for it will not be given here.

The first possibility is that the value of the discriminant is less than zero so that the quadratic has no real roots. In this case, the parabola lies entirely above the zero axis and the value of the quadratic is always positive. The inequality is never satisfied and $-\omega < d\phi_2/dt < 0\ \forall\ t$. The tangent to the curve of $\phi_2(t)$ is then never parallel to the lines $y(t) = -\omega t + k'\pi/2$.

The second possibility is that the value of the discriminant equals zero and the quadratic has one real root. This means the the parabola touches the zero axis at one point and the value of the quadratic is positive for all other values of t. The inequality is satisfied at the single root of the quadratic and $-\omega \leq d\phi_2/dt < 0\ \forall\ t$. The tangent to the curve of $\phi_2(t)$ is parallel to the lines $y(t)$ for the one value of t where $-\omega = d\phi_2/dt$.

In either of these cases, there can be one and only one intersection of the curve of $\phi_2(t)$ with one of the lines $y(t)$ in each interval $[-\beta_1+k'\pi/2, -\beta_1+\pi+k'\pi/2]$. These cases are illustrated in Figure 5.5.



× Sign change for curvature of $f(t)$ ($d^2f/dt^2 = 0$)
— — — — $y(t) = -\omega t + k'\pi/2$ for k' odd

Figure 5.5: The intersections of $\phi_2(t)$ and the lines $y(t)$ for decreasing $\phi_2(t)$ when multiple intersections cannot occur

The last possibility is that the value of the discriminant is positive and the quadratic has two real roots. The parabola then crosses the zero axis at these two roots and the value of the quadratic is negative between them. The inequality is satisfied in the interval bounded by the two roots of the quadratic. Let $\tau_1 < \tau_2$ denote the values of the two real roots of the quadratic. Then, the tangent to the curve of $\phi_2(t)$ is parallel to the lines $y(t)$ at both τ_1 and τ_2. Between τ_1 and τ_2, the tangent to the curve of $\phi_2(t)$ has a slope less than $-\omega$. In this case, the curve of $\phi_2(t)$ *may* intersect one of the lines $y(t)$ more than once in one of the intervals $[-\beta_1 + k'\pi/2, -\beta_1 + \pi + k'\pi/2]$.

The locations of τ_1 and τ_2 determine whether or not multiple intersections will actually occur. If the curve of $\phi_2(t)$ is to intersect a line $y(t)$ more than once, then both $\omega\tau_1$ and $\omega\tau_2$ must lie in the same interval $[-\beta_1 + k'\pi/2, -\beta_1 + \pi + k'\pi/2]$. That is, there must be one value $k'_1 \in \{\ldots, -3, -1, +1, +3, \ldots\}$ for which

$$-\beta_1 + k'_1\pi/2 < \omega\tau_1 < \omega\tau_2 < -\beta_1 + \pi + k'_1\pi/2$$

However, this is not a sufficient condition to insure that multiple intersections take place in that interval. The additional conditions which will determine the number and location of the intersection points are derived in the next paragraph.

If τ_1 and τ_2 are both in the same interval, then there are three ways by which the curve of $\phi_2(t)$ can intersect the line $y(t)$ more than once. Let $\phi_2(\tau_1)$ and $\phi_2(\tau_2)$ be the values of the phase angle at the roots of the quadratic τ_1 and τ_2, respectively. When

$$\begin{array}{rcccl} \phi_2(\tau_1) & = & y(\tau_1) & = & -\omega\tau_1 + k'_1\pi/2 \\ \phi_2(\tau_2) & < & y(\tau_2) & = & -\omega\tau_2 + k'_1\pi/2 \end{array}$$

there are two intersections of the curve of $\phi_2(t)$ with the line $y(t)$. The first of these occurs at $t = \tau_1$ where the tangent to the $\phi_2(t)$ curve is coincident with the line $y(t)$. The second occurs between $\omega\tau_2$ and the end of the interval $-\beta_1 + \pi + k'_1\pi/2$. Similarly, when

$$\begin{array}{rcccl} \phi_2(\tau_1) & > & y(\tau_1) & = & -\omega\tau_1 + k'_1\pi/2 \\ \phi_2(\tau_2) & = & y(\tau_2) & = & -\omega\tau_2 + k'_1\pi/2 \end{array}$$

there are two intersections of the curve of $\phi_2(t)$ with the line $y(t)$. The first intersection point lies between the beginning of the interval $-\beta_1 + k'_1\pi/2$

and $\omega\tau_1$. The second intersection point is $t = \tau_2$ where the tangent to the $\phi_2(t)$ curve is coincident with the line $y(t)$. Now, when

$$\begin{array}{rcrcl} \phi_2(\tau_1) & > & y(\tau_1) & = & -\omega\tau_1 + k_1'\pi/2 \\ \phi_2(\tau_2) & < & y(\tau_2) & = & -\omega\tau_2 + k_1'\pi/2 \end{array}$$

there are three intersections of the curve of $\phi_2(t)$ with the line $y(t)$. The first intersection point falls between the beginning of the interval $-\beta_1 + k_1'\pi/2$ and $\omega\tau_1$, the second intersection points falls between $\omega\tau_1$ and $\omega\tau_2$, and the third intersection point falls between $\omega\tau_2$ and the end of the interval $-\beta_1 + \pi + k_1'\pi/2$. These three ways for multiple intersections to take place are illustrated in Figure 5.6.

The two cases where τ_1 and τ_2 are both in the same interval but multiple intersections do not occur are

$$\begin{array}{rcrcl} \phi_2(\tau_1) & < & y(\tau_1) & = & -\omega\tau_1 + k_1'\pi/2 \\ \phi_2(\tau_2) & < & y(\tau_2) & = & -\omega\tau_2 + k_1'\pi/2 \end{array}$$

and

$$\begin{array}{rcrcl} \phi_2(\tau_1) & > & y(\tau_1) & = & -\omega\tau_1 + k_1'\pi/2 \\ \phi_2(\tau_2) & > & y(\tau_2) & = & -\omega\tau_2 + k_1'\pi/2 \end{array}$$

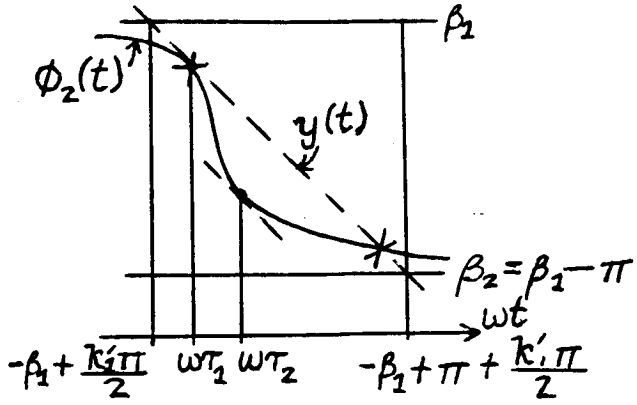
There is only one intersection of the curve of $\phi_2(t)$ and the line $y(t)$ in the interval $[-\beta_1+k_1'\pi/2, -\beta_1+\pi+k_1'\pi/2]$ in these cases. The single intersection point is located between $\omega\tau_2$ and the end of the interval $-\beta_1 + \pi + k_1'\pi/2$ when $\phi_2(\tau_1)$ and $\phi_2(\tau_2)$ lie below the line $y(t)$. When $\phi_2(\tau_1)$ and $\phi_2(\tau_2)$ lie above the line $y(t)$, the single intersection point is located between the beginning of the interval $-\beta_1 + k_1'\pi/2$ and $\omega\tau_1$. These two cases are shown in Figure 5.7.

In summary, when $d_\phi < 0$ and $\phi_2(t)$ is decreasing, it is possible for two or three intersections to occur between the curve of $\phi_2(t)$ and a line $y(t) = -\omega t + k_1'\pi/2$ in one particular interval $[-\beta_1+k_1'\pi/2, -\beta_1+\pi+k_1'\pi/2]$. The three conditions that determine if multiple intersections can occur and that identify the interval where they will occur are:

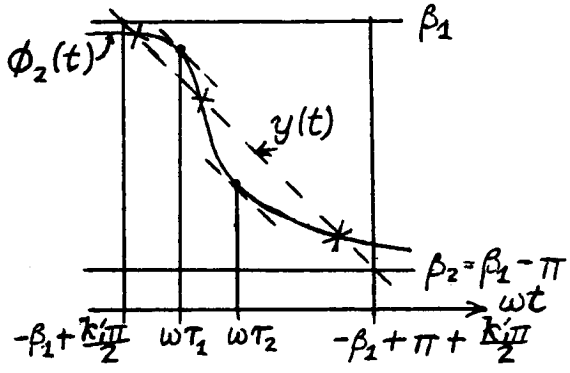
1. The quadratic

$$\omega^2\left(m_2^2 + m_3^2\right)t^2 + 2\omega^2\left(m_2 l_2 + m_3 l_3\right)t + \left[6\left(m_2^2 + m_3^2\right) + 5\omega\left(m_2 l_3 - m_3 l_2\right) + \omega^2\left(l_2^2 + l_3^2\right)\right]$$

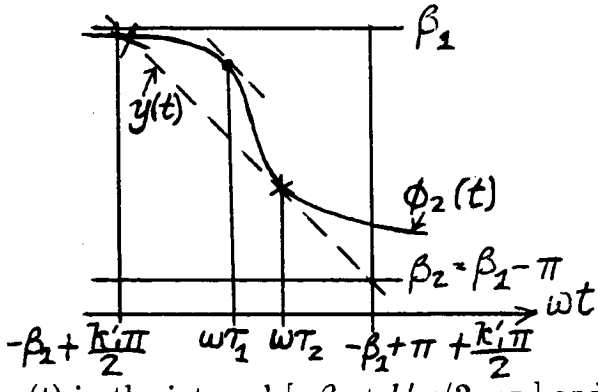
has two real roots, τ_1 and τ_2.



$\phi_2(t)$ intersects $y(t)$ at $\omega t = \omega\tau_1$ and in the interval $[\omega\tau_2, -\beta_1 + \pi + k_1'\pi/2]$.

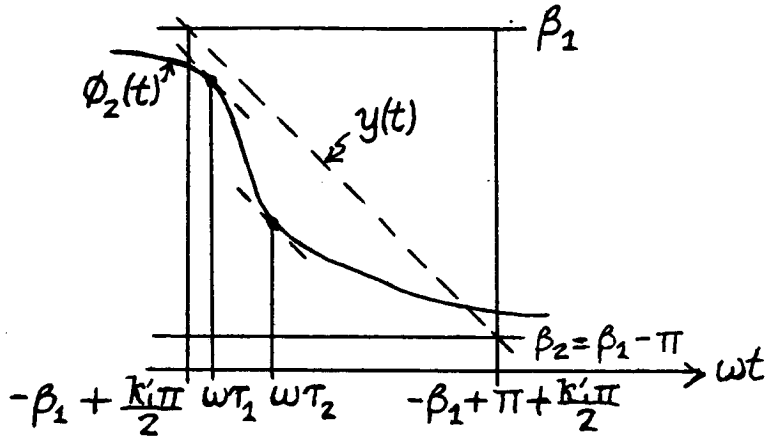


$\phi_2(t)$ intersects $y(t)$ in the three intervals $[-\beta_1 + k_1'\pi/2, \omega\tau_1]$, $[\omega\tau_1, \omega\tau_2]$, $[\omega\tau_2, -\beta_1 + \pi + k_1'\pi/2]$.

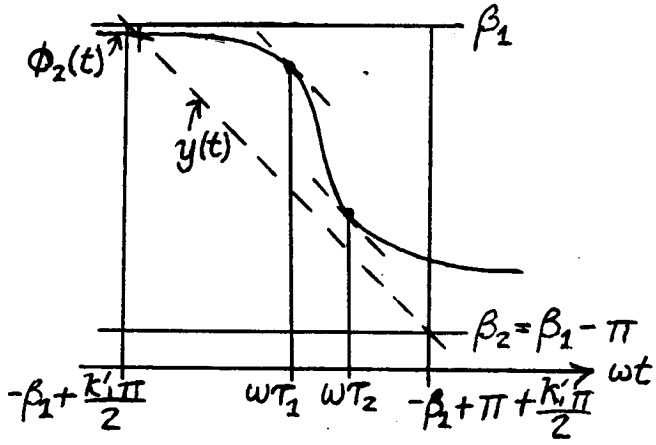


$\phi_2(t)$ intersects $y(t)$ in the interval $[-\beta_1 + k_1'\pi/2, \omega\tau_1]$ and at $\omega t = \omega\tau_2$.

Figure 5.6: The three possibilities for multiple intersections of $\phi_2(t)$ and the line $y(t)$ when $d_\phi < 0$



$$\begin{aligned} \phi_2(\tau_1) &< y(\tau_1) \\ \phi_2(\tau_2) &< y(\tau_2) \end{aligned}$$



$$\begin{aligned} \phi_2(\tau_1) &> y(\tau_1) \\ \phi_2(\tau_2) &> y(\tau_2) \end{aligned}$$

Figure 5.7: The two cases where multiple intersections do not occur when τ_1 and τ_2 lie in the same interval

2. There is some odd integer k_1' for which

$$-\beta_1 + k_1'\pi/2, < \omega\tau_1 < \omega\tau_2 < -\beta_1 + \pi + k_1'\pi/2$$

3. The values of $\phi_2(t)$ and $y(t)$ at τ_1 and τ_2 satisfy the inequalities

$$\begin{array}{rcrcl} \phi_2(\tau_1) & \geq & y(\tau_1) & = & -\omega\tau_1 + k_1'\pi/2 \\ \phi_2(\tau_2) & \leq & y(\tau_2) & = & -\omega\tau_2 + k_1'\pi/2 \end{array}$$

If any of these three conditions is not satisfied, then there will be one and only one intersection of the curve of $\phi_2(t)$ with the line $y(t) = -\omega t + k'\pi/2$ in each interval $[-\beta_1 + k'\pi/2, -\beta_1 + \pi + k'\pi/2]$. In other words, each interval will contain a single zero of d^2f/dt^2.

If the three conditions are all satisfied, then there will be either two or three intersections of the curve $\phi_2(t)$ in the interval $[-\beta_1 + k_1'\pi/2, -\beta_1 + \pi + k_1'\pi/2]$ that contains $\omega\tau_1$ and $\omega\tau_2$. The second derivative of $f(t)$ then has two or three zeros in this interval. When $\phi_2(\tau_1) = y(\tau_1)$, there are two zeros of d^2f/dt^2 in this interval. The first zero is located at $\omega t = \omega\tau_1$ and the second zero lies in the subinterval $[\omega\tau_2, -\beta_1 + \pi + k_1'\pi/2]$. When $\phi_2(\tau_1) > y(\tau_1)$ and $\phi_2(\tau_2) < y(\tau_2)$, there are three zeros of d^2f/dt^2 in this interval. One zero is located in each of the subintervals $[-\beta_1 + k_1'\pi/2, \omega\tau_1]$, $[\omega\tau_1, \omega\tau_2]$, and $[\omega\tau_2, -\beta_1 + \pi + k_1'\pi/2]$. Finally when $\phi_2(\tau_2) = y(\tau_2)$, there are again two zeros of d^2f/dt^2 in this interval. The first zero is located in the subinterval $[-\beta_1 + k_1'\pi/2, \omega\tau_1]$. The second zero is located at $\omega t = \omega\tau_2$. There is a single zero of d^2f/dt^2 in all the remaining intervals $[-\beta_1 + k'\pi/2, -\beta_1 + \pi + k'\pi/2]$ where $k' \neq k_1'$.

The preceding analysis has shown that in most cases there is a single zero of d^2f/dt^2 in each interval $[-\beta_1 - \pi + k'\pi/2, -\beta_1 + k'\pi/2]$ or $[-\beta_1 + k'\pi/2, -\beta_1 + \pi + k'\pi\ 2]$ of ωt. It is possible to divide each interval into smaller parts, one of which must contain the zero of d^2f/dt^2. The rules for finding the bounds on the part of an interval where the intersection point of the curve of $\phi_2(t)$ and the line $y(t)$ is located are presented in Appendix A. These rules are generated using the values of t and $\phi_2(t)$ where $\cos\phi_2(t) = 0$ and $\sin\phi_2(t) = 0$. The new bounds depend on the values of m_2, m_3, and d_ϕ as shown in Appendix A.

5.2.2.3 The Zero Search Procedure for Case 5

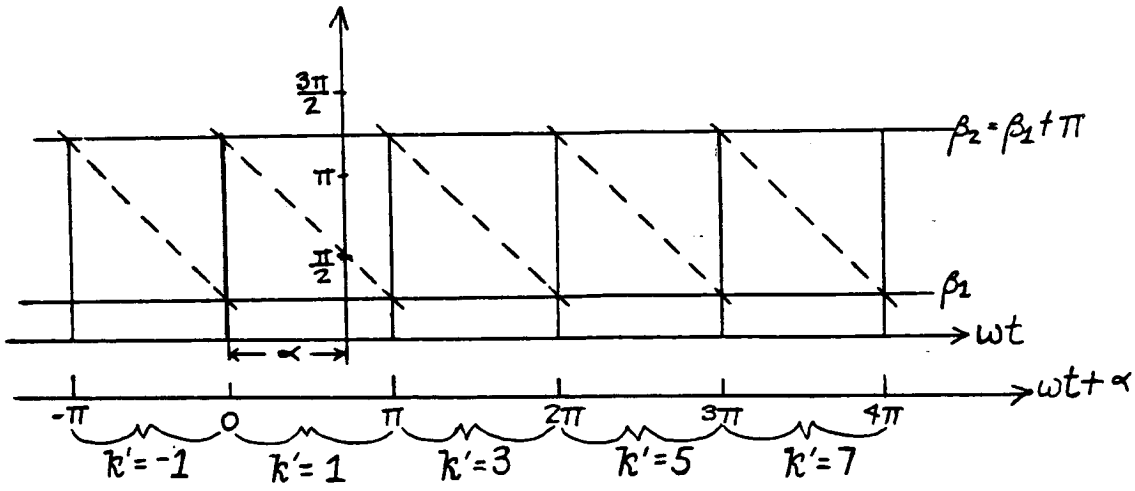
Recall the description of the zero search procedure for cases 6 and 7 of class *I* presented in section 5.1.5.2. This search procedure worked with a sequence of test points where a single zero of the function was guaranteed to lie between two consecutive points if the function value had opposite sign at those two points. The analysis in the previous section can be used to generate such a sequence of test points when the function to be searched for zeros is the second derivative of $f(t)$. This suggests a three-stage zero search procedure for finding the zeros of $f(t)$ for these cases.

The first stage searches for zeros of the second derivative of $f(t)$ within the bounds of the overall search interval. This stage divides the overall search interval into subintervals which contain no more than one zero of d^2f/dt^2. The secant method is used to iterate for the zero of d^2f/dt^2 in each subinterval. The result of the first stage is an ordered sequence of points consisting of the bounds of the search interval and those points within this interval where the second derivative of $f(t)$ is zero. This sequence is used as input for the second stage of the zero search procedure.

The second stage searches for the points where the first derivative of $f(t)$ is zero. By definition, the zeros of df/dt must lie at one of the points in the input sequence or between two consecutive points in the sequence. In the second stage, the values of df/dt are checked at each pair of consecutive points in the input sequence. If the value of df/dt changes sign at two consecutive points, the seacnt method is used to iterate for the zero of df/dt which lies between them. The result of the second stage is an ordered sequence of points consisting of the bounds of the overall search interval and the points in the interval where either the first or second derivative of $f(t)$ is zero. This is the same sequence of points that was generated analytically as input to the zero search procedure in cases 6 and 7 of class *I*.

The third stage of the zero search procedure for case 5 of class *II* finds the zeros of $f(t)$ in the same manner as the procedure for cases 6 and 7 of class *I*. The input test points are checked in pairs for changes in sign of $f(t)$. When a sign change occurs between two consecutive test points, the secant method is used to iterate for the zero of $f(t)$ between them.

The zero search for case 5 of class *II* can also be conducted on a periodic basis. The intervals used to find the points where the second derivative is



The intervals $[-\beta_1 - \pi + k'\pi/2, -\beta_1 + k'\pi/2]$ of ωt become the intervals $[n'\pi, (n'+1)\pi]$ of $\omega t + \alpha$.

$$- - - - \; y(t) = -\omega t + k'\pi/2 \text{ for } k' \text{ odd}$$

Figure 5.8: The new phase angle α

zero are of length π. Two of these intervals can be considered a single cycle of $\omega t + \alpha$. The new phase angle α plays the same role in this procedure that the phase angle ϕ_f plays in the zero search procedure for cases 6 and 7 of class *I*. The value of α is determined from the lower bounds of the intervals used in the analysis of d^2f/dt^2. The new phase angle α is defined to be the lower bound of one of these intervals for a particular odd integer k'_2.

As an example, consider the intervals shown in Figure 5.3. In this case, $\beta_1 \in (0, \pi/2)$ and $\beta_2 = \beta_1 + \pi \in (\pi, 3\pi/2)$. Choosing $k'_2 = 1$, the value of α becomes $\beta_1 + \pi/2$. Figure 5.8 shows the intervals defined by the lines $y(t)$ and the angles β_1 and β_2 for two different axes. The original axis is the ωt axis where the intervals are $[-\beta_1 - \pi + k'\pi/2, -\beta_1 + k'\pi/2]$. The second axis is the $\omega t + \alpha$ axis. The intervals become $[n'\pi, (n'+1)\pi]$ where n' is an integer. Clearly, two such intervals define an entire period of $\omega t + \alpha$ and $n' = 0$ when $\omega t = -\alpha$. The choice of k'_2, like the values of β_1 and β_2, depends on the values of m_2, m_3, and d_ϕ. This choice and the resulting values of α are listed in Appendix A for the sixteen possible combinations

of values of m_2, m_3, and d_ϕ.

There are several preliminary computations which must be made before the first stage of the case 5 zero search procedure can begin. First, the quantity d_ϕ is computed. When $d_\phi = 0$, the new phase angle α is taken to be ϕ_m as defined for the special case of d^2f/dt^2. When $d_\phi \neq 0$, the values of β_1, k_2', and α are computed using the values of m_2 and m_3 according to the sixteen possible cases listed in Appendix A. If $d_\phi < 0$, the three conditions for multiple zeros of d^2f/dt^2 to occur in one particular half-period of $\omega t + \alpha$ are checked sequentially as follows. The discriminant of the quadratic is computed. If the value of the discriminant is nonpositive, then the first condition is not satisfied. A flag is set which indicates that multiple zeros do not occur. If the value of the discriminant is positive, then the values of the two real roots of the quadratic, τ_1 and τ_2, are computed. Next, the second condition is checked by finding the two intervals where

$$-\beta_1 + k'\pi/2 < \omega\tau_1 < -\beta_1 + \pi + k'\pi/2$$

and

$$-\beta_1 + k'\pi/2 < \omega\tau_2 < -\beta_1 + \pi + k'\pi/2$$

If the value of k' is not the same for these two intervals, then the second condition is not satisfied. The flag which indicates that multiple intersections do not occur is set. If the value of k' is the same for these two intervals, then the second condition is satisfied and the third condition is checked. The values of $\phi_2(\tau_1)$ and $\phi_2(\tau_2)$ are computed and compared to the values of $y(\tau_1)$ and $y(\tau_2)$. If these values do not satisfy the inequalities for the third condition, the flag that indicates that multiple intersections do not occur is set. If these values do satisfy the inequalities for the third condition, then the flag indicating that multiple intersections *do* occur is set. In this case, an additional flag is set that distinguishes the three ways for multiple intersections to occur.

Once the value of α is known, the overall zero search interval is divided into subintervals at the values of t which define separate cycles of $\omega t + \alpha$. The bounds of each subinterval are usually the points where $\omega t + \alpha$ is equal to consecutive multiples of 2π. The lower bound of a subinterval is replaced by the lower bound of the search interval when the overall bound lies in the period defined by that subinterval. Similarly, the upper bound of a subinterval is replaced by the upper bound of the search interval when it

lies in the period defined by that subinterval. When one (or both) of its normal bounds is replaced by a bound of the overall search interval, the subinterval becomes only a partial cycle of $\omega t + \alpha$.

The three stages of the zero search procedure are performed for each subinterval within the overall search interval. The results of the preliminary checks and the bounds on the location of the zeros of d^2f/dt^2 are used only in the first stage. A brief discussion of the first stage procedure for the various cases of the behavior of d^2f/dt^2 is presented below. The second and third stages are not influenced by the behavior of d^2f/dt^2 and are executed as previously described for each subinterval once the first stage is completed.

In order to simplify the discussion, assume for the moment that the current subinterval defines an entire period of $\omega t + \alpha$, $[2n\pi, 2(n+1)\pi]$. The first stage of the zero search procedure further divides this subinterval at the point which separates the half-cycles of $\omega t + \alpha$. It considers those parts of the subinterval where $\omega t + \alpha \in [2n\pi, (2n+1)\pi]$ and where $\omega t + \alpha \in [(2n+1)\pi, 2(n+1)\pi]$. The first and final points in the sequence of points to be used in the second stage are set equal to the beginning and end of the current cycle, respectively. Zeros of d^2f/dt^2 in this cycle are added to the sequence in numerical order between these two points.

When $d_\phi = 0$, one zero of d^2f/dt^2 is located in the first half-cycle at $\omega t + \alpha = \pi/2 + 2n\pi$ and another is located in the second half-cycle at $\omega t + \alpha = 3\pi/2 + 2n\pi = \pi/2 + (2n+1)\pi$. The first stage need only add these two points to the sequence to be used in the second stage. There is another zero of d^2f/dt^2 at $\omega t + \alpha = \omega c_2 + \alpha$. If $\omega c_2 + \alpha$ falls within the current cycle, then it is also placed in the sequence according to its value relative to the other two zeros of d^2f/dt^2. In this special case, no iteration is necessary to find the zeros of the second derivative.

When $d_\phi > 0$ or when $d_\phi < 0$ and the preliminary checks indicate that no multiple intersections occur, there can be only one zero of d^2f/dt^2 in each half-cycle of $\omega t + \alpha$. The first stage executes the secant method twice to iterate for the zeros in each of the current two half-cycles. The starting values for the iteration are found from the guidelines given in Appendix A. The two zeros of d^2f/dt^2 resulting from application of the secant method are added to the input sequence for the second stage.

The first stage of the zero search procedure when $d_\phi < 0$ and when the preliminary checks indicate that multiple intersections occur is identical

to that described in the preceding paragraph except when one of the half-cycles in the current subinterval is the one where multiple intersections take place. The first stage compares the bounds of each half-cycle with the value of $\omega\tau_1$. If the value of $\omega\tau_1$ is within one of the half-cycles, then multiple intersections take place in this half-cycle. A flag set by the preliminary checks determines how many times the secant method is executed and the starting values it uses in this half-cycle. When this flag indicates that two zeros of d^2f/dt^2 are located in this half-cycle, then the secant method is executed only once because one of the zeros is located at either $\omega\tau_1$ or $\omega\tau_2$. When the flag indicates that the half-cycle contains three zeros of d^2f/dt^2, then the secant method is executed three times with three different sets of starting values. The special guidelines for the starting values in this half-cycle were listed in section 5.2.2.2. In this case, there will be three or four zeros of d^2f/dt^2 to be added to the input sequence for the second stage. One half-cycle will contain only a single zero of d^2f/dt^2 and the other will contain two or three zeros of d^2f/dt^2.

The first stage of the zero search must be modified somewhat when the current subinterval is not a full period of $\omega t + \alpha$. The entire first or second half-cycle may not be included in the subinterval or the subinterval may include only a part of the first or second half-cycles. However, even if part of a half-cycle is within the current subinterval, the zero(s) of d^2f/dt^2 do not necessarily lie in that part of the half-cycle. The first stage finds those parts, if any, of the two half-cycles that lie within the subinterval. If part of a half-cycle lies within the subinterval, then the search for zeros of d^2f/dt^2 is restricted to that part. This is done by comparing the interval defined by the starting guesses for the iteration given in Appendix A to the actual part of the half-cycle within the current subinterval. If the interval defined by the starting guesses is entirely contained within that part of the half-cycle, then the zero of d^2f/dt^2 must be in that part. The secant method is executed with the original starting values. If the interval defined by the starting guesses overlaps only partially with the part of the half-cycle in the current subinterval, the values of the second derivative at the ends of the interval of overlap are checked. If these values have opposite sign, then the zero of d^2f/dt^2 is located within the current subinterval. The secant method is executed to iterate for the zero of d^2f/dt^2 with the bounds of the interval of overlap as starting values. If the value of the second derivative does not change sign between the ends of the interval of overlap, then the

C-3

zero of d^2f/dt^2 is located in that part of the half-cycle which is outside of the current subinterval. Any zeros of d^2f/dt^2 that are found within a subinterval that is a partial cycle of $\omega t + \alpha$ are added to the input sequence for the second stage of the zero search.

5.2.2.4 Cases 3 & 4

For case 5, when $m_1 \neq 0$ and $l_1 \neq 0$, the form of the second derivative of $f(t)$ is $(a_1 t + b_1)\cos(\omega t) + (a_2 t + b_2)\sin(\omega t)$. The analysis of section 5.2.2.2 showed how to find intervals of ωt that contain a single zero of this form when $a_1 = -\omega^2 m_2$, $b_1 = 2\omega m_3 - \omega^2 l_2$, $a_2 = -\omega^2 m_3$, and $b_2 = -2\omega m_2 - \omega^2 l_3$. Now suppose that $m_1 = 0$ but $l_1 \neq 0$. Then the equation for df/dt becomes

$$\frac{df}{dt} = [\omega m_3 t + (m_2 + \omega l_3)]\cos(\omega t) + [-\omega m_2 t + (m_3 - \omega l_2)]\sin(\omega t)$$

This is of the form $(a_1 t + b_1)\cos(\omega t) + (a_2 t + b_2)\sin(\omega t)$ with $a_1 = \omega m_3$, $b_1 = m_2 + \omega l_3$, $a_2 = -\omega m_2$, and $b_2 = m_3 - \omega l_2$. A similar analysis can be performed to find the intervals that contain single zeros for df/dt for this case. Then, the zero search procedure can be reduced to two stages. The first stage searches for the zeros of df/dt in the overall search interval. The second stage uses the results of the first stage to search for the zeros for $f(t)$ itself in the overall search interval.

A further simplification is possible when both m_1 and l_1 are zero. Then the function is of the form $(a_1 t + b_1)\cos(\omega t) + (a_2 t + b_2)\sin(\omega t)$ with $a_1 = m_2$, $b_1 = l_2$, $a_2 = m_3$, and $b_2 = l_3$ and the intervals that contain a single zero of $f(t)$ can be identified analytically. The zero search procedure consists of a single stage which iterates for the zeros of $f(t)$ in the overall search interval.

Because the case where $m_1 = l_1 = 0$ potentially has the simplest zero search procedure, it is referred to here as case 3. The case when $m_1 = 0$ but $l_1 \neq 0$ is called case 4. The analysis required for the simplification of the zero search procedure in these two cases is not included in this thesis. Instead, the case 5 zero search procedure is used to find the zeros of $f(t)$ for cases 3 and 4. This simplified the programming required to implement the zero search for these cases.

5.2.3 Summary of Class *II*

The five class *II* cases for $f(t)$ are summarized in Table 5.1. This table gives the conditions that hold for each case as well as the special forms of $f(t)$ in this class. There are analytic formulas for finding the zero(s) of $f(t)$ for cases 1 and 2. There is only one zero of $f(t)$ at $t = -c$ for case 1. For case 2, there are periodic zeros of $f(t)$ whenever its shifted sinusoidal component equals zero, in addition to a zero at $t = -c$.

The zeros of $f(t)$ are found by a numerical search procedure in cases 3, 4, and 5. The search procedure creates a sequence of test points consisting of the bounds of the search interval and any values of t within the interval where the first or second derivative of $f(t)$ equals zero. Unlike the procedure for cases 6 and 7 of class *I*, the points where the first and second derivatives are zero cannot be found analytically. However, it is possible to find those intervals in the overall search interval that contain a single point where the second derivative is zero. Therefore, the zero search procedure is divided into three stages. The first stage finds the points in the search interval where the second derivative is zero. The results of this stage are used in the second stage to find the points in the search interval where the first derivative is zero. The third and final stage uses the results of the first two stages to find the zeros of $f(t)$ itself in the overall search interval. This third stage is identical to the zero search procedure for cases 6 and 7 of class *I*. In all three stages, the secant method is used to iterate for the zeros of the function or its first two derivatives.

The 5 Class II Cases	
Case	Distinguishing Conditions
1	$\exists\, c \in \Re$ such that $l_1 = cm_1$, $l_2 = cm_2$, and $l_3 = cm_3$ $\left\lvert m_1/\sqrt{m_2^2 + m_3^2}\right\rvert > 1$
2	$\exists\, c \in \Re$ such that $l_1 = cm_1$, $l_2 = cm_2$, and $l_3 = cm_3$ $\left\lvert m_1/\sqrt{m_2^2 + m_3^2}\right\rvert \leq 1$
3	$\nexists\, c \in \Re$ such that $l_1 = cm_1$, $l_2 = cm_2$, and $l_3 = cm_3$ $m_1 = l_1 = 0$ ($f(t)$ is of the form $(a_1 t + b_1)\cos(\omega t) + (a_2 t + b_2)\sin(\omega t)$)
4	$\nexists\, c \in \Re$ such that $l_1 = cm_1$, $l_2 = cm_2$, and $l_3 = cm_3$ $m_1 = 0$, $l_1 \neq 0$. (df/dt is of the form $(a_1 t + b_1)\cos(\omega t) + (a_2 t + b_2)\sin(\omega t)$)
5	$\nexists\, c \in \Re$ such that $l_1 = cm_1$, $l_2 = cm_2$, and $l_3 = cm_3$ $m_1 \neq 0$ (d^2f/dt^2 is of the form $(a_1 t + b_1)\cos(\omega t) + (a_2 t + b_2)\sin(\omega t)$)

Class II Forms of $f(t)$	
Case(s)	Forms of $f(t)$
1 & 2	$f(t) = (t + c)(m_1 + m_2\cos(\omega t) + m_3\sin(\omega t))$
3	$f(t) = (m_2 t + l_2)\cos(\omega t) + (m_3 t + l_3)\sin(\omega t)$
4	$f(t) = l_1 + (m_2 t + l_2)\cos(\omega t) + (m_3 t + l_3)\sin(\omega t)$
5	$f(t) = (m_1 t + l_1) + (m_2 t + l_2)\cos(\omega t) + (m_3 t + l_3)\sin(\omega t)$

Table 5.2: The five cases for class II of the generalized C-function

Chapter 6

The Collision Detection Algorithm

The conceptual collision detection procedure introduced in Chapter 2 is basically a search for the zeros of the C-functions. Techniques for bounding the intervals of time that must be searched for these zeros were developed in Chapters 3 and 4. Then, methods for locating the zeros of the C-functions in these bounded search intervals were developed in Chapter 5. This chapter describes the collision detection algorithm that has been implemented for this thesis by combining the ideas introduced in these previous chapters.

This chapter first specifies the inputs required by the algorithm. These inputs include the data modelling the shapes of $\mathcal{A}$ and $\mathcal{B}$ and the quantities used in the equations for the translational and rotational motion of $\mathcal{A}$. Next, the algorithm itself is discussed. Four major procedures are identified and the hierarchy of the overall algorithm using these procedures is described. Then, an example is given to illustrate a typical path through the hierarchy of the algorithm. Finally, two special types of collisions are discussed where numerical errors in the computer's calculations can cause problems for the algorithm.

6.1 The Inputs

6.1.1 The Model for the Shapes of $\mathcal{A}$ and $\mathcal{B}$

The first requirement of the collision detection algorithm is a model of the shapes of the moving object $\mathcal{A}$ and the obstacle $\mathcal{B}$. For this thesis, $\mathcal{A}$ and $\mathcal{B}$ are assumed to be polyhedra, either convex or nonconvex. A given object can be represented to any desired degree of accuracy by adding more faces to the polyhedron. Of course, the storage space and computation time required by the collision detection algorithm increase with the complexity of the polyhedron.

Several schemes for the representation of polyhedra have been developed for computer graphics systems. Typical datastructures are described in references [24,25] and Chapter 10 of reference [11]. Rather than implement

one of these datastructures, a simplified datastructure is used that contains the minimum amount of information necessary for the collision detection algorithm. No attempt is made to make this representation efficient. It is assumed that the information entered in the datastructure describing $\mathcal{A}$ or $\mathcal{B}$ represents a physically realizable solid. The algorithm does not check that these data are consistent with the actual shapes of the two polyhedra.

6.1.1.1 The Datastructure for a Convex Polyhedron

The datastructure used by the collision detection algorithm is a graph-based model. The features of the polyhedron are the "nodes" of the graph. The "branches" between these nodes describe the connections between the features. The numerical information pertaining to each "node" is stored separately from the connectivity information. See pages 431-433 of reference [11] for a more detailed discussion of graph-based models for polyhedra.

The numerical information for a convex polyhedron specifies the number, size, and location of its features. This includes the number of vertices, edges, and faces of the polyhedron and the vectors representing each vertex, edge, and face as defined in section 2.2.1. The algorithm also uses the distance of each face from the origin of the appropriate coordinate system as measured along its outward-pointing normal vector. These distances are the quantities d_i^a and d_j^b defined in sections 4.1.1 and 4.1.2. They can be precomputed using the vertex vectors and the face normal vectors.

The connectivity information for a polyhedron is concerned with the topology of the polyhedron and not its size or location in space. The only information of this kind needed by the collision detection algorithm gives the connections between the vertices and edges and between the edges and faces of the polyhedron. The algorithm requires the numbers of the edges meeting at each vertex of the polyhedron, the numbers of the two faces meeting at each edge of the polyhedron, and the number of the tail vertex for each edge vector.

The connections between the edges and vertices and the edges and faces of the polyhedron are stored in two *connectivity* matrices. The rows and columns of a *connectivity* matrix represent the features of the polyhedron. Its elements are binary. If the element in the ith row and jth column of the matrix is 1, then the ith and jth feature are connected. If this element is equal to 0, these two features are not connected. The vertex-

edge connectivity matrix has row dimension equal to the number of vertices of the polyhedron and column dimension equal to the number of edges of the polyhedron. The column numbers of the nonzero elements in row i of this matrix are the numbers of the edges of the polyhedron that meet at its ith vertex. There must be at least three nonzero elements in each row of this matrix since at least three edges must meet at each vertex of a polyhedron. The edge-face connectivity matrix has row dimension equal to the number of edges of the polyhedron and column dimension equal to the number of faces of the polyhedron. The column numbers of the two nonzero elements in row i of this matrix are the numbers of the two faces of the polyhedron that meet at its ith edge.

In summary, the numerical and connectivity data required by the collision detection algorithm for a *convex* polyhedron $\mathcal{P}$ are:

1. The integer $v_{\mathcal{P}}$ representing the number of vertices of $\mathcal{P}$
2. The integer $e_{\mathcal{P}}$ representing the number of edges of $\mathcal{P}$
3. The integer $f_{\mathcal{P}}$ representing the number of faces of $\mathcal{P}$
4. The $v_{\mathcal{P}}$ position vectors for the vertices of $\mathcal{P}$
5. The $e_{\mathcal{P}}$ edge vectors of $\mathcal{P}$
6. The $f_{\mathcal{P}}$ unit, outward-pointing normal vectors for the faces of $\mathcal{P}$
7. The $f_{\mathcal{P}}$ scalars specifying the distances of the faces of $\mathcal{P}$ from the origin of the appropriate coordinate system measured along the corresponding face normal vectors
8. The $e_{\mathcal{P}}$ integers specifying the numbers of the tail vertices of each edge vector of $\mathcal{P}$
9. The $v_{\mathcal{P}} \times e_{\mathcal{P}}$ connectivity matrix for the vertices and edges of $\mathcal{P}$
10. The $e_{\mathcal{P}} \times f_{\mathcal{P}}$ connectivity matrix for the edges and faces of $\mathcal{P}$

The collision detection algorithm makes use of the numerical information for the features of the polyhedron to compute the constant coefficients of the C-functions and their ACFs as functions of time as well as to check the physical constraints for each C-function. The information describing

the topology of the polyhedron is used to determine the applicability intervals of the C-functions. The vertex-edge connectivity matrix is used to find the ACFs for the type (a) and (b) C-functions. Both the vertex-edge and the edge-face connectivity matrices are needed to find the ACFs for the type (c) C-functions. The number of the tail vertex of each edge vector is used to determine the correct orientation of an edge vector when used in an ACF. By definition, the ACF requires the edge vector to have a certain tail vertex. If the vector stored in the datastructure for that edge does not have that vertex as its tail vertex, its direction must be reversed before it is used to compute the coefficients of the ACF.

6.1.1.2 The Datastructures for $\mathcal{A}$ and $\mathcal{B}$

Because convexity is a key assumption for the applicability tests, the collision detection algorithm regards each object as the union of convex polyhedra. For this reason, the datastructures for $\mathcal{A}$ and $\mathcal{B}$ are divided into substructures that describe their convex polyhedral components.

Recalling the notation of section 2.8, let $\mathcal{S}_a$ be the set of n_a *convex* polyhedra $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_{n_a}$ whose union is $\mathcal{A}$ and let $\mathcal{S}_b$ be the set of n_b *convex* polyhedra $\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_{n_b}$ whose union is $\mathcal{B}$. The integers n_a and n_b are the first entries in the datastructures describing $\mathcal{A}$ and $\mathcal{B}$, respectively. The radius r_a of the sphere enclosing $\mathcal{A}$ is part of the datastructure for $\mathcal{A}$. The radius r_b of the sphere enclosing $\mathcal{B}$ and the position vector $\mathbf{x}^b$ of its center C^b are part of the datastructure for $\mathcal{B}$. These quantities can be precomputed from the vertex vectors of $\mathcal{A}$ and $\mathcal{B}$, as explained in section 4.3. The remaining data for $\mathcal{A}$ are divided into n_a substructures describing the individual convex polyhedra $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_{n_a}$. Similarly, the remaining data for $\mathcal{B}$ are divided into n_b substructures describing each of the convex polyhedra $\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_{n_b}$. Note that when $\mathcal{A}$ or $\mathcal{B}$ is a single convex polyhedron, $n_a = 1$ or $n_b = 1$ and there is only one substructure describing the single convex polyhedron $\mathcal{A}_1 \equiv \mathcal{A}$ or $\mathcal{B}_1 \equiv \mathcal{B}$.

The substructure for each convex polyhedron $\mathcal{P} \in \mathcal{S}_a$ or $\mathcal{S}_b$ contains the ten quantities listed in the preceding section. The vectors in this substructure for $\mathcal{P} \in \mathcal{S}_a$ are expressed in the body coordinates of $\mathcal{A}$ with origin at C^a, the center of rotation of $\mathcal{A}$. When $\mathcal{P} \in \mathcal{S}_b$, the vectors are expressed in reference coordinates.

In section 2.2.1, the vectors representing the ith vertex, edge, and face

of $\mathcal{A}$ or $\mathcal{B}$ were denoted $\mathbf{a}_i$, $\mathbf{e}_i^a$, $\mathbf{u}_i^a$ or $\mathbf{b}_i$, $\mathbf{e}_i^b$, $\mathbf{u}_i^b$, respectively. Since $\mathcal{A}$ and $\mathcal{B}$ were assumed to be single convex polyhedra for that section, only one subscript was needed to identify each vector. When $\mathcal{A}$ or $\mathcal{B}$ consist of more than one convex polyhedron, the single subscript is not sufficient to identify a particular feature. The number of the convex polyhedron on which the feature is located and the number of the feature are both required. In order to avoid more cumbersome notation, a second subscript will not be introduced. Instead, the relevant convex polyhedron will be specified in the text. For example, the vector representing the fifth vertex of the second convex polyhedron $\mathcal{A}_2 \in \mathcal{S}_a$ will be referred to as "$\mathbf{a}_5$ of $\mathcal{A}_2$."

6.1.2 The Model for the Motion of $\mathcal{A}$

The second requirement for the collision detection algorithm is a model for the motion of $\mathcal{A}$ relative to $\mathcal{B}$. For this thesis, $\mathcal{A}$ is assumed to be moving with constant linear and angular velocity relative to $\mathcal{B}$. The equations describing the trajectory of $\mathcal{A}$ are then:

$$\begin{aligned} \mathbf{x}(t) &= \mathbf{x}_o + \mathbf{v}t \\ \mathbf{Q}(t) &= \mathbf{Q}_\omega(t)\,\mathbf{Q}_o = [\ \cos\left(\tfrac{\omega t}{2}\right)\ \ \sin\left(\tfrac{\omega t}{2}\right)\ \mathbf{u}_\omega^T\ ]^T\,\mathbf{Q}_o \end{aligned}$$

where $\mathbf{x}(t)$ is the position of C^a in reference coordinates at time t and $\mathbf{Q}(t)$ is the quaternion representing the orientation of the body coordinates of $\mathcal{A}$ relative to the reference coordinate axes at time t. The quantities ω and $\mathbf{u}_\omega$ are computed from the constant angular velocity vector $\vec{\omega}$.

The three vectors $\mathbf{x}_o$, $\mathbf{v}$, and $\vec{\omega}$ and the quaternion $\mathbf{Q}_o$ must be specified as input to the collision detection algorithm. If this algorithm were being executed onboard a spacecraft, the quantities $\mathbf{x}_o$ and $\mathbf{Q}_o$ would be available from navigation software that uses sensor measurements to determine the position and orientation of the spacecraft relative to some other vehicle. The navigation routine would also provide estimates of $\mathbf{v}$ and $\vec{\omega}$, the linear and angluar velocities relative to the other vehicle. These two vectors are used to predict the trajectory of the moving spacecraft. The implementation of the algorithm for this thesis requires that the quantities $\mathbf{x}_o$, $\mathbf{v}$, $\vec{\omega}$, and $\mathbf{Q}_o$ be entered by the user prior to the start of the algorithm.

The initial configuration of $\mathcal{A}$ is specified by $\mathbf{x}_o$ and $\mathbf{Q}_o$. The initial position vector $\mathbf{x}_o$ must be expressed in reference coordinates. The initial orientation of the body coordinate axes of $\mathcal{A}$ with respect to the reference

coordinate axes can be entered in any convenient format and converted to the quaternion $\mathbf{Q}_o$. For this thesis, the initial orientation of $\mathcal{A}$ was specified using yaw, pitch, and roll Euler angles. This corresponds to sequential rotations about the Z_a, Y_a, and X_a body axes, respectively, as explained in section 3.1.1.

All of the information about the future motion of $\mathcal{A}$ is contained in the two vectors $\mathbf{v}$ and $\vec{\omega}$. The linear velocity vector $\mathbf{v}$ must be expressed in reference coordinates. The angular velocity vector $\vec{\omega}$ may be expressed in either reference coordinates or in the body coordinates of $\mathcal{A}$. When it is expressed in the body coordinates of $\mathcal{A}$, the quaternion $\mathbf{Q}_o$ is used to convert it to reference coordinates prior to the start of the collision detection algorithm. This vector is unchanged by subsequent rotations of $\mathcal{A}$ away from its initial orientation because constant angular velocity is assumed.

6.2 The Algorithm

Four major procedures can be identified as parts of the overall collision detection algorithm:

1. Static spherical approximation
2. Dynamic spherical approximation
3. Configuration space static collision detection
4. Configuration space dynamic collision detection

Each of these procedures has different inputs and is designed to answer different questions related to the collision detection problem. These procedures are discussed in detail in the next three sections. Section 6.2.4 describes the hierarchy of the overall algorithm that results from uniting these four procedures.

6.2.1 The Static and Dynamic Spherical Approximation Procedures

The static and dynamic spherical approximation procedures are closely related. As the names imply, both of them make use of the approximation

that $\mathcal{A}$ and $\mathcal{B}$ are spheres. The static procedure determines if the spheres enclosing $\mathcal{A}$ and $\mathcal{B}$ are overlapping, just touching, or disjoint at the initial position of $\mathcal{A}$ when $t = 0$. The dynamic procedure determines the interval of time, if any, when the spheres enclosing $\mathcal{A}$ and $\mathcal{B}$ overlap.

These two procedures determine the relationship of the spheres enclosing $\mathcal{A}$ and $\mathcal{B}$ by comparing the distance between the centers of the two spheres, $\|\mathbf{x} - \mathbf{x}^b\|$, to their combined radii, $R = r_a + r_b$. When $\|\mathbf{x} - \mathbf{x}^b\| > R$, the two spheres are disjoint. When $\|\mathbf{x} - \mathbf{x}^b\| \leq R$, the two spheres are either just touching or overlapping.

The possible outcomes of the static and dynamic spherical approximation procedures are described below.

The Static Spherical Approximation Procedure This procedure is the simplest of the four major procedures. It compares the magnitude of the vector $\mathbf{x}_o - \mathbf{x}^b$ to R. When $\|\mathbf{x}_o - \mathbf{x}^b\| > R$, the procedure indicates that the two spheres are initially disjoint. When $\|\mathbf{x}_o - \mathbf{x}^b\| \leq R$, the procedure indicates that the two spheres are just touching or overlapping at $t = 0$.

The Dynamic Spherical Approximation Procedure The dynamic spherical approximation was discussed in section 4.3. A test was derived that determines the interval of time, if any, when the spheres enclosing $\mathcal{A}$ and $\mathcal{B}$ overlap along the trajectory of $\mathcal{A}$. There are three possible results of this test. The first possible result is that the spheres enclosing $\mathcal{A}$ and $\mathcal{B}$ do not come into contact at any point along the trajectory of $\mathcal{A}$. In this case, no collision can take place between $\mathcal{A}$ and $\mathcal{B}$. The second possible result is that the spheres enclosing $\mathcal{A}$ and $\mathcal{B}$ are just touching at one point t_v along the trajectory of $\mathcal{A}$. Therefore, t_v is the only time when a collision may take place between $\mathcal{A}$ and $\mathcal{B}$. The third possible result is that the spheres enclosing $\mathcal{A}$ and $\mathcal{B}$ overlap for the finite interval $[T_1, T_2]$. Any collisions between $\mathcal{A}$ and $\mathcal{B}$ must take place for values of t in this interval.

The dynamic spherical approximation procedure makes some additional checks in the case of the second or third results for this test. These new checks are required because the goal of the collision detection algorithm is to predict the time of the first collision, if any, that will take place between $\mathcal{A}$ and $\mathcal{B}$ *after* the intial time $t = 0$.

When the spheres are just touching at a single point t_v, the procedure must check that $t_v \geq 0$. If $t_v < 0$, the procedure will indicate that the

spheres do not overlap for $t \geq 0$.

When the spheres overlap for a finite interval $[T_1, T_2]$, the procedure must determine if any part of this interval lies on the positive half of the t axis. If $T_1 < T_2 < 0$, then the procedure indicates that the spheres enclosing $\mathcal{A}$ and $\mathcal{B}$ do not overlap when $t > 0$. If $T_1 < 0$ but $T_2 > 0$, then the procedure sets $T_1 = 0$. The values of T_1 and T_2 are retained without change if $T_2 > T_1 \geq 0$.

6.2.2 The Configuration Space Static Collision Detection Procedure

The configuration space static collision detection procedure uses the theory presented in Chapter 2 to determine if $\mathcal{A}$ and $\mathcal{B}$ are just touching or overlapping or are disjoint at a fixed time τ. This procedure is basically a check of the third collision condition given in section 2.3.3 at time τ. Recall that this third condition distinguishes those configurations of $\mathcal{A}$ that lie on *or* inside of $\mathbf{CO}_{\mathcal{B}}^{\mathcal{A}}$ from those configurations of $\mathcal{A}$ in free space.

This procedure makes a separate check of the third collision condition for each pair of convex polyhedra $\mathcal{A}_i \in \mathcal{S}_a$ and $\mathcal{B}_j \in \mathcal{S}_b$. First, a "collision" flag is initialized to indicate that $\mathcal{A}_i$ and $\mathcal{B}_j$ *are* in contact at time τ. It is only necessary to find one applicable C-function whose value violates the third collision condition to prove that this assumption of contact is false and that $\mathcal{A}_i$ and $\mathcal{B}_j$ are *not* in contact. Hence, the procedure next examines each C-function defined by the features of $\mathcal{A}_i$ and $\mathcal{B}_j$. The values of the ACFs for each C-function at time τ are computed. These values are used to determine whether or not the function is applicable at time τ. They also determine the orientation of the C-function when it is applicable. If a C-function is found to be applicable at time τ, then its value at time τ is computed. If *any* applicable C-function has positive orientation and positive value or has negative orientation and negative value, then the configuration of $\mathcal{A}$ is outside of $\mathbf{CO}_{\mathcal{B}_j}^{\mathcal{A}_i}$. The "collision" flag is then set to indicate that $\mathcal{A}_i$ and $\mathcal{B}_j$ are not in contact at time τ. Otherwise, the "collision" flag retains its initial setting indicating that $\mathcal{A}_i$ and $\mathcal{B}_j$ are in contact at time τ.

The vector $\mathbf{x}(\tau)$ and the quaternion $\mathbf{Q}(\tau)$ representing the configuration of $\mathcal{A}$ at time τ are assumed to be available at the start of this procedure. The values of the ACFs and the C-functions are computed directly. The vectors representing features of $\mathcal{A}_i$ that appear in the ACFs and the C-

functions are transformed to reference coordinates from the body coordinates of $\mathcal{A}$ using the quaternion $\mathbf{Q}(\tau)$.

The configuration space static collision detection procedure finally checks the "collision" flag for each combination of values for i and j. If the "collision" flag indicates that $\mathcal{A}_i$ and $\mathcal{B}_j$ are in contact at time τ for at least one pair of values $i = 1, \ldots, n_a$ and $j = 1, \ldots, n_b$, then the procedure indicates that $\mathcal{A}$ and $\mathcal{B}$ are in contact at time τ. The procedure indicates that $\mathcal{A}$ and $\mathcal{B}$ are *not* in contact at time τ only if the "collision" flag shows that $\mathcal{A}_i$ and $\mathcal{B}_j$ are not in contact for *all* values of $i = 1, \ldots, n_a$ and $j = 1, \ldots, n_b$.

6.2.3 The Configuration Space Dynamic Collision Detection Procedure

The configuration space dynamic collision detection procedure determines the time T_c, if any, where $\mathcal{A}$ and $\mathcal{B}$ first come into contact during a given time interval where $t > 0$. As was the case for the static procedure, the configuration space dynamic collision detection procedure considers each pair of convex polyhedra $\mathcal{A}_i \in \mathcal{S}_a$ and $\mathcal{B}_j \in \mathcal{S}_b$ separately.

This procedure searches for the first allowable zero of any C-function that satisfies the third collision condition. First, the applicability and orientation timelines for each of the C-functions defined by $\mathcal{A}_i$ and $\mathcal{B}_j$ are computed and stored as discussed in Chapter 3. These timelines are used to restrict the zero search for the C-functions and to simplify the check of the third collision condition at an allowable zero.

Once the applicability and orientation timelines have been found, the procedure examines each of the C-functions individually. This is the most complex part of the configuration space dynamic collision detection procedure. The examination of a single C-function uses the information from checks of applicability, from the physical constraint, and from the analytic root interval(s) to determine if a search should be made to locate the allowable zeros of that C-function. If a zero search is performed and an allowable zero for the C-function is found, the third collision condition is checked to determine if that zero is a valid collision of $\mathcal{A}_i$ and $\mathcal{B}_j$ and, hence, of $\mathcal{A}$ and $\mathcal{B}$. The interaction of the checks used to examine a C-function will be described in detail in the next section.

Let $[\mathcal{T}_1, \mathcal{T}_2]$ be the interval of time specified as input to this procedure where $\mathcal{T}_2 > \mathcal{T}_1 \geq 0$. When any valid collision is found between $\mathcal{A}_i$ and $\mathcal{B}_j$

at time t' in this interval, $\mathcal{A}$ and $\mathcal{B}$ must be in contact at time t'. The goal of the procedure is to find the time of the *first* contact between $\mathcal{A}$ and $\mathcal{B}$. Therefore, any valid collisions that take place after t' can be ignored. The search for zeros of the C-function that is zero at t' can be terminated and the search for zeros of the remaining C-functions can be confined to the interval $[\mathcal{T}_1, t']$. This includes the remaining C-functions defined by the current pair of convex polyhedra $\mathcal{A}_i$ and $\mathcal{B}_j$ and all of the C-functions defined by any remaining pairs of convex polyhedra. The procedure updates the zero search interval by setting $\mathcal{T}_2$ equal to t'.

The order in which the C-functions for each pair of convex polyhedra are examined is first type (b), then type (a), and finally type (c). This order is chosen because the type (b) C-functions are all in class I and therefore have a simpler zero search procedure than the type (a) and (c) C-functions. If a valid collision is found for any of the type (b) C-functions, the search interval for the type (a) and (c) C-functions will be further restricted by this collision as explained in the paragraph above. This means that the more complicated zero search procedure for the type (a) and (c) C-functions in calss II will be executed over a smaller interval than would be the case if the type (a) or (c) C-functions were examined before the type (b) C-functions.

6.2.3.1 The Examination of a Single C-function

The examination of a particular C-function begins with the current search interval $[\mathcal{T}_1, \mathcal{T}_2]$. The upper bound of this interval, $\mathcal{T}_2$, may have been modified from its initial value as input to the configuration space dynamic collision detection procedure by previous examinations of other C-functions. The first contact, if any, between $\mathcal{A}$ and $\mathcal{B}$ must take place in this interval.

This examination can be viewed as a two-stage process. In the first stage, various checks are made to determine if there are any subintervals of $[\mathcal{T}_1, \mathcal{T}_2]$ that may contain zeros of the C-function that can represent valid collisions. The purpose of these checks is to limit the zero search for the C-function as much as possible because of the complexity of the numerical zero search procedures. The checks are much simpler to implement than these numerical procedures. As will be shown below, it is possible that no zero search will be required for the C-function as a result of these checks. If the checks in the first stage of the examination of the C-function do not eliminate it from a zero search, then the examination moves into the second

stage which is the execution of the appropriate zero search procedure.

The hierarchy of the checks performed in examining a C-function is described below. This hierarchy is summarized in the flowchart shown in Figure 6.1. The points labeled "EXIT" in the flowchart are points where the examination of the current C-function is terminated and the procedure begins the examination of the next C-function.

The first step in the examination of a C-function is a check of the applicability of that C-function. If the C-function is not applicable for any portion of a cycle of ωt, then no zeros of this C-function can be valid collisions between $\mathcal{A}$ and $\mathcal{B}$. No further checks are then necessary for this C-function and the configuration space dynamic collision detection procedure moves on to examine the next C-function.

If the C-function is applicable for one or more intervals in a cycle of ωt, then the procedure checks the physical constraint for the C-function. If the physical constraint is not satisfied for the C-function, then no zeros of this C-function can be valid collisions between $\mathcal{A}$ and $\mathcal{B}$ and the procedure moves on to the next C-function.

Now consider the special case where the physical constraint is satisfied at a single value of t, t_p. In this case, t_p is equal to t_a or t_c as defined in sections 4.4.1 and 4.4.3, respectively. First, the procedure checks to see if t_p lies in the current zero search interval. If t_p is not in the current interval, then no zero of this C-function in the interval can be a valid collision between $\mathcal{A}$ and $\mathcal{B}$. The procedure moves on to the next C-function. If $\mathcal{T}_1 \leq t_p \leq \mathcal{T}_2$, then this is the only value of t where the C-function *may* have a zero that is a valid collision between $\mathcal{A}$ and $\mathcal{B}$. The value of ωt_p modulo 2π is checked to see if it lies in an applicability interval of the C-function. If t_p is not in an interval where the C-function is applicable, then there are no zeros of the C-function representing valid collisions between $\mathcal{A}$ and $\mathcal{B}$. The procedure then begins the examination of the next C-function. If the C-function is applicable at t_p then its value is computed at t_p. If the C-function equals zero at t_p, then the third collision condition is checked. If this condition is not satisfied, then t_p is not a valid collision between $\mathcal{A}$ and $\mathcal{B}$ and the procedure moves on to the next C-function. If the third collision condition is satisfied, then t_p is a valid collision. The current time of first contact, T_c, is set equal to t_p and the search interval is updated by setting $\mathcal{T}_2 = t_p$. The procedure then moves on to the next C-function.

If the physical constraint is satisfied for some finite interval of time

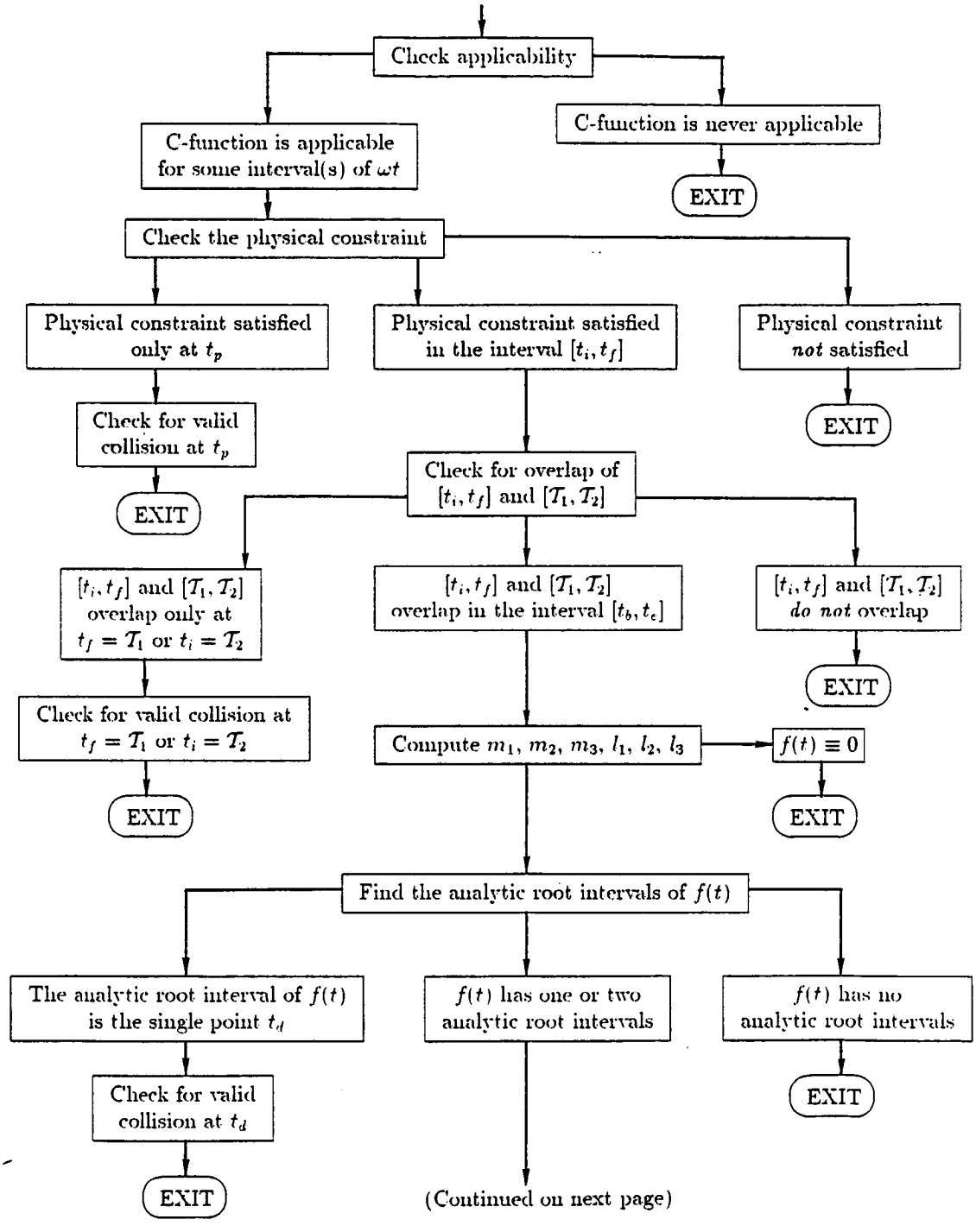


Figure 6.1: The flowchart for the examination of a single C-function

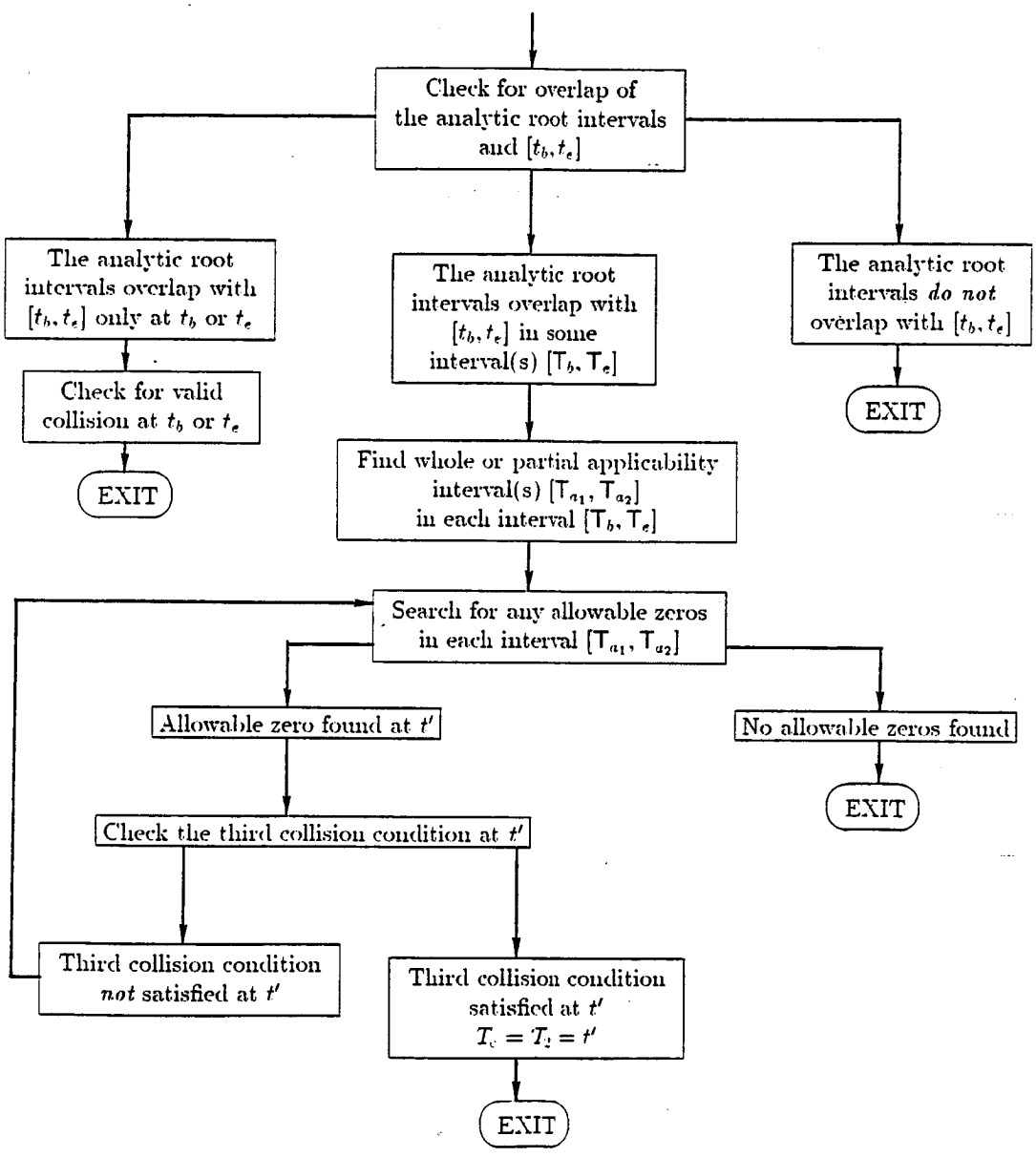


Figure 6.1: (con't) The flowchart for the examination of a single C-function

$[t_i, t_f]$, then this interval is compared with the current zero search interval $[\mathcal{T}_1, \mathcal{T}_2]$. If these two intervals do not overlap, then no zeros of the C-function in the current search interval can be valid collisions of $\mathcal{A}$ and $\mathcal{B}$ and the procedure moves on to the next C-function.

If these two intervals have only one endpoint in common, that is if $t_f = \mathcal{T}_1$ or $t_i = \mathcal{T}_2$, then this common value of t is the only possible value where a zero of the C-function can be a valid collision of $\mathcal{A}$ and $\mathcal{B}$. The value of ωt_i or ωt_f modulo 2π is checked to see if it lies in an applicability interval of the C-function. If t_i or t_f is not in an interval where the C-function is applicable, then there are no zeros of the C-function representing valid collisions between $\mathcal{A}$ and $\mathcal{B}$. The procedure begins the examination of the next C-function. If the C-function is applicable at t_i or t_f, then the value of the function itself is computed at t_i or t_f. If the value of the C-function is not zero at t_i or t_f, then the procedure moves on to the next C-function. If its value is zero, then the third collision condition is checked. If this condition is not satisfied, then t_i or t_f is not a valid collision between $\mathcal{A}$ and $\mathcal{B}$ and the procedure moves on to the next C-function. If the third collision condition is satisfied, then t_i or t_f is a valid collision. The current time of first contact, T_c, is set equal to t_i or t_f and the search interval is updated by setting $\mathcal{T}_2 = t_i$ or t_f. The procedure then moves on to the next C-function.

If the intervals $[\mathcal{T}_1, \mathcal{T}_2]$ and $[t_i, t_f]$ overlap at more than one point, then the examination of the C-function can be restricted to the interval $[t_b, t_e]$ that includes all the points where these two intervals overlap. For the special case where the physical constraint is always satisfied, the procedure sets $t_b = \mathcal{T}_1$ and $t_e = \mathcal{T}_2$. In both of these cases, the physical constraint is satisfied for $t_b \leq t \leq t_e$ where $\mathcal{T}_1 \leq t_b < t_e \leq \mathcal{T}_2$. The next step in these cases is to compute the six constant coefficients m_1, m_2, m_3, l_1, l_2, and l_3 for the generalized form of the C-function as a function of t. The values of these coefficients are examined to determine to which class and to which case within that class the C-function belongs.

If all of its coefficients are zero, then the C-function falls into case 1 of class I and can be ignored. The procedure then moves on to the next C-function. If any coefficients are nonzero, the analytic root intervals are determined according to the criteria in section 4.2. If the values of the coefficients are such that there are no analytic root intervals for the C-function, the procedure moves on to the next C-function. This will only

happen when the form of the C-function belongs to case 2 or 3 of class *I*.

If the analytic root interval is a single point $t = t_d$, then this is the only value of t where the C-function is zero. This can only happen for C-functions having the form of case 5 of class *I* or case 1 of class *II*. If t_d is not in the interval $[t_b, t_e]$, then there are no zeros of the C-function in this interval and the procedure moves on to the next C-function. If $t_b \leq t_d \leq t_e$, then this is the only zero of the C-function in this interval and it is checked to see if it meets the two criteria for allowable zeros. First, ωt_d modulo 2π is checked to see if it lies in an applicability interval of the C-function. If not, then t_d cannot be a valid collision and the procedure moves on to the next C-function. If t_d is in an applicability interval of the C-function, its orientation at t_d must be considered. When $m_1 > 0$, the C-function changes from positive to negative at t_d. If the C-function has positive orientation at t_d, then t_d is an allowable zero. When $m_1 < 0$, the C-function changes from negative to positive at t_d and t_d is an allowable zero if the C-function has negative orientation at t_d. If t_d does not meet this second criterion for an allowable zero, it cannot represent the first contact between $\mathcal{A}$ and $\mathcal{B}$ and the procedure moves on to the next C-function. If t_d is an allowable zero, the third collision condition is checked. If this condition is not satisfied at t_d, then the procedure moves on to the next C-function. If the third collision condition is satisfied, then t_d is a valid collision. The current time of first contact, T_c, is set equal to t_d and the search interval is updated by setting $\mathcal{T}_2 = t_d$. The procedure then moves on to the next C-function.

When the C-function has the form of cases 4, 6, or 7 of class *I* or cases 2, 3, 4, or 5 of class *II*, there are one or two analytic root intervals. These intervals can be bounded or unbounded. The next step in these cases is to merge the analytic root interval(s) with the bounded interval $[t_b, t_e]$. If there are no values of t where the analytic root interval(s) overlap with the interval $[t_b, t_e]$, then the C-function cannot have any zeros in that interval and the procedure moves on to the next C-function.

The analytic root intervals(s) for the C-function may overlap with the interval $[t_b, t_e]$ at t_b and/or t_e but not at any point between t_b and t_e. If only one of these points is a point of overlap, then the following steps are taken. First, the value of ωt_b or ωt_e modulo 2π is checked to see if it lies in an applicability interval for the C-function. If not, then t_b or t_e cannot represent a valid collision between $\mathcal{A}$ and $\mathcal{B}$ and the procedure moves on to the next C-function. If the C-function is applicable at t_b or t_e, then its

value is computed at t_b or t_e. If the value of the C-function at t_b or t_e is not zero, then the procedure moves on to the next C-function. If the C-function is zero at t_b or t_e, then the third collision condition is checked. If this condition is not satisfied at t_b or t_e, then the procedure moves on to the next C-function. If the third collision condition is satisfied at t_b or t_e, then t_b or t_e is a valid collision. The current time of first contact, T_c, is set equal to t_b or t_e and the search interval is updated by setting $\mathcal{T}_2 = t_b$ or t_e. The procedure then moves on to the next C-function. If both t_b and t_e must be checked, then the above steps are followed first for t_b. If t_b is found to be a valid collision, then t_e is not checked and the procedure moves on to the next C-function. If t_b is not found to be a valid collision, then the same steps are followed for t_e before moving on to the next C-function.

If there are one or two finite intervals of overlap between the analytic root interval(s) and the interval $[t_b, t_e]$, then the allowable zeros of the C-function in the(se) interval(s) of overlap must be found. By definition, an allowable zero for the C-function must lie in an interval where the C-function is applicable. Therefore, the zero search for the C-function can be confined to all of the applicability intervals or parts of these intervals that fall within an interval of overlap between the analytic root intervals and the interval $[t_b, t_e]$.

Let $[\mathsf{T}_b, \mathsf{T}_e]$ be (one of) the interval(s) of overlap between the analytic root interval(s) and the interval $[t_b, t_e]$. Those applicability intervals or parts of such intervals that fall within the interval $[\mathsf{T}_b, \mathsf{T}_e]$ can be found from the applicability timeline for the C-function. Now suppose $[\mathsf{T}_{a_1}, \mathsf{T}_{a_2}]$ is such an interval entirely contained in (one of) the interval(s) $[\mathsf{T}_b, \mathsf{T}_e]$. The numerical zero search procedures for cases 6 and 7 of class *I* and for case 5 of class *II* described in Chapter 5 can be executed to find any zeros of the C-function between T_{a_1} and T_{a_2}. Although analytic formulas exist for the zeros of the C-function for case 4 of class *I* and for case 2 of class *II*, these formulas are not used in the algorithm implemented for this thesis. The zeros of the C-function for case 4 of class *I* are found using the zero search procedure for cases 6 and 7 of class *I*. Similarly, the zeros of the C-function for case 2 of class *II* are found using the numerical search procedure for case 5 of class *II*. The zeros of the C-function for cases 3 and 4 of class *II* are also found by using the search procedure for case 5 of class *II*, as explained in section 5.2.2.4.

Only a slight modification of these two numerical procedures is necessary

to check the second criterion for allowable zeros. This second criterion for allowable zeros states that the value of the C-function must approach zero from a region where its value is positive if it has positive orientation or from a region where its value is negative if it has negative orientation. The orientation of the C-function in an interval $[\mathsf{T}_{a_1}, \mathsf{T}_{a_2}]$ is found from the orientation timeline. In the case where a zero lies between two test points, the search procedure need only distinguish between the cases where the value of the C-function is positive at the first point and negative at the second point and vice-versa. If the C-function has positive orientation and the value of the C-function changes from positive to negative between two consecutive test points, then there is a single allowable zero between these two points. Similarly, if the C-function has negative orientation and its value changes from negative to positive between two consecutive test points, there is a single allowable zero of the C-function between them. In either of these cases, the procedure uses the secant method to find the allowable zero between the two test points. If the sign change occurs in the opposite way from that described above, then the single zero of the C-function between the two test points is not allowable and cannot represent the first contact between $\mathcal{A}$ and $\mathcal{B}$. In these cases, the procedure ignores the sign change between the two test points and goes on to check the next pair of test points.

There are two special cases for an allowable zero that are not covered by the above modification. A C-function with positive orientation can have an allowable zero at one of its relative minima and a C-function with negative orientation can have an allowable zero at one of its relative maxima. Fortunately, the relative maxima and minima of the C-function are included in the test points for the zero search procedure. When the value of the C-function is zero at one of the test points, the procedure checks to see if it is a relative maximum or minimum of the C-function. If so, the orientation of the C-function determines if it is an allowable zero.

When the numerical search procedure has found an allowable zero of the C-function at some time t', the third collision condition is checked at t'. The set of applicable C-functions at t' is found by comparing the value of $\omega t'$ modulo 2π with the applicability intervals stored in the applicability timelines of the C-functions. The values of all the other applicable C-functions at t', besides the one that is zero, are computed and their orientations at t' are determined from the orientation timelines. If these values are such that

the third collision condition is not satisfied, then the procedure continues to search for allowable zeros of the current C-function. If the values of the other applicable C-functions at t' are such that the third collision condition is satisfied, then t' is a valid collision between $\mathcal{A}$ and $\mathcal{B}$. The zero search for the current C-function is terminated. The time of first contact, T_c, and the upper bound of the search interval for the remaining C-functions, $\mathcal{T}_2$, are set equal to t'. The procedure moves on to check the next C-function using the updated search interval.

In summary, there are five conditions under which the numerical zero search procedure need not be performed for a single C-function:

1. The C-function is never applicable for any value of ωt. In this case, any zeros of the C-function do not satisfy the second collision condition.

2. The physical constraint for the C-function is not satisfied for any value of t. In this case, any zeros of the C-function do not satisfy the third collision condition.

3. There is no overlap between the current search interval $[\mathcal{T}_1, \mathcal{T}_2]$ and the interval $[t_i, t_f]$ where the physical constraint of the C-function is satisfied. Any allowable zeros of the C-function in the interval $[\mathcal{T}_1, \mathcal{T}_2]$ cannot satisfy the third collision condition.

4. There are no analytic root intervals for the C-function. In this case, the form of the C-function is such that it has no zeros.

5. The C-function has analytic root interval(s) but there is no overlap between its analytic root interval(s) and the interval $[t_b, t_e]$ where the search interval and the physical constraint interval overlap. The C-function has no zeros in the interval $[t_b, t_e]$.

6.2.4 The Hierarchy of the Algorithm

The overall hierarchy of the collision detection algorithm is summarized in the flowchart given in Figure 6.2. The flowchart shows interaction of the four major procedures described in the preceding sections. This interaction is discussed in greater detail below.

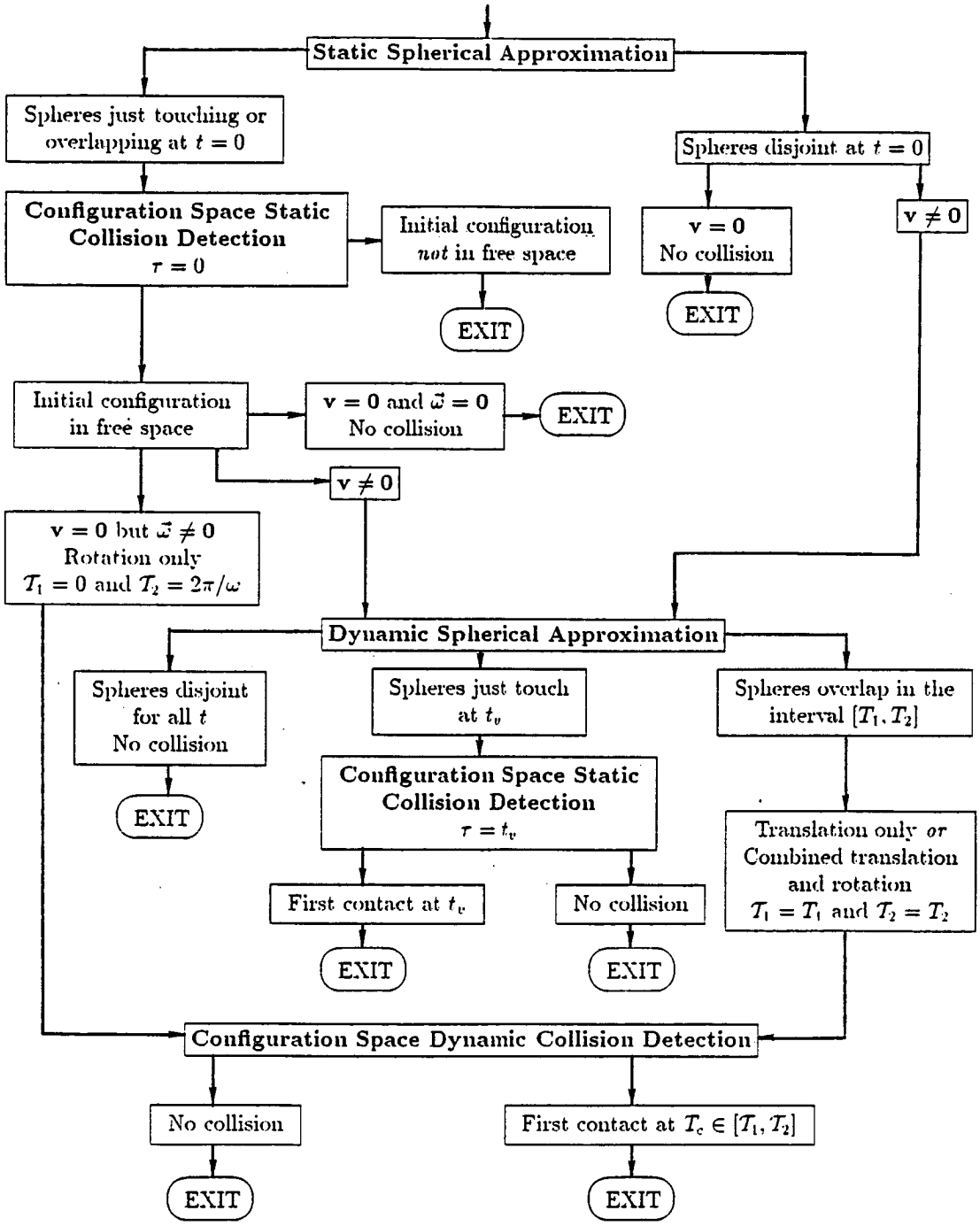


Figure 6.2: The flowchart for the collision detection algorithm

The first step in the collision detection algorithm is to check that the initial configuration of $\mathcal{A}$ given by $\mathbf{x}_o$ and $\mathbf{Q}_o$ is located in free space. The algorithm executes the static spherical approximation procedure. This determines the initial relationship of the spheres enclosing $\mathcal{A}$ and $\mathcal{B}$. There are two possible outcomes of this procedure. If the static spherical approximation procedure indicates that the two spheres are either just touching or overlapping at $t = 0$, no conclusion can be drawn regarding the location of the initial configuration. Further tests are necessary to determine whether or not it is in free space. The algorithm executes the configuration space static collision detection procedure with $\tau = 0$.

There are two possible outcomes of the configuration space static collision detection procedure. If this procedure indicates that $\mathcal{A}$ and $\mathcal{B}$ are just touching or overlapping at $t = 0$, then the algorithm prints a message that the initial configuration is not in free space. If the procedure indicates that $\mathcal{A}$ and $\mathcal{B}$ are disjoint at $t = 0$, then the initial configuration is located in free space and the algorithm checks the linear and angular velocity vectors. For the trivial case where both $\mathbf{v}$ and $\vec{\omega}$ are zero vectors and $\mathcal{A}$ is stationary relative to $\mathcal{B}$, the algorithm prints a message that no collision can occur. If $\mathbf{v} = \mathbf{0}$ and $\vec{\omega} \neq \mathbf{0}$, then $\mathcal{A}$ is rotating but not translating. The rotational motion of $\mathcal{A}$ is periodic and the first contact between $\mathcal{A}$ and $\mathcal{B}$, if any, *must* occur between $t = 0$ and the end of the first cycle of rotation $t = 2\pi/\omega$. The configuration space dynamic collision detection procedure is executed with $\mathcal{T}_1 = 0$ and $\mathcal{T}_2 = 2\pi/\omega$. If $\mathbf{v} \neq \mathbf{0}$, then $\mathcal{A}$ is translating and the algorithm executes the dynamic spherical approximation procedure.

If the static spherical approximation procedure indicates that the two spheres are disjoint at $t = 0$, then the initial configuration *must* be in free space. If $\mathcal{A}$ is not translating, that is if $\mathbf{v} = \mathbf{0}$, then $\mathcal{A}$ cannot touch $\mathcal{B}$. The algorithm prints a message that no collision can occur. If $\mathcal{A}$ is translating, $\mathbf{v} \neq \mathbf{0}$ and the algorithm executes the dynamic spherical approximation procedure.

There are three possible outcomes of the dynamic spherical approximation procedure. If the procedure indicates that the spheres enclosing $\mathcal{A}$ and $\mathcal{B}$ do not overlap for any $t \geq 0$, then no contact is possible between $\mathcal{A}$ and $\mathcal{B}$. The algorithm prints a message that no collision can occur. If the procedure indicates that the spheres are just touching at $t_v > 0$, then contact between $\mathcal{A}$ and $\mathcal{B}$ is possible only at t_v. The algorithm executes the configuration space static collision detection procedure with $\tau = t_v$. If

the static procedure indicates that $\mathcal{A}$ and $\mathcal{B}$ are in contact at t_v, the algorithm prints a message that the first contact between $\mathcal{A}$ and $\mathcal{B}$ will occur at t_v. If the static procedure indicates that $\mathcal{A}$ and $\mathcal{B}$ are not in contact at t_v, the algorithm prints a message that no collision will occur. If the dynamic spherical approximation procedure indicates that the spheres enclosing $\mathcal{A}$ and $\mathcal{B}$ overlap for a finite interval $[T_1, T_2]$ where $T_2 > T_1 \geq 0$, then the algorithm executes the configuration space dynamic collision detection procedure with $\mathcal{T}_1 = T_1$ and $\mathcal{T}_2 = T_2$.

The interval $[\mathcal{T}_1, \mathcal{T}_2]$ that is input to the configuration space dynamic collision detection procedure is chosen such that the first collision, if any, must occur at some time $\mathcal{T}_1 \leq t \leq \mathcal{T}_2$. There are two possible outcomes of this procedure. The first outcome is that no valid collision is found for $\mathcal{A}$ and $\mathcal{B}$ in the given interval $[\mathcal{T}_1, \mathcal{T}_2]$. In this case, the algorithm prints a message that no collision will occur. The second outcome is that a valid collision is found between $\mathcal{A}$ and $\mathcal{B}$. In this case, the algorithm prints the resulting time of the first contact, T_c.

6.3 An Example

The interaction of the various procedures used in the collision detection algorithm will now be illustrated by an example. The moving object $\mathcal{A}$ and the obstacle $\mathcal{B}$ will be represented by the two rectangular boxes introduced in section 3.3.1. Figure 3.8 shows the sizes and the numbering of the features of these two boxes.

6.3.1 The Inputs

In this case, both $\mathcal{A}$ and $\mathcal{B}$ are single convex polyhedra so that $n_a = 1$ and $n_b = 1$. The center of rotation of $\mathcal{A}$, C^a, is located at the geometric center of a rectangular box with dimensions $l_x = 8$, $l_y = 6$, and $l_z = 24$. Therefore, the radius of the smallest sphere that completely encloses $\mathcal{A}$ is $r_a = 13$. The origin of the reference coordinate system is located at the geometric center of the rectangular box representing $\mathcal{B}$. The dimensions of this box are $l_x = 32$, $l_y = 24$, and $l_z = 96$. Following the procedure described in section 4.3, the location of C^b, the center of the sphere enclosing $\mathcal{B}$, is found to be $\mathbf{x}^b = \mathbf{0}$ and its radius is $r_b = 52$. Because of the symmetry of the

rectangular box, these values of r_b and $\mathbf{x}^b$ represent the exact solution for the sphere of smallest radius that completely encloses $\mathcal{B}$.

The datastructures for $\mathcal{A}$ and $\mathcal{B}$ have only one substructure for each of the single convex polyhedra representing $\mathcal{A}$ and $\mathcal{B}$. Each box has eight vertices, twelve edges, and six faces so that $v_{\mathcal{A}} = v_{\mathcal{B}} = 8$, $e_{\mathcal{A}} = e_{\mathcal{B}} = 12$, and $f_{\mathcal{A}} = f_{\mathcal{B}} = 6$. Because $\mathcal{A}$ and $\mathcal{B}$ have the same shape and their features are numbered in the same way, the connectivity information for each of these objects is the same. The vertex-edge and edge-face connectivity matrices and the tail vertices chosen for each edge vector are shown in Table 6.1. The numerical data in the substructures for $\mathcal{A}$ and $\mathcal{B}$ are given in Tables 6.2 and 6.3.

Vertex-Edge Connectivity Matrix

$$\begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Edge-Face Connectivity Matrix

$$\begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Tail Vertices for Edge Vectors

Edge #	1	2	3	4	5	6	7	8	9	10	11	12
Tail Vertex #	1	2	3	4	5	6	7	8	5	6	8	7

Table 6.1: The connectivity information for the rectangular boxes representing $\mathcal{A}$ and $\mathcal{B}$

Vertex vectors for $\mathcal{A}$

$$\begin{array}{lllllll}
\mathbf{a}_1 & = & [\ -4 & +3 & +12\ ]^T & \mathbf{a}_5 & = & [\ -4 & -3 & +12\ ]^T \\
\mathbf{a}_2 & = & [\ -4 & +3 & -12\ ]^T & \mathbf{a}_6 & = & [\ -4 & -3 & -12\ ]^T \\
\mathbf{a}_3 & = & [\ +4 & +3 & +12\ ]^T & \mathbf{a}_7 & = & [\ +4 & -3 & -12\ ]^T \\
\mathbf{a}_4 & = & [\ +4 & +3 & +12\ ]^T & \mathbf{a}_8 & = & [\ +4 & -3 & +12\ ]^T
\end{array}$$

Edge vectors for $\mathcal{A}$

$$\begin{array}{lllllll}
\mathbf{e}_1^a & = & [\ 0 & 0 & -24\ ]^T & \mathbf{e}_7^a & = & [\ 0 & 0 & +24\ ]^T \\
\mathbf{e}_2^a & = & [\ +8 & 0 & 0\ ]^T & \mathbf{e}_8^a & = & [\ -8 & 0 & 0\ ]^T \\
\mathbf{e}_3^a & = & [\ 0 & 0 & +24\ ]^T & \mathbf{e}_9^a & = & [\ 0 & +6 & 0\ ]^T \\
\mathbf{e}_4^a & = & [\ -8 & 0 & 0\ ]^T & \mathbf{e}_{10}^a & = & [\ 0 & +6 & 0\ ]^T \\
\mathbf{e}_5^a & = & [\ 0 & 0 & -24\ ]^T & \mathbf{e}_{11}^a & = & [\ 0 & +6 & 0\ ]^T \\
\mathbf{e}_6^a & = & [\ +8 & 0 & 0\ ]^T & \mathbf{e}_{12}^a & = & [\ 0 & +6 & 0\ ]^T
\end{array}$$

Unit, outward-pointing normal vectors for faces of $\mathcal{A}$

$$\begin{array}{lllllll}
\mathbf{u}_1^a & = & [\ -1 & 0 & 0\ ]^T & \mathbf{u}_4^a & = & [\ 0 & -1 & 0\ ]^T \\
\mathbf{u}_2^a & = & [\ +1 & 0 & 0\ ]^T & \mathbf{u}_5^a & = & [\ 0 & 0 & +1\ ]^T \\
\mathbf{u}_3^a & = & [\ 0 & +1 & 0\ ]^T & \mathbf{u}_6^a & = & [\ 0 & 0 & -1\ ]^T
\end{array}$$

Distances of faces from C^a along face normal vectors for $\mathcal{A}$

$$\begin{array}{lllll}
d_1^a & = & +4 & d_4^a & = & +3 \\
d_2^a & = & +4 & d_5^a & = & +12 \\
d_3^a & = & +3 & d_6^a & = & +12
\end{array}$$

Table 6.2: The numerical data for the vertices, edges, and faces of $\mathcal{A}$

The initial position chosen for $\mathcal{A}$ is $\mathbf{x}_o = [\ 10 \quad 72 \quad -25\ ]^T$. The initial orientation of $\mathcal{A}$ is such that its body coordinate axes are aligned with the reference coordinate axes. The initial yaw, pitch, and roll angles are all zero and the initial quaternion is simply the identity quaternion $\mathbf{Q}_o = [\ 1 \quad \mathbf{0}^T\ ]^T$. The linear velocity vector is chosen to point from $\mathbf{x}_o$ to the center of face 3 of $\mathcal{B}$ and the linear speed is set to 1 unit of length per unit of time. The resulting value of $\mathbf{v}$ is

$$\mathbf{v} = [\ -0.15205718 \quad -0.91234311 \quad +0.38014296\ ]^T$$

The rotation axis is the same as that chosen for the examples of section 3.3.1 and the angular speed is set equal to 55 degrees per unit of time. The value

Vertex vectors for $\mathcal{B}$

$$\begin{array}{lllllll}
\mathbf{b}_1 & = & [\,-16 & +12 & +48\,]^T & \mathbf{b}_5 & = & [\,-16 & -12 & +48\,]^T \\
\mathbf{b}_2 & = & [\,-16 & +12 & -48\,]^T & \mathbf{b}_6 & = & [\,-16 & -12 & -48\,]^T \\
\mathbf{b}_3 & = & [\,+16 & +12 & +48\,]^T & \mathbf{b}_7 & = & [\,+16 & -12 & -48\,]^T \\
\mathbf{b}_4 & = & [\,+16 & +12 & +48\,]^T & \mathbf{b}_8 & = & [\,+16 & -12 & +48\,]^T
\end{array}$$

Edge vectors for $\mathcal{B}$

$$\begin{array}{lllllll}
\mathbf{e}_1^b & = & [\,0 & 0 & -96\,]^T & \mathbf{e}_7^b & = & [\,0 & 0 & +96\,]^T \\
\mathbf{e}_2^b & = & [\,+32 & 0 & 0\,]^T & \mathbf{e}_8^b & = & [\,-32 & 0 & 0\,]^T \\
\mathbf{e}_3^b & = & [\,0 & 0 & +96\,]^T & \mathbf{e}_9^b & = & [\,0 & +24 & 0\,]^T \\
\mathbf{e}_4^b & = & [\,-32 & 0 & 0\,]^T & \mathbf{e}_{10}^b & = & [\,0 & +24 & 0\,]^T \\
\mathbf{e}_5^b & = & [\,0 & 0 & -96\,]^T & \mathbf{e}_{11}^b & = & [\,0 & +24 & 0\,]^T \\
\mathbf{e}_6^b & = & [\,+32 & 0 & 0\,]^T & \mathbf{e}_{12}^b & = & [\,0 & +24 & 0\,]^T
\end{array}$$

Unit, outward-pointing normal vectors for faces of $\mathcal{B}$

$$\begin{array}{lllllll}
\mathbf{u}_1^b & = & [\,-1 & 0 & 0\,]^T & \mathbf{u}_4^b & = & [\,0 & -1 & 0\,]^T \\
\mathbf{u}_2^b & = & [\,+1 & 0 & 0\,]^T & \mathbf{u}_5^b & = & [\,0 & 0 & +1\,]^T \\
\mathbf{u}_3^b & = & [\,0 & +1 & 0\,]^T & \mathbf{u}_6^b & = & [\,0 & 0 & -1\,]^T
\end{array}$$

Distances of faces from origin of the reference coordinate system along face normal vectors for $\mathcal{B}$

$$\begin{array}{lllllll}
d_1^b & = & +16 & d_4^b & = & +12 \\
d_2^b & = & +16 & d_5^b & = & +48 \\
d_3^b & = & +12 & d_6^b & = & +48
\end{array}$$

Table 6.3: The numerical data for the vertices, edges, and faces of $\mathcal{B}$

of $\vec{\omega}$ is

$$\begin{aligned}\vec{\omega} &= (55°/t)\,[\ -0.11547005 \quad -0.57735027 \quad +0.80829038\ ]^T \\ &= [\ -6.3508528 \quad -31.754265 \quad 44.455971\ ]^T\,(°/t)\end{aligned}$$

The initial configuration of $\mathcal{A}$ relative to $\mathcal{B}$ is shown in Figure 6.3. The rotation axis and the direction of the velocity vector are also indicated in this figure.

6.3.2 The Algorithm

The first step in the collision detection algorithm is to execute the static spherical approximation procedure. For this example, $\|\mathbf{x}_o - \mathbf{x}^b\| = \|\mathbf{x}_o\| = 76.87$ and $R = 13 + 52 = 65$. The spheres enclosing $\mathcal{A}$ and $\mathcal{B}$ are initially disjoint because $\|\mathbf{x}_o - \mathbf{x}^b\| > R$. This means that the initial configuration is in free space. There is no need to execute the configuration space static collision detection procedure at $t = 0$.

Now, $\|\mathbf{v}\| = 1 \neq 0$. Therefore, the algorithm executes the dynamic spherical approximation procedure. The result of this procedure is that the spheres enclosing $\mathcal{A}$ and $\mathcal{B}$ overlap between $T_1 = 11.899$ and $T_2 = 141.527$. This leads to execution of the configuration space dynamic collision detection procedure with $\mathcal{T}_1 = T_1 = 11.899$ and $\mathcal{T}_2 = T_2 = 141.527$.

The configuration space dynamic collision detection procedure examines each of the C-functions defined by $\mathcal{A}$ and $\mathcal{B}$. Since $\mathcal{A}$ and $\mathcal{B}$ are rectangular boxes, there are $f_{\mathcal{A}} \cdot v_{\mathcal{B}} = 48$ type (a) C-functions, $v_{\mathcal{A}} \cdot f_{\mathcal{B}} = 48$ type (b) C-functions, and $e_{\mathcal{A}} \cdot e_{\mathcal{B}} = 144$ type (c) C-functions, for a total of 240 C-functions. The applicability and orientation timelines for each of these C-functions are computed and stored. Examples of finding these timelines for each type of C-function were given in section 3.3.1.

The examination of the C-functions determines that the first contact between $\mathcal{A}$ and $\mathcal{B}$ is a type (b) contact when vertex 5 of $\mathcal{A}$ touches face 3 of $\mathcal{B}$ at $T_c = 54.496$. In other words, the zero of $f^b_{5,3}$ that occurs at $T_c = 54.496$ is the smallest allowable zero found that represents a valid collision between $\mathcal{A}$ and $\mathcal{B}$. The configuration of $\mathcal{A}$ relative to $\mathcal{B}$ at the time of first contact is shown in Figure 6.4.

The results of the examination of all 240 C-functions are summarized in Table 6.4. This table shows the number of C-functions of each type satisfying each of the five conditions listed at the end of section 6.2.3.1. For

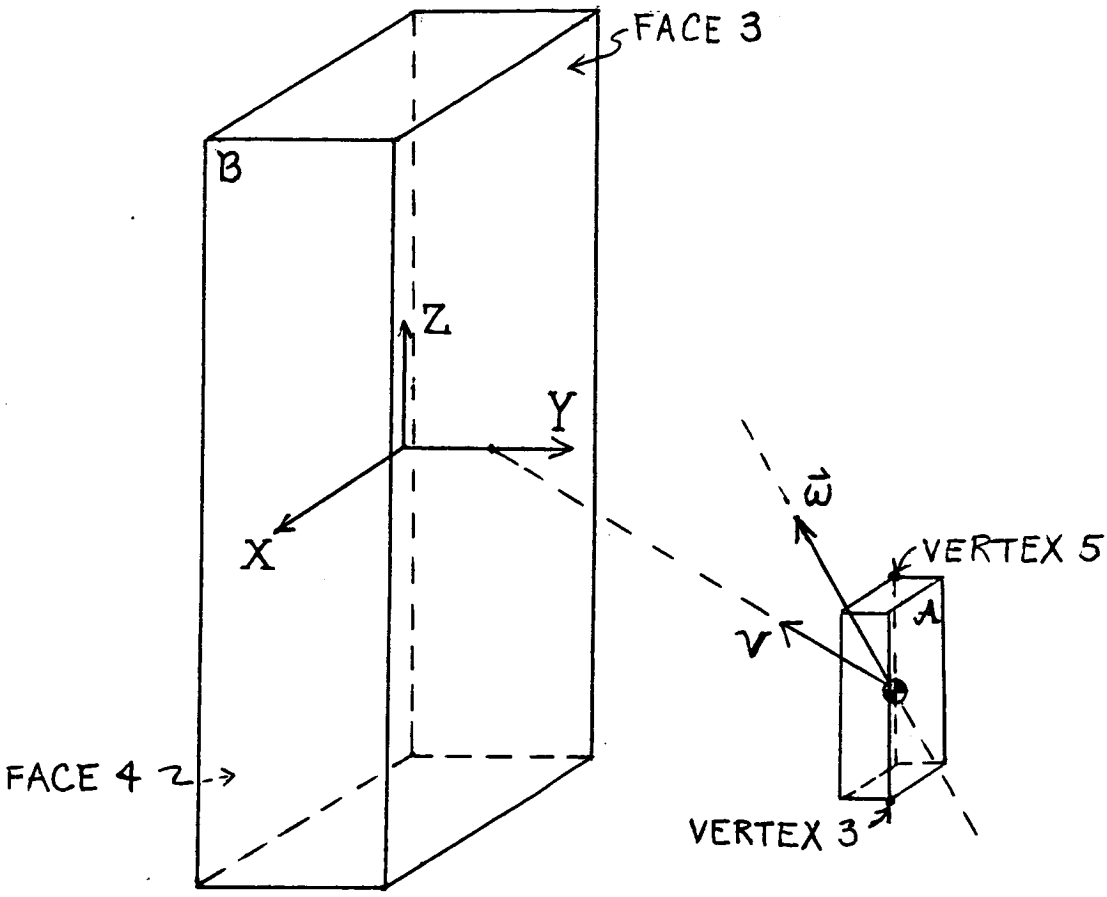


Figure 6.3: The initial configuration of $\mathcal{A}$ and $\mathcal{B}$

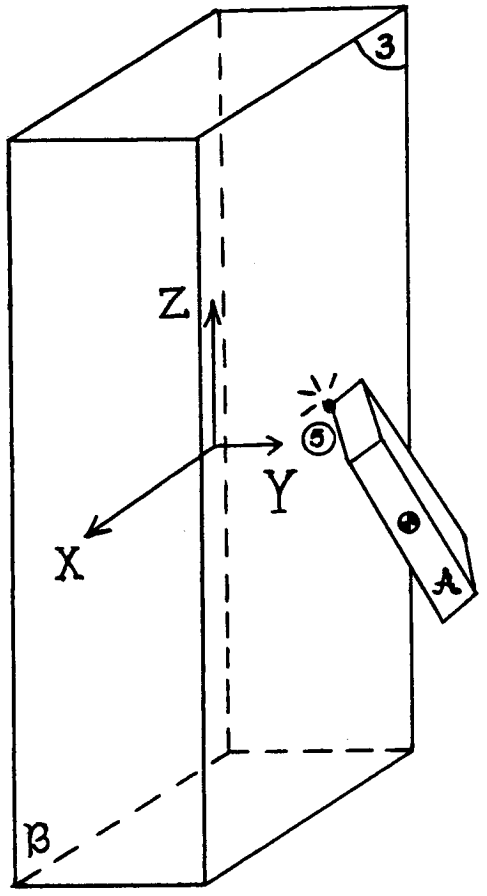


Figure 6.4: The configuration of $\mathcal{A}$ and $\mathcal{B}$ at the time of first contact, $T_c = 54.496$

Description	Type (a)	Type (b)	Type (c)	Total
Initial # of C-functions	48	48	144	240
# of C-functions that are never applicable	22	22	56	100
# of C-functions not satisfying the physical constraint for any value of t	26	0	67	93
# of C-functions for which the physical constraint is not satisfied in the interval $[\mathcal{T}_1, \mathcal{T}_2]$	—	10	21	31
# of C-functions that cannot have zeros for any value of t	—	0	—	0
# of C-functions that cannot have zeros in the interval $[t_b, t_e]$	—	2	—	2
# of C-functions having no allowable zeros in the interval $[t_b, t_e]$	—	12	—	12
# of C-functions having allowable zeros in the interval $[t_b, t_e]$	—	2	—	2

Table 6.4: Summary of the examination of the C-funcions

this example, all but 14 of the original 240 C-functions are eliminated from the numerical zero search due to one of these five conditions. Of the 14 C-functions for which the numerical zero search procedure was executed, only 2 of them were found to have any allowable zeros. The type (b) C-function $f^b_{5,3}$ representing the first contact between $\mathcal{A}$ and $\mathcal{B}$ is one of these two C-functions.

6.3.2.1 The Type (b) Zero Search

The type (b) C-functions are the first C-functions to be examined by the configuration space dynamic collision detection procedure. The examination of four of these C-functions will be discussed below. Three of these functions are chosen because they each satisfy one of the conditions under which a numerical search for the allowable zeros of the C-function is unnecessary. The fourth C-function is an example where no allowable zeros of the function are found even though the numerical zero search is executed. It also illustrates the interaction of all of the possible restrictions to the zero search from applicability, the physical constraints, and the analytic root intervals. In addition to these four type (b) C-functions, the two type (b) C-functions having zeros representing valid collisions between $\mathcal{A}$ and $\mathcal{B}$ are discussed. The next six sections present these type (b) C-functions in the same order in which they are considered by the algorithm.

$f^b_{1,1}$ The arbitrary choice of numbers for the features of $\mathcal{A}$ and $\mathcal{B}$ causes the type (b) C-function $f^b_{1,1}$ to be the first C-function examined by the algorithm. This C-function is applicable in the interval $\omega t \in [198.37°, 269.23°]$. Since this is the first C-function to be examined, the current search interval is the interval found by the dynamic spherical approximation procedure. Therefore, $\mathcal{T}_1 = T_1 = 11.899$ and $\mathcal{T}_2 = T_2 = 141.527$. The physical constraint for this C-function is satisfied between $t_i = 85.49$ and $t_f = 256.5$. The physical constraint interval overlaps the search interval between $t_b = t_i = 85.49$ and $t_e = \mathcal{T}_2 = 141.527$.

The six coefficients for this C-function are such that it has the form of case 7 of class I. The analytic root interval for $f^b_{1,1}$ is bounded by $t_2 = 99.94$ and $t_1 = 229.2$. This interval overlaps with the interval $[t_b, t_e]$ between $\mathsf{T}_b = t_2 = 99.94$ and $\mathsf{T}_e = t_e = 141.527$. The numerical zero search procedure for case 7 of class I is executed in the subintervals of $[99.94, 141.527]$ where $f^b_{1,1}$

is applicable. Figure 6.5 shows a plot of the C-function in this interval. The subintervals where it is applicable are shaded with plus signs to indicate the positive orientation of the C-function. It is clear from the plot that all of the zeros of $f^b_{1,1}$ occur outside of its applicability intervals. The numerical search procedure does not find any allowable zeros for this C-function.

$\mathbf{f}^b_{1,4}$ The type (b) C-function $f^b_{1,4}$ is never applicable for any value of ωt. This is confirmed by the plots of the three ACFs for this C-function shown in Figure 6.6. There is no value of ωt where all three of the ACFs are positive. Because $f^b_{1,4}$ is never applicable, any value of t where it is equal to zero cannot represent a valid collision between $\mathcal{A}$ and $\mathcal{B}$. It is not necessary to perform a numerical search for the zeros of this C-function.

$\mathbf{f}^b_{1,6}$ The type (b) C-function $f^b_{1,6}$ is applicable in the interval $\omega t \in [332.2°, 360°]$. The current search interval at the time of its examination is still the interval found by the dynamic spherical approximation procedure so that $\mathcal{T}_1 = T_1 = 11.899$ and $\mathcal{T}_2 = T_2 = 141.527$. The physical constraint for this C-function is satisfied between $t_i = -94.7$ and $t_f = -26.3$. Figure 6.7 depicts the physical constraint region for this C-function and the sphere of radius R centered at C^b that represents the region where the spheres enclosing $\mathcal{A}$ and $\mathcal{B}$ overlap. The trajectory of C^a never passes through the volume where the physical constraint region intersects this sphere. This confirms the fact that the physical constraint interval does not overlap with the search interval. Any zero of $f^b_{1,6}$ in the search interval cannot satisfy the third collision condition and, hence, cannot represent a valid collision between $\mathcal{A}$ and $\mathcal{B}$. The numerical zero search procedure is not executed.

$\mathbf{f}^b_{3,4}$ The type (b) C-function $f^b_{3,4}$ is applicable in the interval $\omega t \in [27.8°, 120.0°]$. The current search interval at the time of its examination is still the interval between $\mathcal{T}_1 = T_1 = 11.899$ and $\mathcal{T}_2 = T_2 = 141.527$. The physical constraint for this C-function is satisfied between $t_i = 77.8$ and $t_f = 106.3$. The physical constraint interval is entirely contained in the search interval so that $t_b = t_i = 77.8$ and $t_e = t_f = 106.3$.

The six coefficients for this C-function are such that its form is that of case 7 of class *I*. The analytic root interval for $f^b_{3,4}$ is bounded by $t_2 = 94.9$ and $t_1 = 104.3$. This interval is entirely contained in the interval between

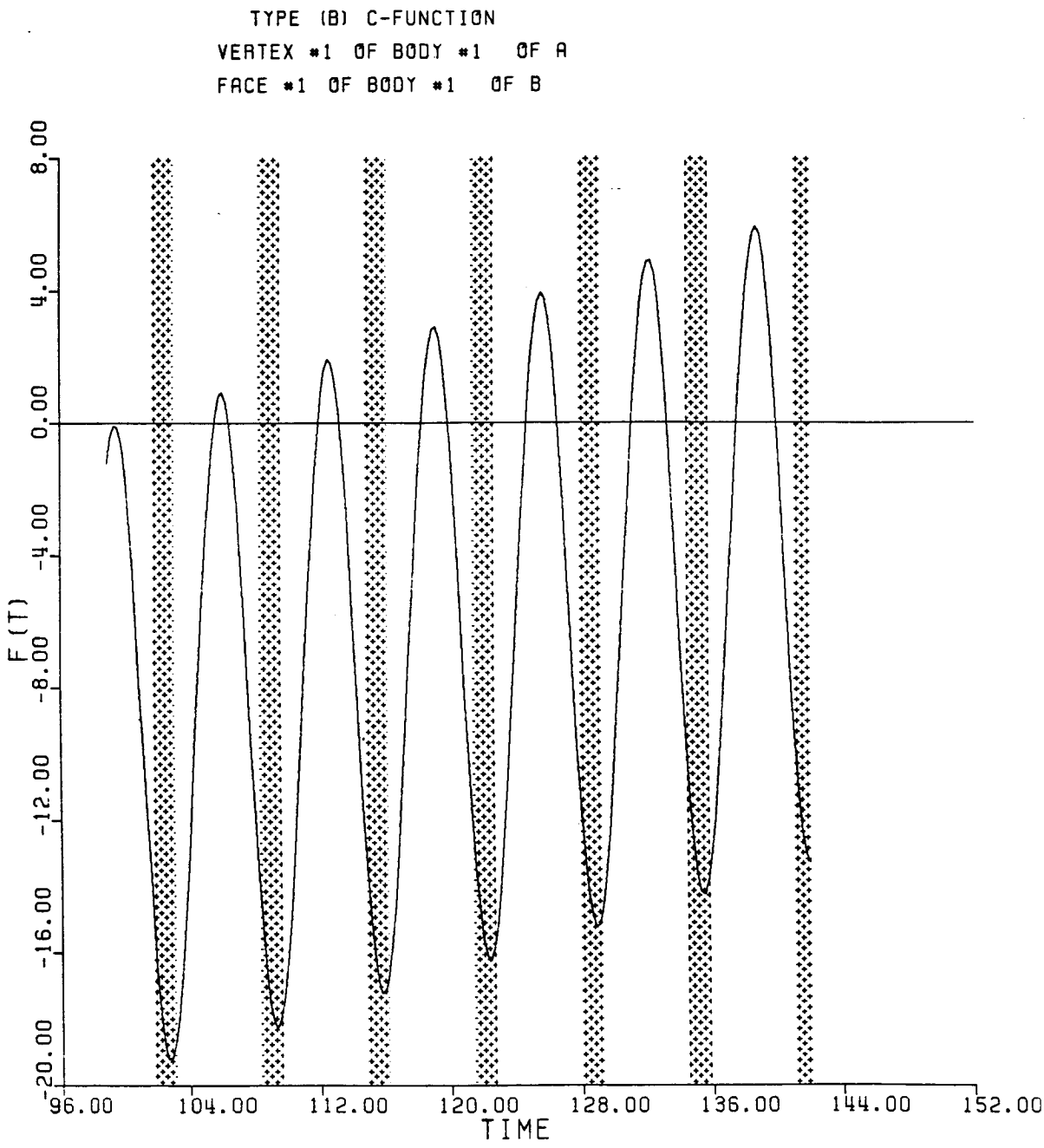


Figure 6.5: The type (b) C-function $f^b_{1,1}$ in the interval [99.94, 141.527]

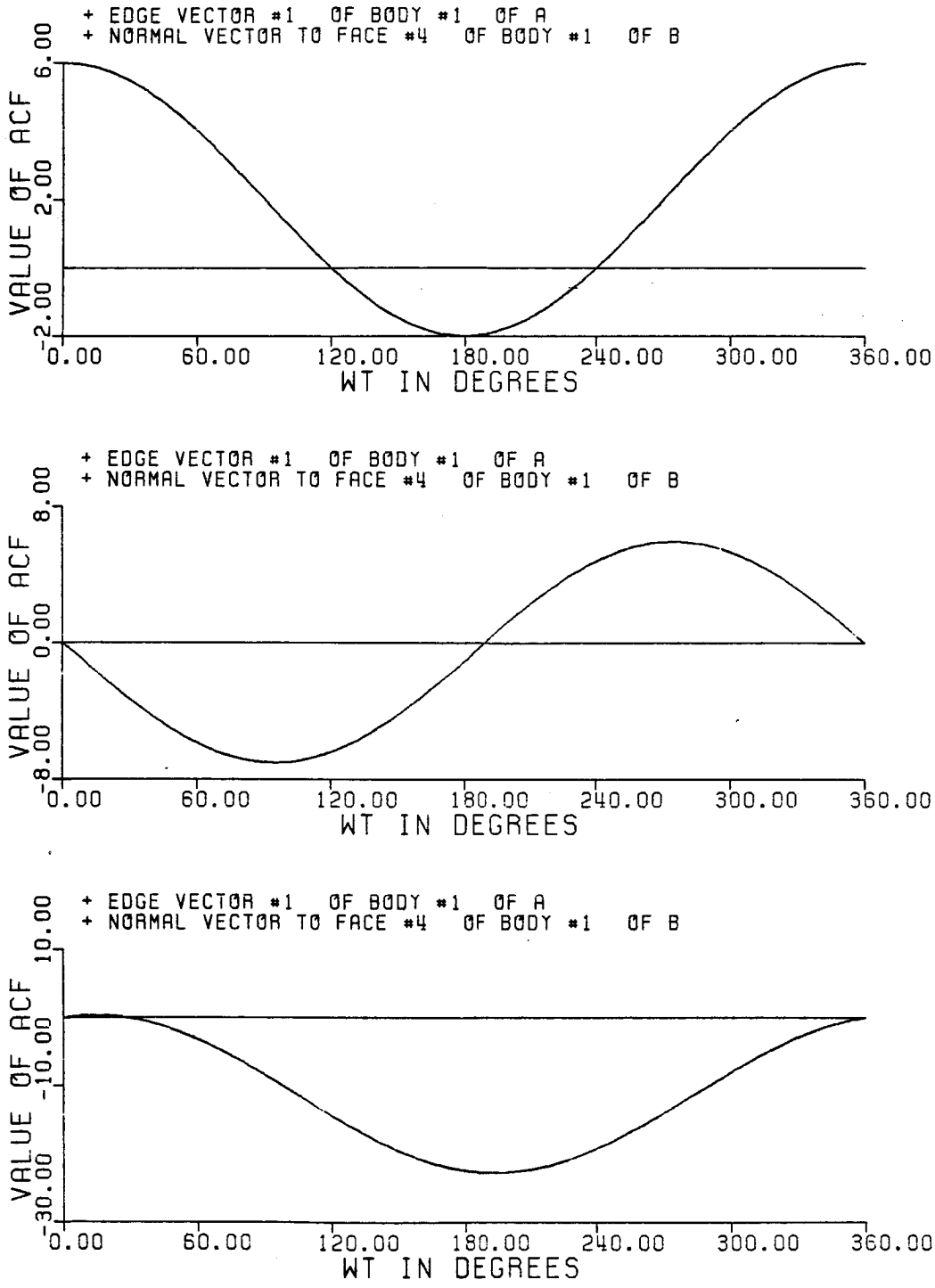
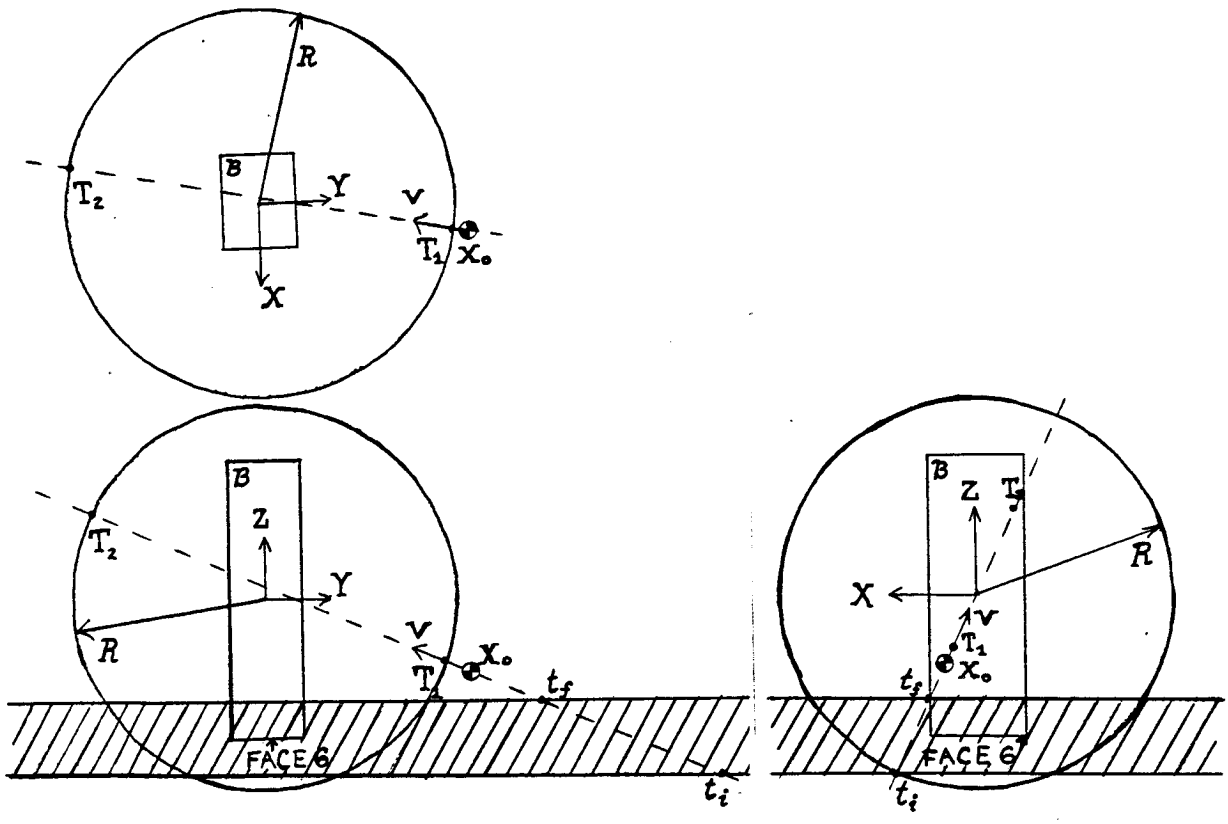


Figure 6.6: The three ACFs for the type (b) C-function $f^b_{1,4}$



The volume representing the physical constraint region for $f^b_{1,6}$

Figure 6.7: The physical constraint region for $f^b_{1,6}$

t_b and t_e. Therefore, $\mathsf{T}_b = t_2 = 94.9$ and $\mathsf{T}_e = t_1 = 104.3$. The numerical zero search procedure for case 7 of class I is executed in the subintervals of $[94.9, 104.3]$ where $f^b_{3,4}$ is applicable.

Figure 6.8 shows a plot of this type (b) C-function in the interval $[94.9, 104.3]$. There is only one subinterval where it is applicable. This subinterval is shaded with plus signs to indicate the positive orientation of the C-function. There is a single zero of $f^b_{3,4}$ in this subinterval. From the plot, it can be seen that the value of the function changes from positive to negative at this zero so that the zero is allowable. The numerical search procedure iterates to find that the zero is located at $t' = 99.28$. The value of $\omega t'$ modulo $360°$ is $60.4°$, which verifies that $f^b_{3,4}$ is applicable at t'. The third collision condition is satisfied at this zero of $f^b_{3,4}$. Thus, $t' = 99.28$ is a valid collision between vertex 3 of $\mathcal{A}$ and face 4 of $\mathcal{B}$. The time of first contact, T_c, is set equal to t' and the upper bound of the search interval for the remaining C-functions becomes $\mathcal{T}_2 = t' = 99.28$.

It is clear from the trajectory of $\mathcal{A}$, that the collision found for $f^b_{3,4}$ cannot be the true first contact between $\mathcal{A}$ and $\mathcal{B}$. $\mathcal{A}$ must pass through face 3 before it can touch face 4 of $\mathcal{B}$. The zero at $t' = 99.28$ represents a collision where $\mathcal{A}$ has emerged from inside of $\mathcal{B}$ and its third vertex has rotated back to touch face 4 of $\mathcal{B}$. In other words, the trajectory of the configuration of $\mathcal{A}$ has emerged from inside of $\mathbf{CO}^{\mathcal{A}}_{\mathcal{B}}$ prior to t' and then reversed direction to touch the boundary of $\mathbf{CO}^{\mathcal{A}}_{\mathcal{B}}$ again from free space at $t' = 99.28$. The algorithm finds this collision first due to the order in which the C-functions are considered. The C-function $f^b_{5,3}$ having the zero that represents the true first contact between $\mathcal{A}$ and $\mathcal{B}$ is examined after $f^b_{3,4}$.

$\mathbf{f^b_{5,3}}$ The type (b) C-function $f^b_{5,3}$ is also applicable in the interval $\omega t \in [27.8°, 120.0°]$. The current search interval, updated using the valid collision found for $f^b_{3,4}$, lies between $\mathcal{T}_1 = T_1 = 11.899$ and $\mathcal{T}_2 = 99.28$. The physical constraint for this C-function is satisfied between $t_i = 51.5$ and $t_f = 80.0$. The physical constraint interval is entirely contained in the search interval so that $t_b = t_i = 51.5$ and $t_e = t_f = 80.0$.

The values of the six coefficents for this C-function are such that it has the form of case 7 of class I. The analytic root interval for $f^b_{5,3}$ is bounded by $t_2 = 53.5$ and $t_1 = 62.9$. This interval is entirely contained in the interval between t_b and t_e. Therefore, $\mathsf{T}_b = t_2 = 53.5$ and $\mathsf{T}_e = t_1 = 62.9$. The numerical zero search procedure for case 7 of class I is executed in the

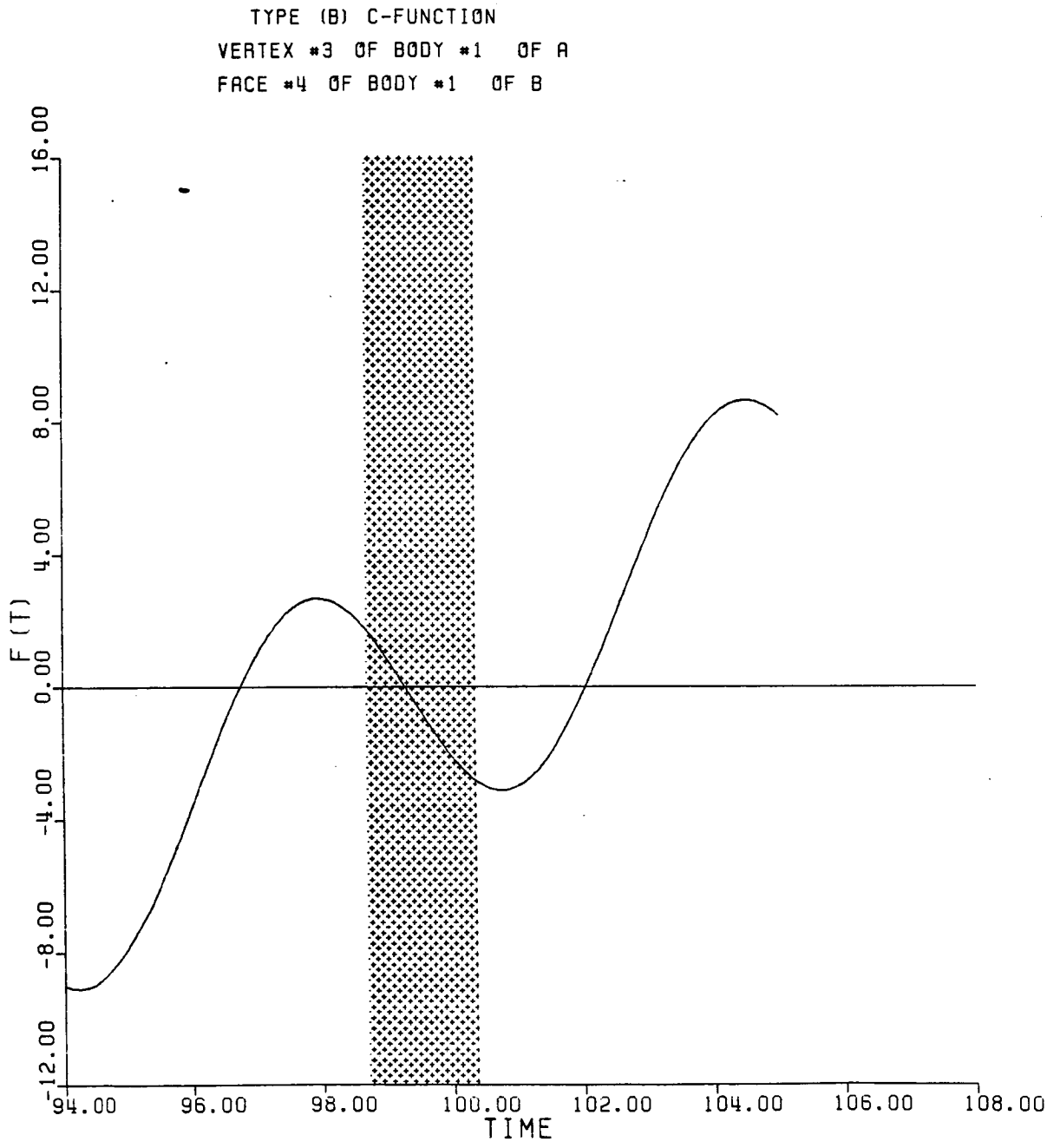


Figure 6.8: The type (b) C-function $f^b_{3,4}$ in the interval $[94.9, 104.3]$

subintervals of $[53.5, 62.9]$ where $f^b_{5,3}$ is applicable.

Figure 6.9 shows a plot of this type (b) C-function in the interval $[53.5, 62.9]$. There are two subintervals where $f^b_{5,3}$ is applicable. These subintervals are shaded with plus signs to indicate the positive orientation of the C-function. There is a single zero of $f^b_{5,3}$ in the first subinterval. From the plot, it can be seen that the value of the function changes from positive to negative at this zero so that the zero is allowable. The numerical search procedure iterates to find that the zero is located at $t' = 54.496$. The value of $\omega t'$ modulo 360° is 117.28° which lies in the applicability interval of $f^b_{5,3}$ as expected. The third collision condition is satisfied at this zero of $f^b_{5,3}$. Thus, $t' = 54.496$ is a valid collision between vertex 5 of $\mathcal{A}$ and face 3 of $\mathcal{B}$. The numerical zero search procedure for $f^b_{5,3}$ is terminated at this point. There is no need to search for zeros in the second subinterval where the C-function is applicable, although the plot shows that there is an allowable zero in this subinterval. The time of first contact, T_c, is set equal to 54.496 and the upper bound of the search interval for the remaining C-functions becomes $\mathcal{T}_2 = t' = 54.496$.

It has already been stated that the zero at $t' = 54.496$ represents the first contact between $\mathcal{A}$ and $\mathcal{B}$. However, the algorithm must continue to check the remaining type (b) C-functions and all of the type (a) and (c) C-functions after it has found this zero of $f^b_{5,3}$ to be a valid collision. It is possible that one of these C-functions may have a zero between T_1 and $t' = 54.496$ that is also a valid collision between $\mathcal{A}$ and $\mathcal{B}$. An examination of the remaining C-functions for this particular example does not yield any other zeros representing valid collisions before $t' = 54.496$.

$f^b_{6,3}$ The type (b) C-function $f^b_{6,3}$ is applicable in the interval $\omega t \in [0°, 27.8°]$. The current search interval, updated using the valid collision found for $f^b_{5,3}$, lies between $\mathcal{T}_1 = T_1 = 11.899$ and $\mathcal{T}_2 = 54.496$. The physical constraint for this C-function is satisfied between $t_i = 51.5$ and $t_f = 80.0$. The physical constraint interval overlaps with the search interval between $t_b = t_i = 51.5$ and $t_e = \mathcal{T}_2 = 54.496$.

The six coefficients for this C-function are such that it falls into case 7 of class *I*. The analytic root interval for $f^b_{6,3}$ is bounded by $t_2 = 61.0$ and $t_1 = 80.0$. For this C-function, the analytic root interval does not overlap with the interval $[t_b, t_e]$ and $f^b_{6,3}$ cannot have any zeros in that interval. Figure 6.10 shows a plot of this type (b) C-function between $t = 50$ and

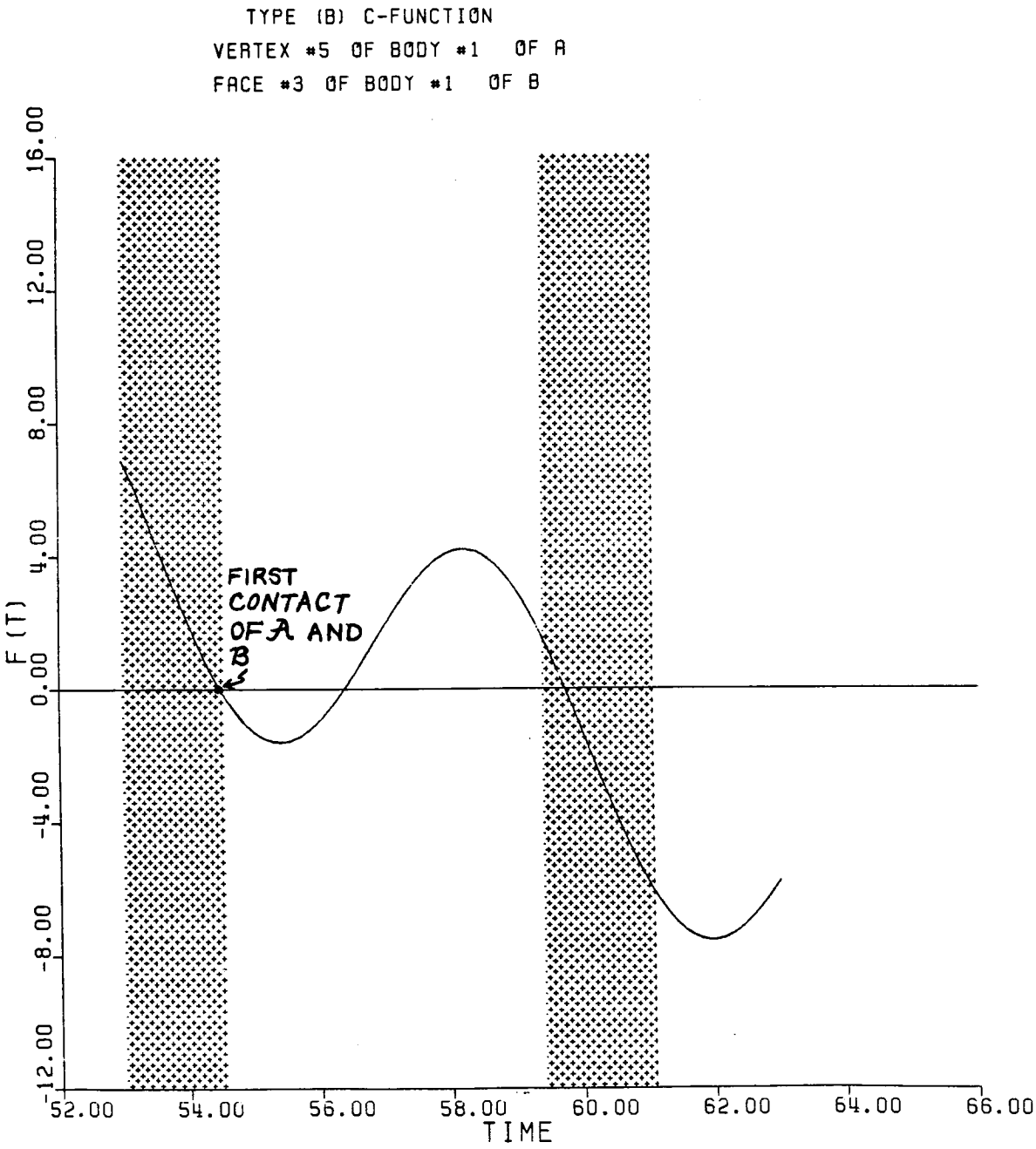


Figure 6.9: The type (b) C-function $f^b_{5,3}$ in the interval $[53.5, 62.9]$

$t = 80$. The subintervals where the C-function is applicable are shaded with plus signs to indicate the positive orientation of the C-function. This plot confirms that the zeros of $f^b_{6,3}$ are confined to the interval $[61.0, 80.0]$. Since a valid collision has already been found at $t' = 54.496$, none of the zeros of $f^b_{6,3}$ can represent the first contact between $\mathcal{A}$ and $\mathcal{B}$. There is no need to perform the numerical zero search.

6.3.2.2 The Type (a) and (c) Zero Searches

For this example, the numerical zero search procedure is not executed for any of the type (a) or type (c) C-functions. The physical constraints are never satisfied for any of the type (a) C-functions, whether or not they are applicable for some value(s) of ωt. This is due to the relative sizes of $\mathcal{A}$ and $\mathcal{B}$ and the trajectory chosen for $\mathcal{A}$. Because of the symmetry of $\mathcal{A}$, the physical constraint regions for the type (a) C-functions are the eight spheres of radius 13 centered at each of the vertices of $\mathcal{B}$. The trajectory of $\mathcal{A}$ is such that its center of rotation never passes through any of these spheres. This is illustrated in Figure 6.11. For this trajectory, none of the zeros of any type (a) C-function can satisfy the third collision condition.

Most of the type (c) C-functions that are applicable for some interval of ωt do not satisfy their physical constraint. The physical constraint regions for the type (c) C-functions are tubes of radius 13 with axes along the lines containing the twelve edges of $\mathcal{B}$. The trajectory of $\mathcal{A}$ is such that C^a only passes through the tubes around edges 5, 9, and 12 of $\mathcal{B}$. Only the type (c) C-functions involving these three edges can satisfy the physical constraint. However, for these C-functions, the interval where the physical constraint is satisfied does not overlap with the current search interval.

Figure 6.12 shows the type (c) physical constraint regions around edges 9 and 12 of $\mathcal{B}$. This figure shows that C^a is never located in the volumes where the tubes around edges 9 and 12 and the sphere of radius R centered at C^b intersect. Thus, none of the zeros of the type (c) C-functions involving edges 9 and 12 of $\mathcal{B}$ between T_1 and T_2 can satisfy the third collision condition. Since the current search interval always lies entirely between T_1 and T_2, none of the zeros of these C-functions in the current search interval can satisfy the third collision condition.

The type (c) physical constraint region around edge 5 of $\mathcal{B}$ is shown in Figure 6.13. The center of rotation of $\mathcal{A}$, C^a, passes through the tube

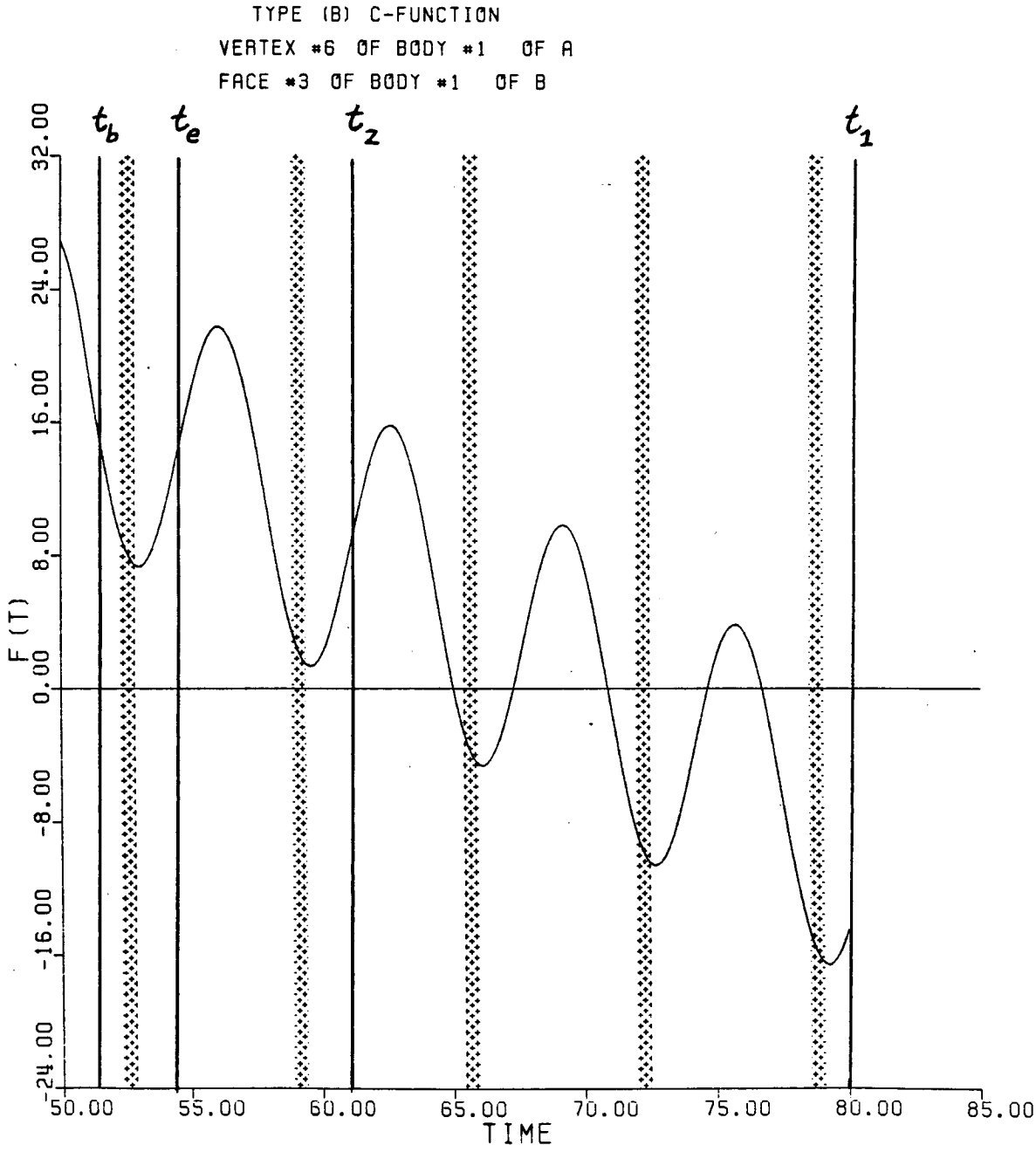
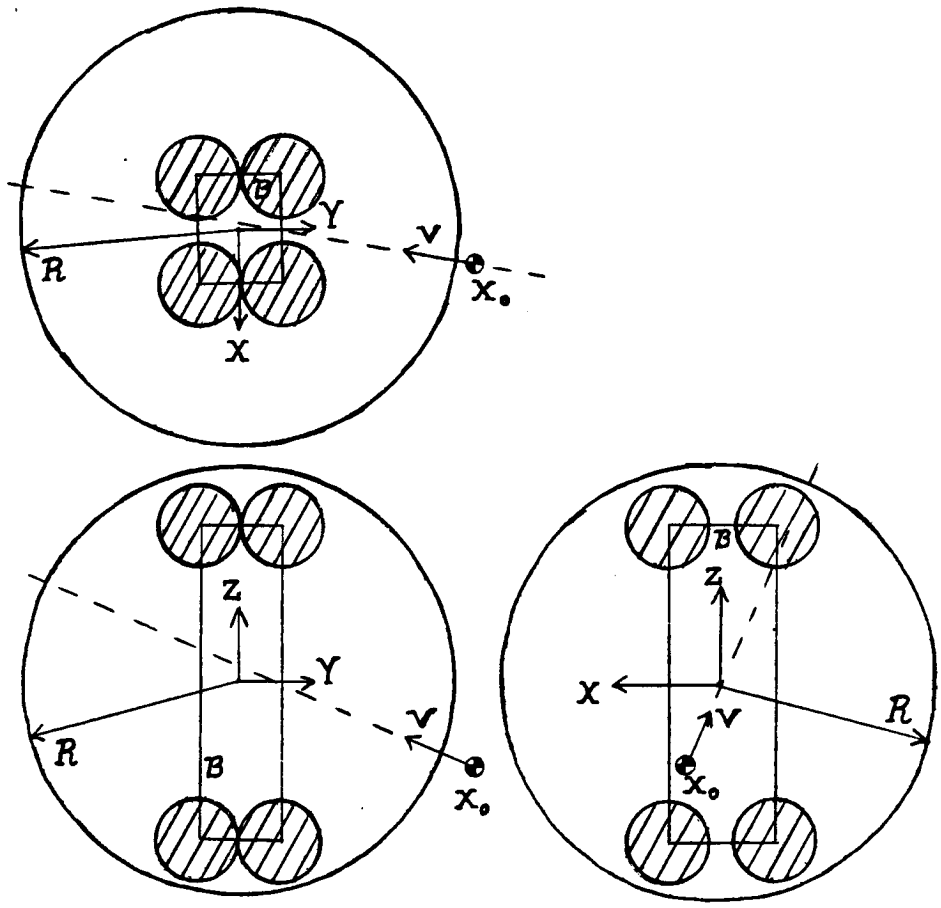
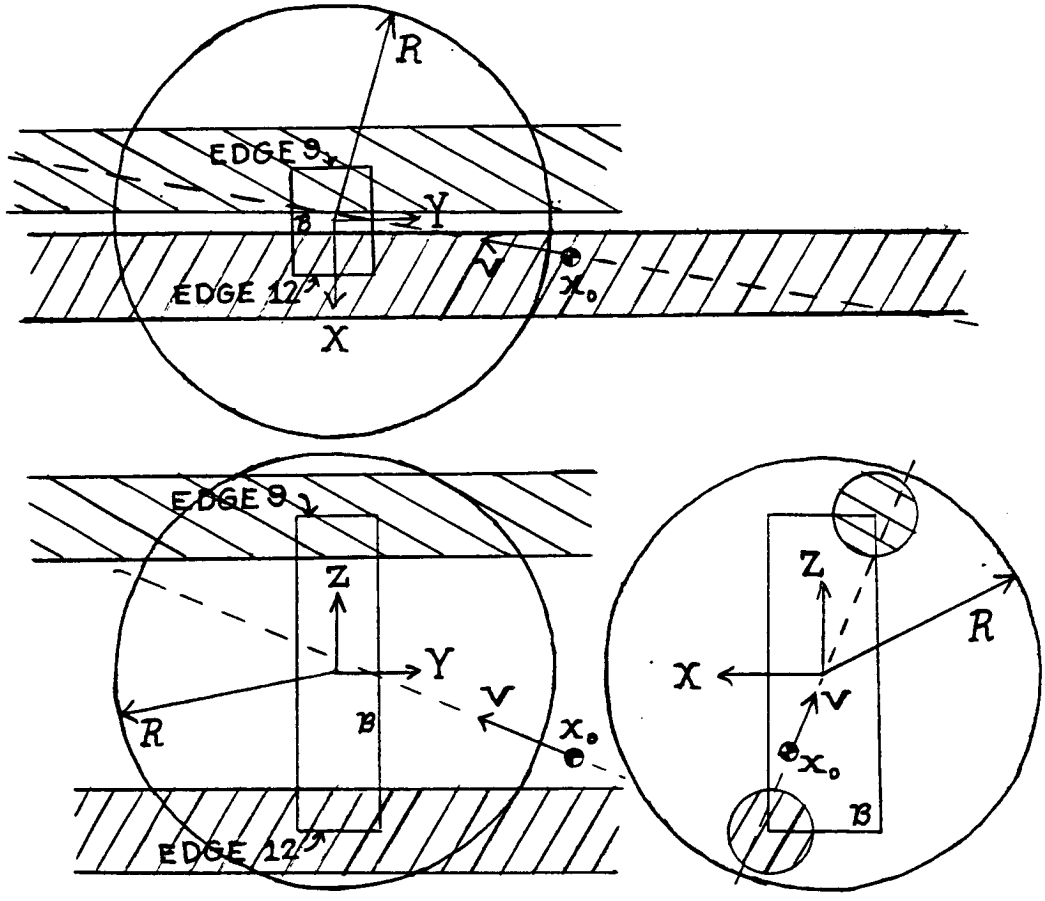


Figure 6.10: The type (b) C-function $f^b_{6,3}$ in the interval $[50.0, 80.0]$



The spheres representing the physical constraint regions for the type (a) C-functions

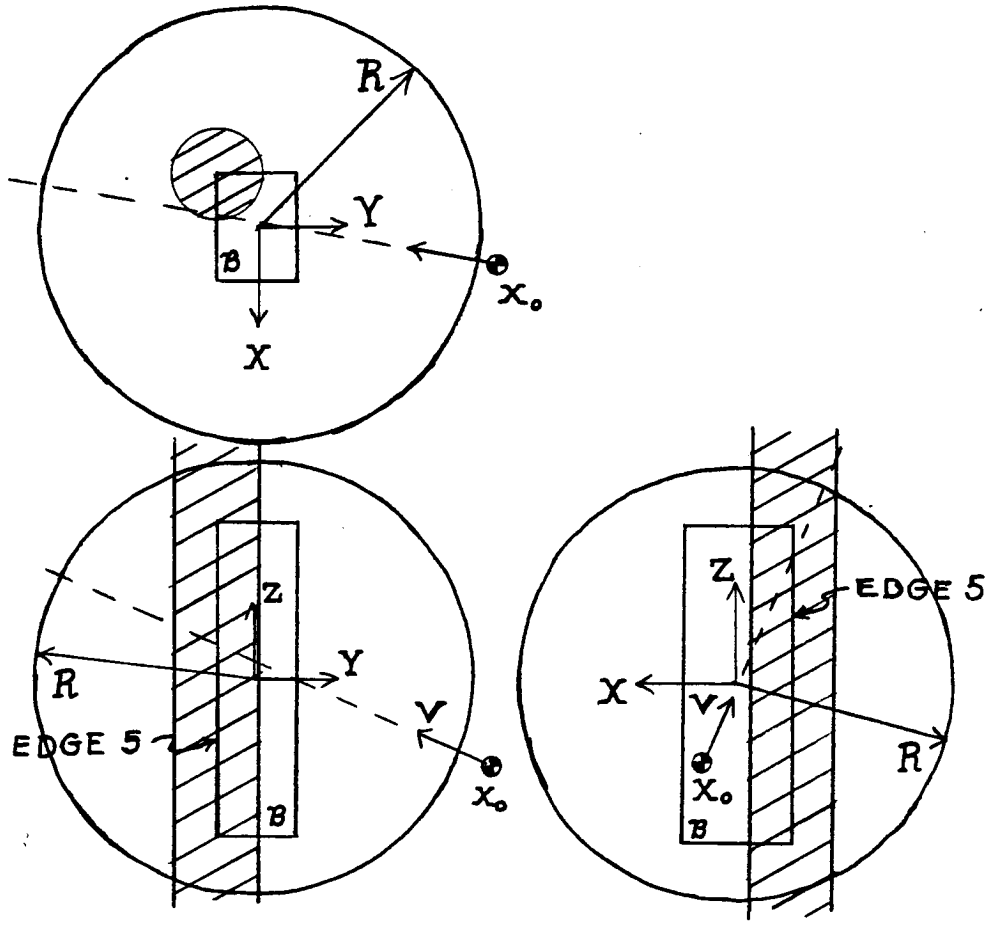
Figure 6.11: The physical constraint regions for the type (a) C-functions



The tube representing the physical constraint region for the type (c) C-functions involving edge 9 of $\mathcal{B}$

The tube representing the physical constraint region for the type (c) C-functions involving edge 12 of $\mathcal{B}$

Figure 6.12: The physical constraint regions for the type (c) C-functions involving edges 9 and 12 of $\mathcal{B}$



The tube representing the physical constraint region for the type (c) C-functions involving edge 5 of $\mathcal{B}$

Figure 6.13: The physical constraint region for the type (c) C-functions involving edge 5 of $\mathcal{B}$

around edge 5 of $\mathcal{B}$ inside the interval where the spheres enclosing $\mathcal{A}$ and $\mathcal{B}$ overlap, but this interval is no longer the current search interval. The new search interval has been updated to reflect the type (b) collision found for $f^b_{5,3}$. The interval where C^a lies inside the tube around edge 5 of $\mathcal{B}$ occurs after this collision between $\mathcal{A}$ and $\mathcal{B}$. Any zeros of the type (c) C-functions involving edge 5 of $\mathcal{B}$ that may satisfy the third collision condition occur in an interval after the first contact between $\mathcal{A}$ and $\mathcal{B}$.

6.4 Limitations of the Algorithm Due to Numerical Inaccuracies

For the example presented in the previous section, no problems were encountered in the execution of the collision detection algorithm due to limitations on the accuracy of the computer's calculations. However, there are two special types of collisions where numerical errors can potentially cause a problem. This section discusses the reasons for the problems encountered in these two special cases.

Because the computer cannot perform its computations exactly, the algorithm incorporates a tolerance on the values that it treats as zero. This tolerance was set to 10^{-6} for the single precision calculations performed by the version of the algorithm implemented for this thesis. Values of the C-functions or ACFs whose absolute values were less than this tolerance were considered to be zero by the collision detection algorithm. The tolerance is also used to test the convergence of the iteration for a zero of a C-function using the secant method.

The first type of collision where the choice of the tolerance for the zero value may cause problems occurs when more than one C-function is zero at the time of first contact. Examples of this type of collision are a face of the moving object touching a face of the obstacle or an edge of either polyhedron touching a face of the other polyhedron. In these cases, the algorithm will find the time of first contact as a zero of several of the C-functions. It will then check the third collision condition by computing the values of the other applicable C-functions at the computed value for the zero when it examines each of these C-functions. A small error in the computed value of the zero for one of the C-functions that equals zero at first contact can cause the values of the other C-functions to have small, but nonzero, values when

computed by the algorithm. That is, the values of the other C-functions that are exactly equal to zero at the time of first contact may have absolute value greater than the tolerance because of accumulated numerical errors in the computer's calculations. This can cause the algorithm to conclude that the third collision condition is not satisfied and, hence, that the computed zero of one of these C-functions does not represent a true collision.

The inaccuracies in checking the third collision condition when there are multiple C-functions that are zero at the time of first contact will not necessarily cause the algorithm to "miss" that collision. The algorithm will only miss the collision if the numerical errors are such that it concludes that the third collision condition is not satisfied at the computed values of the zero for *all* of these C-functions. This is unlikely because different errors will be made in the calculation of the zero for each of these C-functions. If any one of the computed values is slightly larger than the true value of the zero, then the values of the other C-functions at this time will probably be such that the third collision condition is satisfied. This is because values of time slightly larger than the true time of first contact represent configurations that are inside of the configuration space obstacle.

The second case of a collision where numerical problems can arise is when the zero that represents the first contact is a relative maximum or minimum of one of the C-functions. This corresponds to a "glancing blow" contact where some feature of the moving object just touches a feature of the obstacle at a single point and then moves away from the obstacle. Again, small errors in the calculation of the points where the first derivative of the C-function is zero can cause the computer to find a value of the C-function at its relative maximum or minimum that is outside of the tolerance on the zero value. The algorithm will conclude that this point is not a zero of the C-function and will fail to recognize it as a collision. (Of course, it is also possible that the value of the C-function really is small but nonzero at its relative maximum or minimum. In this case, the relative maximum or minimum represents a near miss and not a true collision of the two polyhedra.)

Chapter 7

Examples

The first three sections of this chapter give the results of applying the collision detection algorithm described in Chapter 6 to several example cases. Each of these sections presents three examples of collisions for a different combination of moving object and obstacle. The three examples illustrate a type (a), a type (b), and a type (c) contact at collision for each combination. The moving object and the obstacle are both represented by single convex polyhedra for the examples presented in the first two sections. For the examples in the third section, the moving object and obstacle are both represented by nonconvex polyhedra resulting from the union of two nonoverlapping convex polyhedra.

The first three sections are organized in a similar manner.[1] First, the shapes of the moving object $\mathcal{A}$ and the obstacle $\mathcal{B}$ are described. A figure is given showing the polyhedra representing $\mathcal{A}$ and $\mathcal{B}$. The dimensions of the polyhedra and the numbering of their features are also indicated in this figure. The radius, r_a, of the sphere enclosing $\mathcal{A}$ along with the location of the center, $\mathbf{x}^b$, and the radius, r_b, of the sphere enclosing $\mathcal{B}$ are specified. These three quantities have been computed following the procedure described in section 4.3.

Next, the quantities describing the initial configuration and the motion of $\mathcal{A}$ are listed in a table. This includes the initial position vector $\mathbf{x}_o$, the linear velocity vector $\mathbf{v}$, and the angular velocity vector $\vec{\omega}$. All three of these vectors are expressed in reference coordinates. The initial orientation of $\mathcal{A}$ is specified as a set of yaw, pitch, and roll angles.

The four quantities $\|\mathbf{x}_o - \mathbf{x}^b\|$, $\|\mathbf{v}\|$, $\omega = \|\vec{\omega}\|$, and $r_a\omega/\|\mathbf{v}\|$ are also given in this table. The first of these quantities, $\|\mathbf{x}_o - \mathbf{x}^b\|$, measures the initial distance between the centers of the spheres enclosing $\mathcal{A}$ and $\mathcal{B}$. It is used by the static spherical approximation procedure to determine if the initial configuration lies in free space. For all of the examples presented in the first three sections, the value of $\|\mathbf{x}_o - \mathbf{x}^b\|$ is greater than $R = r_a + r_b$.

[1]The author apologizes for the repetitive nature of the first three sections caused by the similar presentation.

Therefore, the intial configurations are all in free space.

The magnitudes of $\mathbf{v}$ and $\vec{\omega}$ are then used to compute $r_a\omega/\|\mathbf{v}\|$. The value of $r_a\omega/\|\mathbf{v}\|$ is a comparison of the rates of translation and rotation. When $r_a\omega/\|\mathbf{v}\|$ is large, the angular speed is high compared to the linear speed. In this case, the C-functions will typically be more oscillatory in the interval of interest. They can have several relative maxima and minima and consequently may have many zeros. The zero search is more complicated for these cases. Conversely, when $r_a\omega/\|\mathbf{v}\|$ is small, the angular speed is low compared to the linear speed. The C-functions do not change as rapidly in the interval of interest. Many of the C-functions will have only a single zero in the search interval. In these cases, the zero search procedure is relatively simple.

The table is followed by discussions of the examples of a type (a), a type (b), and a type (c) contact at collision, respectively. The initial configuration of $\mathcal{A}$ relative to $\mathcal{B}$ and the directions of the velocity vector and the rotation axis are briefly described in the text. This description is augmented by a figure that shows the initial configuration of $\mathcal{A}$ and $\mathcal{B}$. The directions of the velocity vector and the rotation axis are also indicated in this figure. Then, the time of first contact, T_c, predicted by the collision detection algorithm is given and the features of $\mathcal{A}$ and $\mathcal{B}$ that touch at T_c are specified. The number of the cycle of ωt that includes ωT_c is given along with the value of ωT_c modulo 360°. This gives some idea of the relative change in orientation compared to the change in position that has occurred before the collision. Finally, a figure is included that shows the approximate configuration of $\mathcal{A}$ and $\mathcal{B}$ at T_c.

For the examples in the first three sections, the type of contact expected to occur at collision can be predicted from the choice of the initial configuration and the direction of the linear and angular velocity vectors. The fourth section of this chapter gives three examples of "unexpected" contacts at collision, one for each of the combinations of moving object and obstacle used in the first three sections. For these examples, the actual type of contact at collision predicted by the algorithm is not the contact that might be expected based on the initial configuration and the direction of the linear and angular velocity vectors. The quantities describing the initial configuration and the motion of $\mathcal{A}$ for these examples are listed in tables similar to the tables for the examples in the first three sections. The expected contacts and the type and time of first contact, T_c predicted

by the collision detection algorithm are briefly discussed for each example. Figures are included showing the initial configuration of $\mathcal{A}$ and $\mathcal{B}$ and their approximate configuration at the time of first contact.

The fifth and final section of this chapter gives a comparison of the time required to execute the final algorithm with the time required to execute a similar algorithm having a simpler zero search procedure. Three sets of examples for different moving objects and obstacles are discussed. Some aspects of the collision detection problem that influence the execution time for the algorithm are identified.

7.1 Rectangular Box Examples

For the cases presented in this section, $\mathcal{A}$ and $\mathcal{B}$ are represented by the two rectangular boxes introduced in section 3.3.1. These same boxes were used for the detailed example presented in the last section of Chapter 6. The sizes and numbering of the features for these two boxes are shown in Figure 3.8. The rectangular box representing the obstacle $\mathcal{B}$ is much larger than the box representing the moving object $\mathcal{A}$. The location of C^b, the center of the sphere enclosing $\mathcal{B}$, is $\mathbf{x}^b = \mathbf{0}$ and its radius is $r_b = 52$. The radius of the sphere enclosing $\mathcal{A}$ is $r_a = 13$.

Table 7.1 lists the information for the initial configuration and motion of $\mathcal{A}$ for the three rectangular box examples.

Example of a Type (a) Contact at Collision For the first example, the initial position of $\mathcal{A}$ is such that it is located above and behind $\mathcal{B}$ as shown in Figure 7.1. The initial orientation of $\mathcal{A}$ is such that its body axes are parallel to the reference coordinate axes. The directions of the velocity vector and the rotation axis for this example are also shown in Figure 7.1. The velocity vector is directed toward vertex 1 of $\mathcal{B}$. This vertex is the vertex of $\mathcal{B}$ closest to $\mathcal{A}$ at the initial configuration. The rotation axis passes through faces 3 and 4 of $\mathcal{A}$ and points into the $-X_a$, $-Y_a$, $+Z_a$ octant of the body coordinate system for $\mathcal{A}$.

The collision detection algorithm predicts that the first contact between $\mathcal{A}$ and $\mathcal{B}$ will take place at $T_c = 45.67$. This value of T_c occurs in the seventh cycle of ωt when ωT_c modulo 360° is 351.85°. The collision at T_c represents a type (a) contact between face 2 of $\mathcal{A}$ and vertex 1 of $\mathcal{B}$. The approximate

Initial Position of C^a	
Contact at Collision	$\mathbf{x}_o$
Type (a)	$[\ -32 \quad +25 \quad +100\ ]^T$
Type (b)	$[\ +10 \quad +72 \quad -25\ ]^T$
Type (c)	$[\ +50 \quad -37.5 \quad -30\ ]^T$

(Constant) Linear Velocity Vector of $\mathcal{A}$	
Contact at Collision	$\mathbf{v}$
Type (a)	$[\ +0.28603370 \quad -0.23240238 \quad -0.92960952\ ]^T$
Type (b)	$[\ -0.15205718 \quad -0.91234311 \quad +0.38014296\ ]^T$
Type (c)	$[\ -0.65357429 \quad +0.49018072 \quad +0.57668320\ ]^T$

Initial Orientation of $\mathcal{A}$			
Contact at Collision	Yaw (°)	Pitch (°)	Roll (°)
Type (a)	0	0	0
Type (b)	0	0	0
Type (c)	-36.869898	0	0

(Constant) Angular Velocity Vector of $\mathcal{A}$	
Contact at Collision	$\vec{\omega}(°/t)$
Type (a)	$[\ -6.3508528 \quad -31.754265 \quad +44.455971\ ]^T$
Type (b)	$[\ -6.3508528 \quad -31.754265 \quad +44.455971\ ]^T$
Type (c)	$[\ +30 \quad +40 \quad 0\ ]^T$

Contact at Collision	$\|\mathbf{x}_o - \mathbf{x}^b\|$	$\|\mathbf{v}\|$	$\omega = \|\vec{\omega}\|\ (°/t)$	$r_a\omega/\|\mathbf{v}\|$
Type (a)	107.93	1	55	12.48
Type (b)	76.87	1	55	12.48
Type (c)	69.33	1	50	11.34

Table 7.1: The inputs for the rectangular box examples

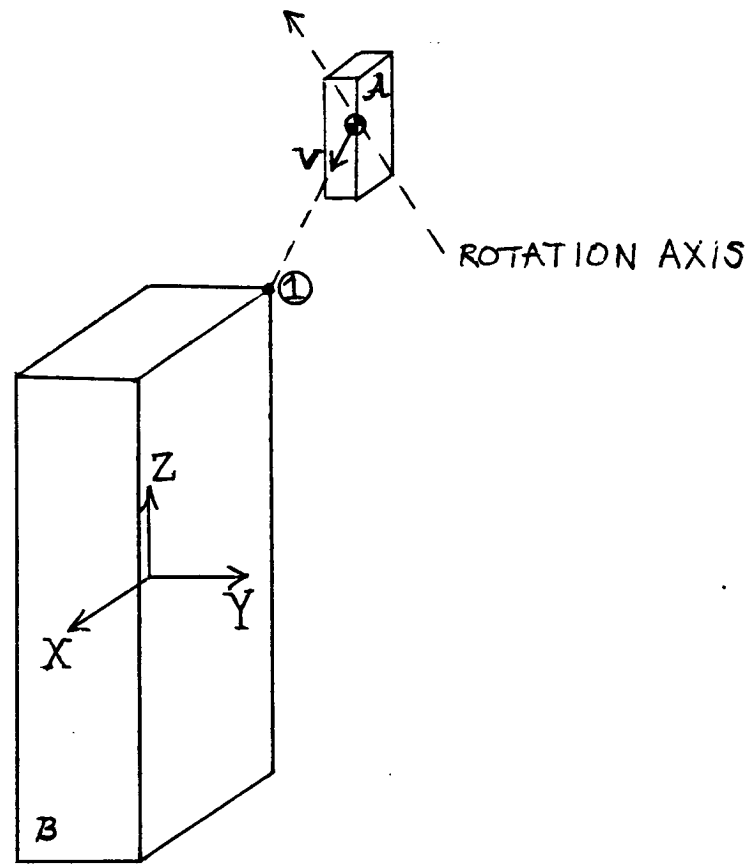


Figure 7.1: The initial configuration of the boxes for the type (a) contact example

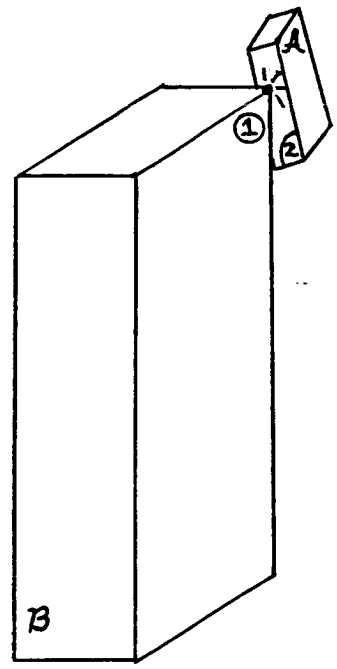


Figure 7.2: The configuration of the boxes at the time of the type (a) contact

configuration of $\mathcal{A}$ and $\mathcal{B}$ at $T_c = 45.67$ is shown in Figure 7.2.

Example of a Type (b) Contact at Collision The second example for the two rectangular boxes is the example discussed in detail in section 6.3. The initial configuration of $\mathcal{A}$ and $\mathcal{B}$ for this example is shown in Figure 6.3. Recall that the collision detection algorithm predicts the time of the first contact between $\mathcal{A}$ and $\mathcal{B}$ to be $T_c = 54.496$. At this time, a type (b) contact occurs between vertex 5 of $\mathcal{A}$ and face 3 of $\mathcal{B}$. The value of T_c lies in the eighth cycle of ωt when the value of ωT_c modulo 360° is 117.28°. Figure 6.4 illustrates the configuration of $\mathcal{A}$ and $\mathcal{B}$ at T_c.

Example of a Type (c) Contact at Collision Figure 7.3 shows the initial configuration of the two boxes for the third example. The initial position of $\mathcal{A}$ is such that C^a lies in the plane passing through edges 1 and 7 of $\mathcal{B}$. The initial orientation of $\mathcal{A}$ is such that faces 1 and 2 of $\mathcal{A}$ are perpendicular to the plane passing through edges 1 and 7 of $\mathcal{B}$, while faces 3 and 4 of $\mathcal{A}$ are parallel to this plane. This is most easily seen in the view of the initial configuration of $\mathcal{A}$ and $\mathcal{B}$ in the X-Y plane shown in

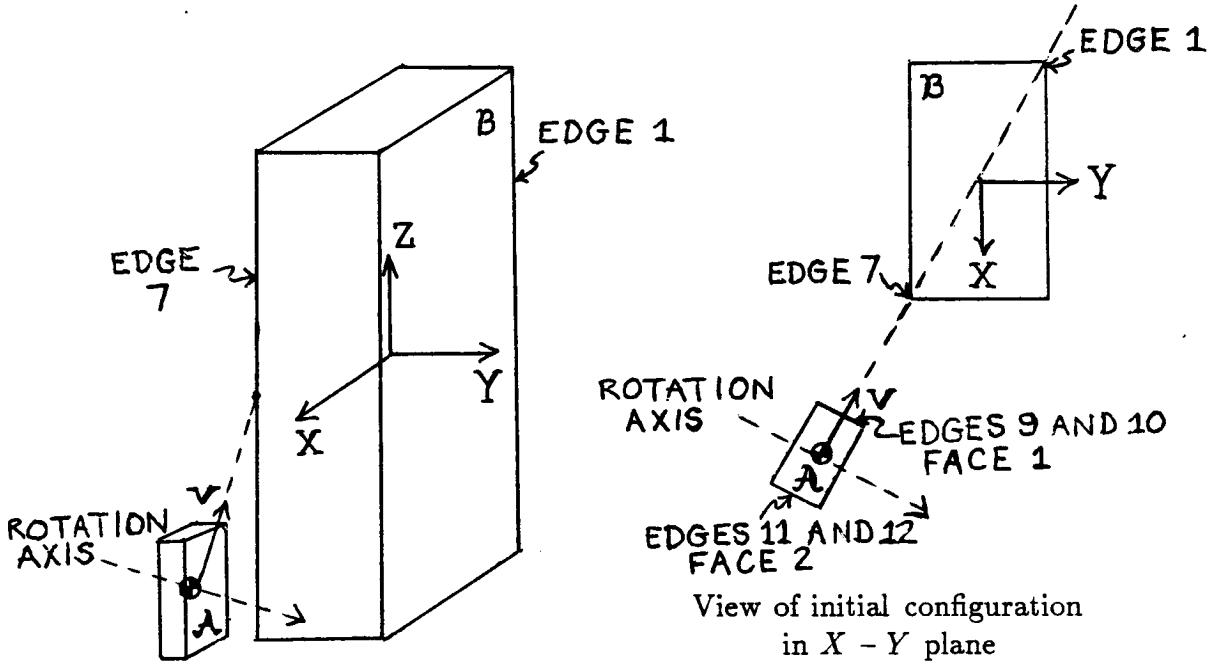


Figure 7.3: The initial configuration of the boxes for the type (c) contact example

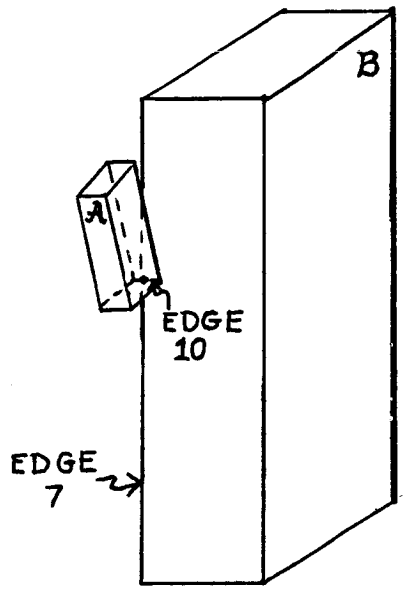


Figure 7.4: The configuration of the boxes at the time of the type (c) contact

Figure 7.3.

As is indicated in Figure 7.3, the velocity vector is directed toward the midpoint of edge 7 of $\mathcal{B}$. This edge is the edge of $\mathcal{B}$ closest to $\mathcal{A}$ at the initial configuration. The axis of rotation is chosen to coincide with the $+Y_a$ axis so that the rotational motion of $\mathcal{A}$ is pure pitch. For this trajectory of $\mathcal{A}$, the first contact is expected to occur when one of edges 9, 10, 11, or 12 of $\mathcal{A}$ touches edge 7 of $\mathcal{B}$.

The collision detection algorithm predicts that the first contact between $\mathcal{A}$ and $\mathcal{B}$ will take place at $T_c = 37.117$. At this time, edge 10 of $\mathcal{A}$ touches edge 7 of $\mathcal{B}$, a type (c) contact. The time of first contact occurs in the fifth cycle of ωt when the value of ωT_c modulo 360° is 55.85°. The approximate configuration of $\mathcal{A}$ and $\mathcal{B}$ at the time of collision is shown in Figure 7.4.

7.2 Octahedron and Pyramid Examples

The polyhedra representing $\mathcal{A}$ and $\mathcal{B}$ in the last section were of the same rectangular shape but had different sizes. For the examples presented in this section, $\mathcal{A}$ and $\mathcal{B}$ are represented by two convex polyhedra having different

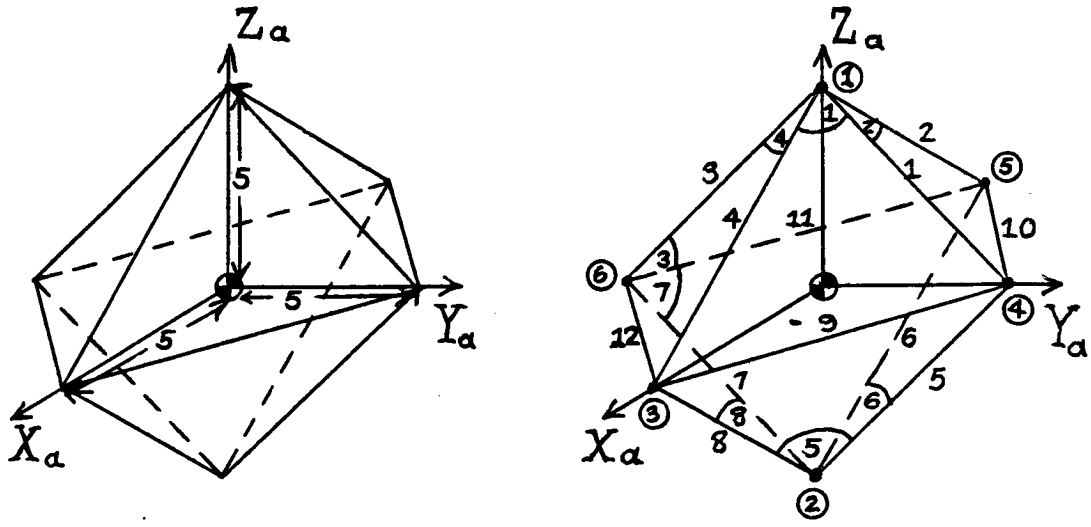
shapes as well as different sizes. The moving object $\mathcal{A}$ is represented by an octahedron while the obstacle $\mathcal{B}$ is represented by a pyramid. The sizes and numbering of the features for the octahedron and the pyramid are shown in Figure 7.5. The center of rotation of $\mathcal{A}$ is located at the geometric center of the octahedron. The radius of the sphere enclosing this octahedron is $r_a = 5$. The origin of the reference coordinate system is located in the center of the base of the pyramid. The location of the center of the sphere enclosing $\mathcal{B}$ is $\mathbf{x}^b = [\ 0 \quad 0 \quad 5\ ]^T$ and its radius is $r_b = 11.18$.

The inputs describing the initial configuration and the motion of $\mathcal{A}$ for the three examples are listed in Table 7.2.

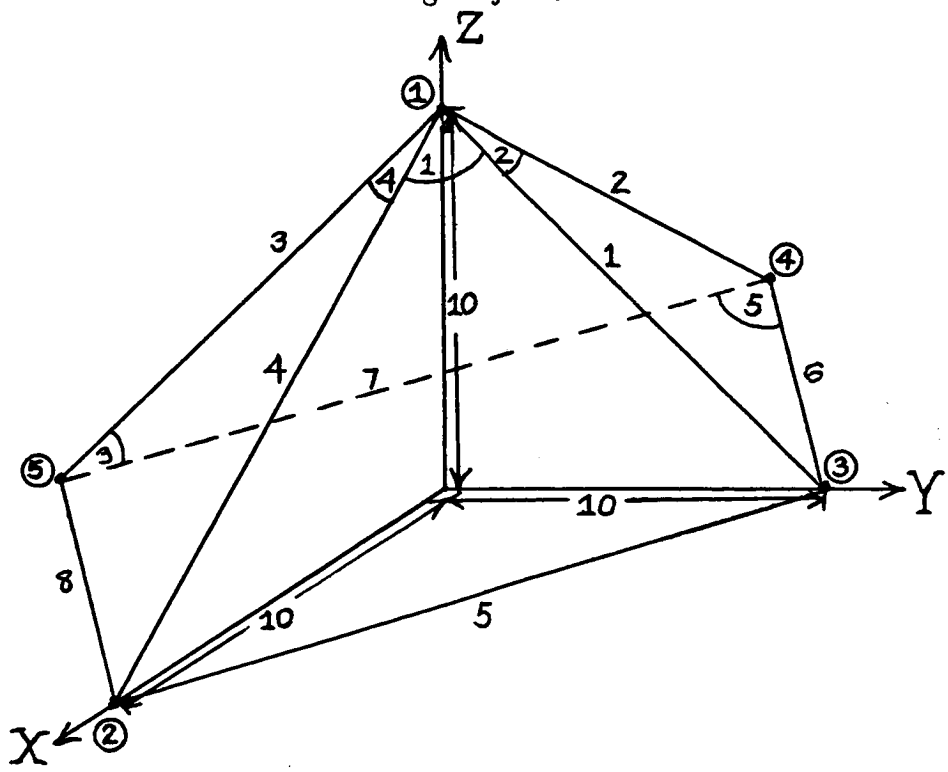
Example of a Type (a) Contact at Collision $\mathcal{A}$ is initially located below and to the right of $\mathcal{B}$ for this example. The initial orientation of $\mathcal{A}$ is such that its body axes are parallel to the reference axes. The velocity vector is directed toward a point slightly in front of vertex 3 of $\mathcal{B}$. This vertex is the vertex of $\mathcal{B}$ closest to $\mathcal{A}$ at $t = 0$. The rotation axis points into the $-X_a$, $-Y_a$, $+Z_a$ octant and pierces faces 3 and 5 of $\mathcal{A}$. The initial configuration of $\mathcal{A}$ relative to $\mathcal{B}$ along with the velocity vector and the rotation axis chosen for this example are shown in Figure 7.6.

The collision detection algorithm predicts that the first contact between $\mathcal{A}$ and $\mathcal{B}$ will take place at $T_c = 64.60$. At this time, face 4 of $\mathcal{A}$ touches vertex 3 of $\mathcal{B}$, a type (a) contact. The value of ωT_c modulo 360° is 313.00° and the actual value of ωT_c is contained in the ninth cycle of ωt. The approximate configuration of $\mathcal{A}$ and $\mathcal{B}$ at the time of collision is shown in Figure 7.7.

Example of a Type (b) Contact at Collision For this second example, $\mathcal{A}$ is initially located almost directly below the base of the pyramid representing $\mathcal{B}$, as shown in Figure 7.8. The center of rotation of $\mathcal{A}$ initially lies in the $+X$, $+Y$, $-Z$ octant of the reference coordinate system, but the magnitudes of the X and Y components of $\mathbf{x}_o$ are small compared to the magnitude of its Z component. The velocity vector is directed toward the center of face 5 of $\mathcal{B}$, which is the origin of the reference coordinate system. The initial orientation of $\mathcal{A}$ is such that the X_a axis is parallel to the X axis while the Y_a and Z_a axes are rotated 30° from the Y and Z axes. The rotation axis passes through faces 1 and 7 of $\mathcal{A}$, pointing into the $+X_a$,



Moving Object $\mathcal{A}$



Obstacle $\mathcal{B}$

Face i Edge i Vertex i

Figure 7.5: The octahedron representing $\mathcal{A}$ and the pyramid representing $\mathcal{B}$

Initial Position of C^a	
Contact at Collision	$\mathbf{x}_o$
Type (a)	$[\ +10 \quad +72 \quad -25\ ]^T$
Type (b)	$[\ +5 \quad +3 \quad -40\ ]^T$
Type (c)	$[\ +30 \quad +40 \quad -20\ ]^T$

(Constant) Linear Velocity Vector of $\mathcal{A}$	
Contact at Collision	$\mathbf{v}$
Type (a)	$[\ -0.15205718 \quad -0.91234311 \quad +0.38014296\ ]^T$
Type (b)	$[\ -0.61846337 \quad -0.37107802 \quad +4.9477070\ ]^T$
Type (c)	$[\ -2.785430 \quad -3.7139068 \quad +1.8569534\ ]^T$

Initial Orientation of $\mathcal{A}$			
Contact at Collision	Yaw (°)	Pitch (°)	Roll (°)
Type (a)	0	0	0
Type (b)	0	0	30
Type (c)	0	0	30

(Constant) Angular Velocity Vector of $\mathcal{A}$	
Contact at Collision	$\vec{\omega}(°/t)$
Type (a)	$[\ -6.3508528 \quad -31.754265 \quad +44.455971\ ]^T$
Type (b)	$[\ +0.81649658 \quad +1.00597 \quad +1.52360\ ]^T$
Type (c)	$[\ +0.81649658 \quad +1.00597 \quad +1.52360\ ]^T$

Contact at Collision	$\lVert \mathbf{x}_o - \mathbf{x}^b \rVert$	$\lVert \mathbf{v} \rVert$	$\omega = \lVert \vec{\omega} \rVert\ (°/t)$	$r_a \omega / \lVert \mathbf{v} \rVert$
Type (a)	78.64	1	55	4.80
Type (b)	45.38	5	2	0.0349
Type (c)	55.90	5	2	0.0349

Table 7.2: The inputs for the examples using the octahedron and the pyramid

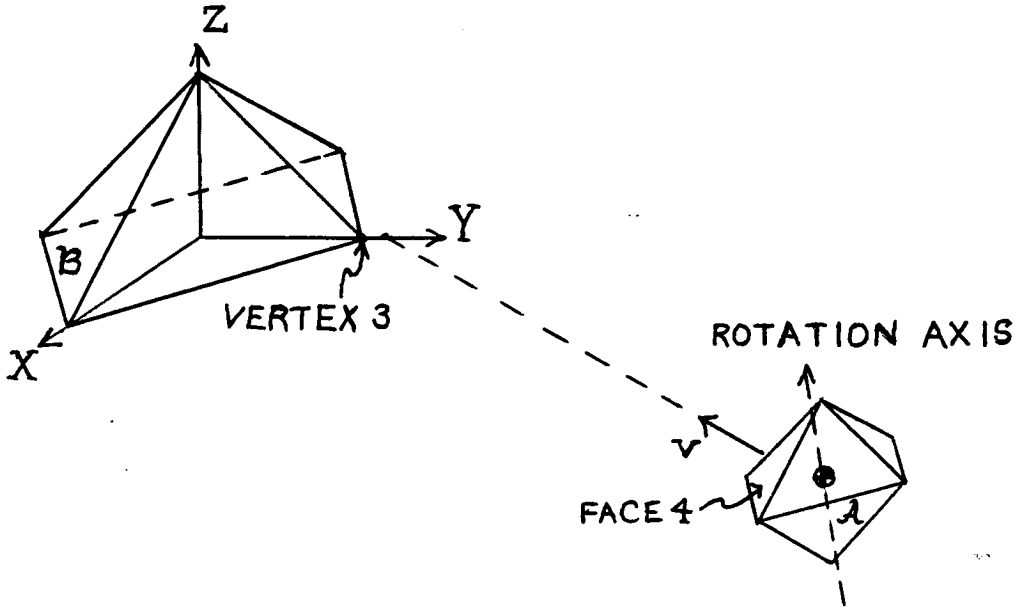


Figure 7.6: The initial configuration of the octahedron and the pyramid for the type (a) contact example

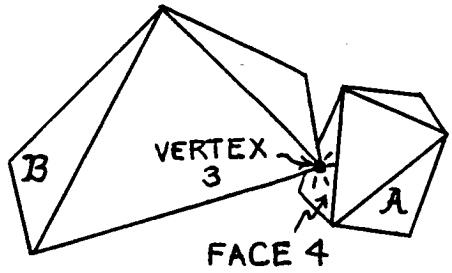


Figure 7.7: The configuration of the octahedron and the pyramid at the time of the type (a) contact

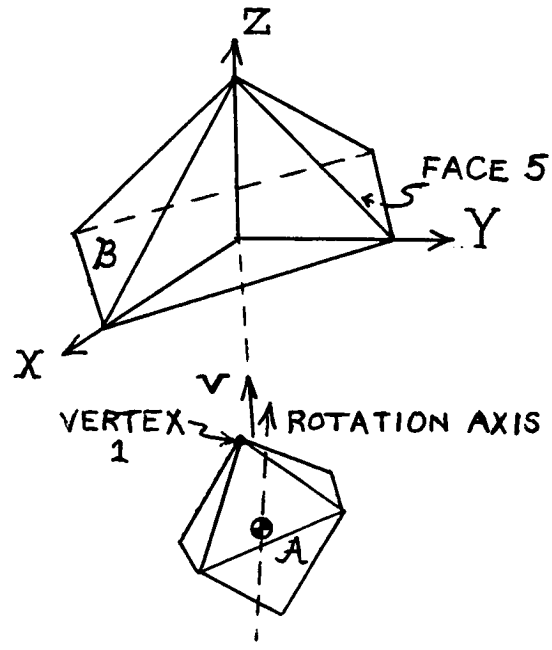


Figure 7.8: The initial configuration of the octahedron and the pyramid for the type (b) contact example

$+Y_a$, $+Z_a$ octant. However, the angle between the Z_a axis and the rotation axis is relatively small.

For the given initial position and velocity vector, this means that the first contact between $\mathcal{A}$ and $\mathcal{B}$ will probably occur when vertex 1 of $\mathcal{A}$ touches face 5 of $\mathcal{B}$, which is the base of the pyramid. This is confirmed by the collision detection algorithm. The predicted time of first contact between $\mathcal{A}$ and $\mathcal{B}$ is $T_c = 7.30$. At this time, a type (b) contact occurs between vertex 1 of $\mathcal{A}$ and face 5 of $\mathcal{B}$. The time of first contact occurs in the first cycle of ωt because of the low rotation rate. The value of ωT_c modulo 360° is 14.60°. The approximate configuration of $\mathcal{A}$ and $\mathcal{B}$ at the time of collision is shown in Figure 7.9.

Example of a Type (c) Contact at Collision Figure 7.10 shows the initial configuration of $\mathcal{A}$ and $\mathcal{B}$ for this example. The initial orientation of $\mathcal{A}$ and the rotation axis are the same as those chosen for the previous example. The initial position of $\mathcal{A}$ is such that it lies below the base of the pyramid representing $\mathcal{B}$ in the $+X$, $+Y$, $-Z$ octant of the reference coordinate system. However, in this case, the three components of $\mathbf{x}_o$ are closer

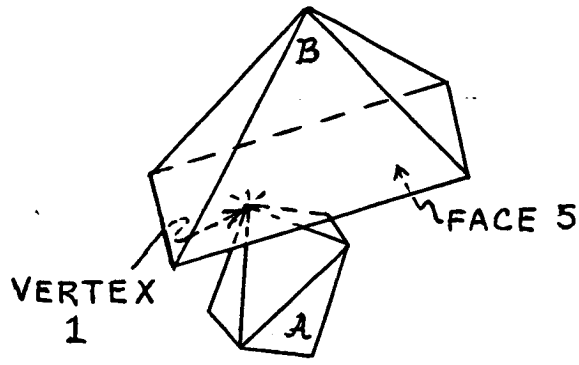


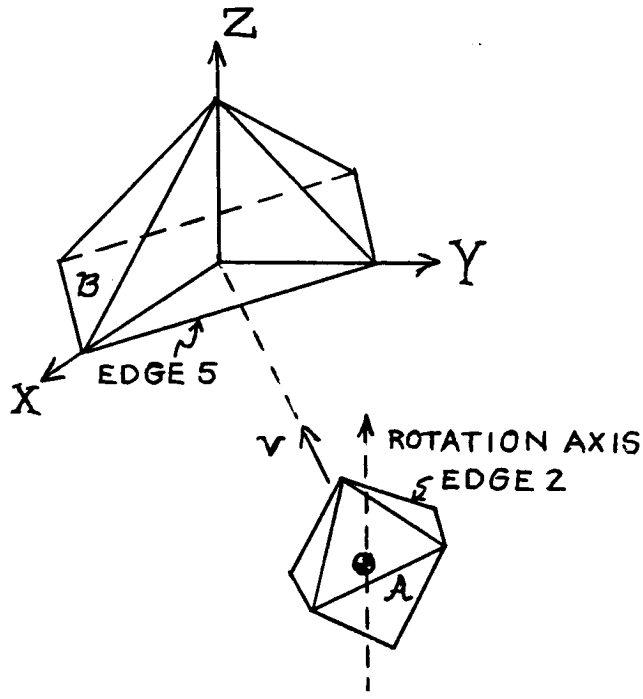
Figure 7.9: The configuration of the octahedron and the pyramid at the time of the type (b) contact

in magnitude than was the case for the previous example. The velocity vector is again directed toward the center of face 5 of $\mathcal{B}$.

The difference in the initial position of $\mathcal{A}$ for this example relative to to the preceding example causes the first contact to be a type (c) collision. The collision detection algorithm predicts that the first contact between $\mathcal{A}$ and $\mathcal{B}$ will take place at $T_c = 8.90$. At this time, edge 2 of $\mathcal{A}$ touches edge 5 of $\mathcal{B}$, which is one of the edges bounding face 5 of $\mathcal{B}$. The time of first contact occurs in the first cycle of ωt when the value of ωT_c modulo 360° is 17.80°. The approximate configuration of $\mathcal{A}$ and $\mathcal{B}$ at the time of collision is shown in Figure 7.11.

7.3 Examples of Two Nonconvex Polyhedra

The moving object and the obstacle were both represented by convex polyhedra for the examples discussed in the two preceding sections. For the examples presented in this section, $\mathcal{A}$ and $\mathcal{B}$ are both represented by *nonconvex* polyhedra. $\mathcal{A}$ consists of two rectangular boxes forming the L-



Compare this figure to Figure 7.8 which shows the initial configuration for the type (b) contact example.

Figure 7.10: The initial configuration of the octahedron and the pyramid for the type (c) contact example

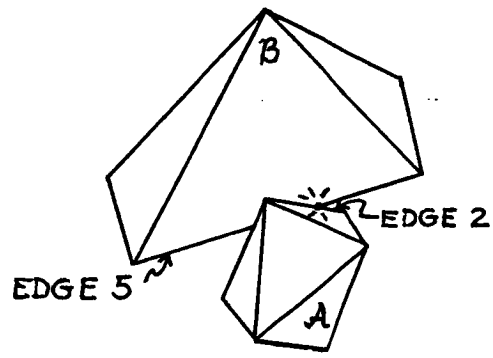


Figure 7.11: The configuration of the octahedron and the pyramid at the time of the type (c) contact

shaped box shown in Figure 7.12. The center of rotation of $\mathcal{A}$, as indicated in the figure, is located at the geometric center of the larger rectangular box. This larger box is denoted $\mathcal{A}_1$ and the smaller box is denoted $\mathcal{A}_2$. The dimensions of the two rectangular boxes and the numbering of their respective features are also indicated in Figure 7.12. The radius of the sphere enclosing $\mathcal{A} \equiv \mathcal{A}_1 \cup \mathcal{A}_2$ is $r_a = 18.68$.

The two convex polyhedra that are joined to form the nonconvex polyhedron representing $\mathcal{B}$ are a rectangular box and a pyramid. The base of the pyramid is connected to the rectangular box in the manner shown in Figure 7.13. The figure also shows the dimensions of the box and the pyramid and the numbering of their features. The origin of the reference coordinate system is located at the geometric center of the rectangular box. The rectangular box is referred to as $\mathcal{B}_1$ and the pyramid is referred to as $\mathcal{B}_2$ so that $\mathcal{B} \equiv \mathcal{B}_1 \cup \mathcal{B}_2$. The center of the sphere enclosing $\mathcal{B}$ is located at $\mathbf{x}^b = [\ 0 \quad 6 \quad 0\ ]^T$ and its radius is $r_b = 53.70$.

The initial configuration of $\mathcal{A}$ and its linear and angular velocity vectors for each of the three examples are listed in Table 7.3.

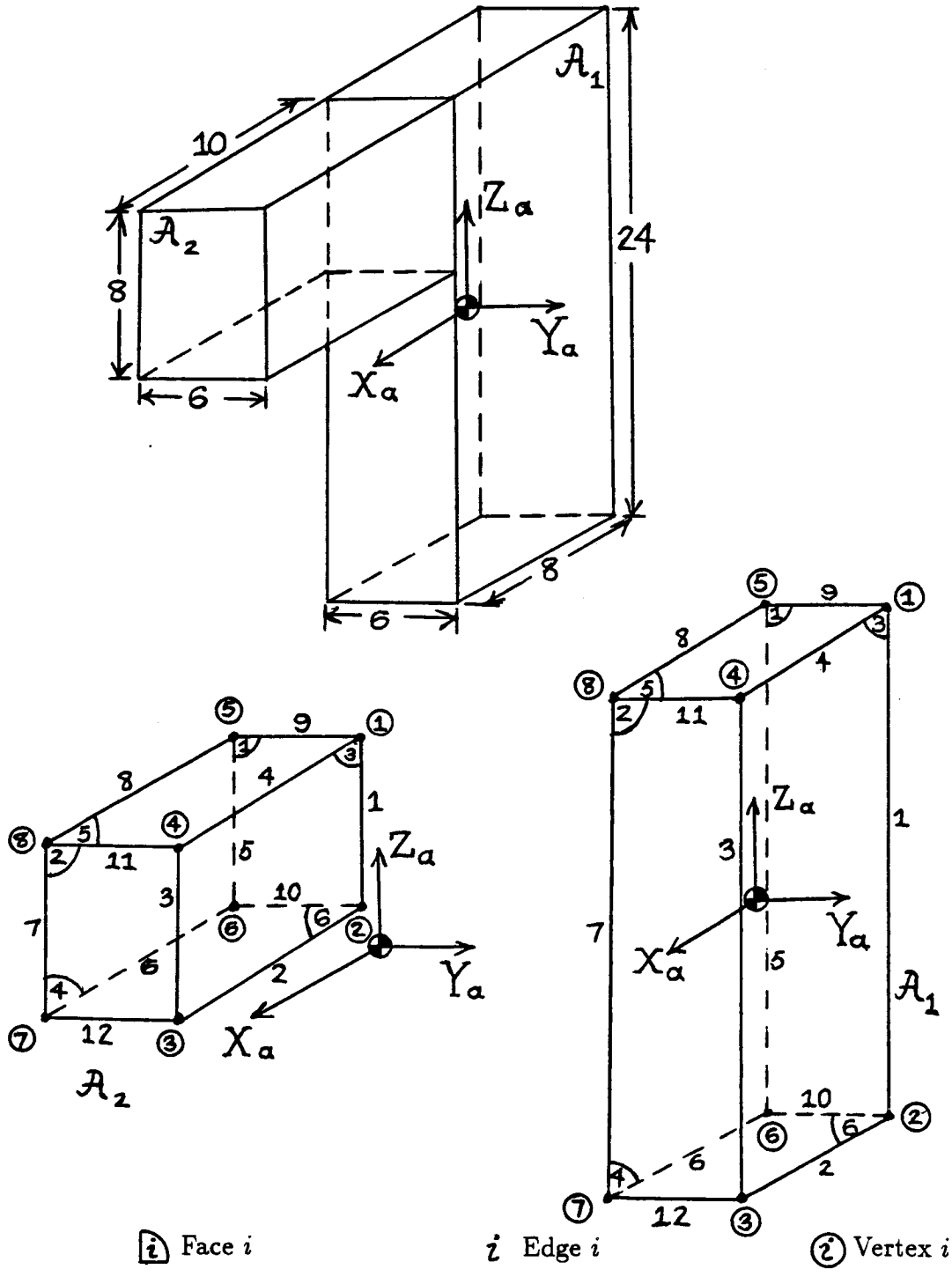


Figure 7.12: $\mathcal{A}$ is the union of two rectangular boxes

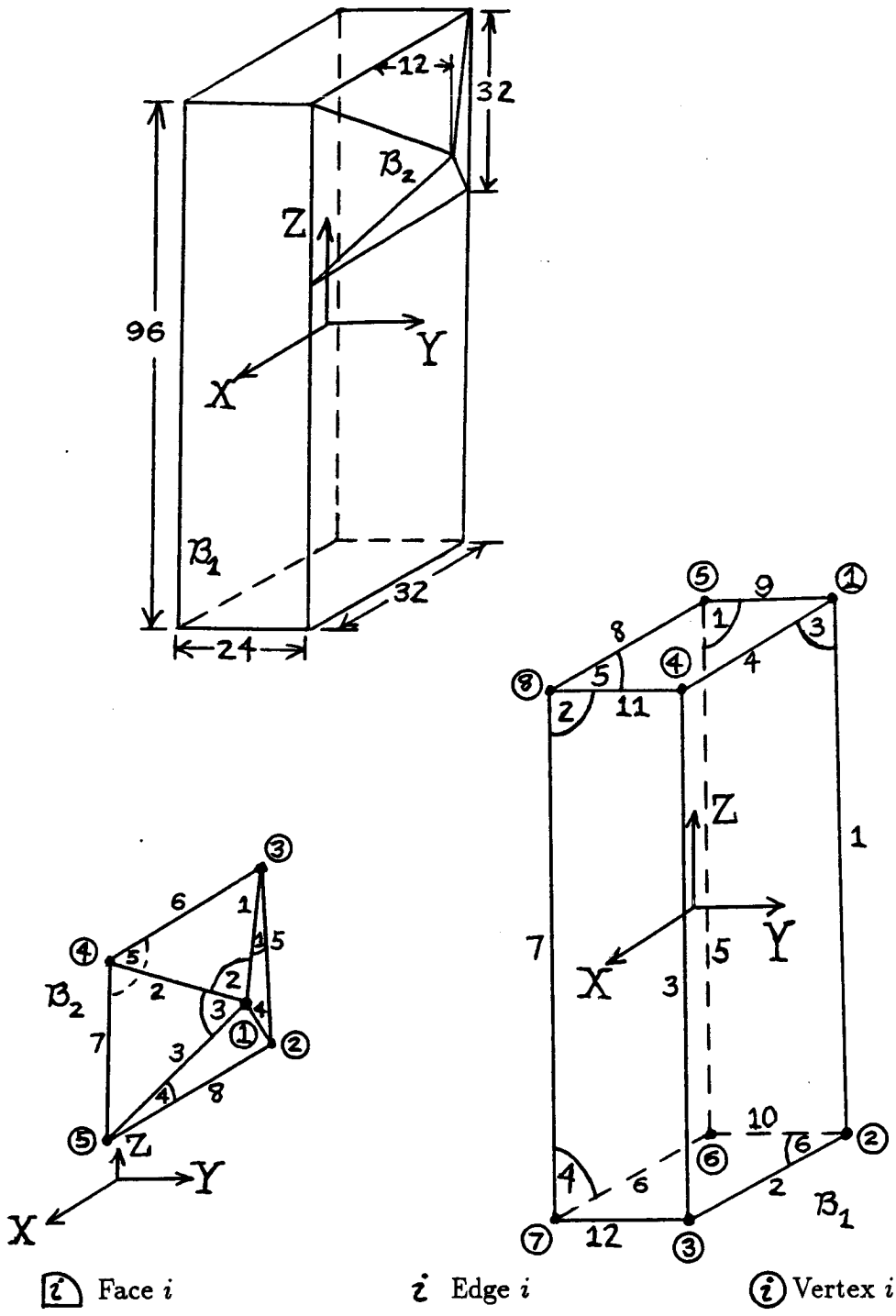


Figure 7.13: $\mathcal{B}$ is the union of a rectangular box and a pyramid

Initial Position of C^a	
Contact at Collision	$\mathbf{x}_o$
Type (a)	$[\ +55 \quad -41.25 \quad +75\ ]^T$
Type (b)	$[\ +10 \quad +56 \quad +75\ ]^T$
Type (c)	$[\ +55 \quad -41.25 \quad -30\ ]^T$

(Constant) Linear Velocity Vector of $\mathcal{A}$	
Contact at Collision	$\mathbf{v}$
Type (a)	$[\ -4.8988316 \quad +3.6741237 \quad -3.3914988\ ]^T$
Type (b)	$[\ -0.18340136 \quad -0.58688434 \quad -0.78862584\ ]^T$
Type (c)	$[\ -8.1759198 \quad +6.1319399 \quad +6.2891691\ ]^T$

Initial Orientation of $\mathcal{A}$			
Contact at Collision	Yaw (°)	Pitch (°)	Roll (°)
Type (a)	-36.869898	0	0
Type (b)	0	0	0
Type (c)	-36.869898	0	0

(Constant) Angular Velocity Vector of $\mathcal{A}$	
Contact at Collision	$\vec{\omega}(°/t)$
Type (a)	$[\ -13.4 \quad -21.2 \quad 0\ ]^T$
Type (b)	$[\ +3.9777864 \quad +2.7844505 \quad +1.1933359\ ]^T$
Type (c)	$[\ -15 \quad -20 \quad 0\ ]^T$

Contact at Collision	$\|\mathbf{x}_o - \mathbf{x}^b\|$	$\|\mathbf{v}\|$	$\omega = \|\vec{\omega}\|\ (°/t)$	$r_a\omega/\|\mathbf{v}\|$
Type (a)	104.32	7	25.08	1.168
Type (b)	90.69	1	5	1.630
Type (c)	78.47	12	25	0.6793

Table 7.3: The inputs for the examples using the two nonconvex polyhedra

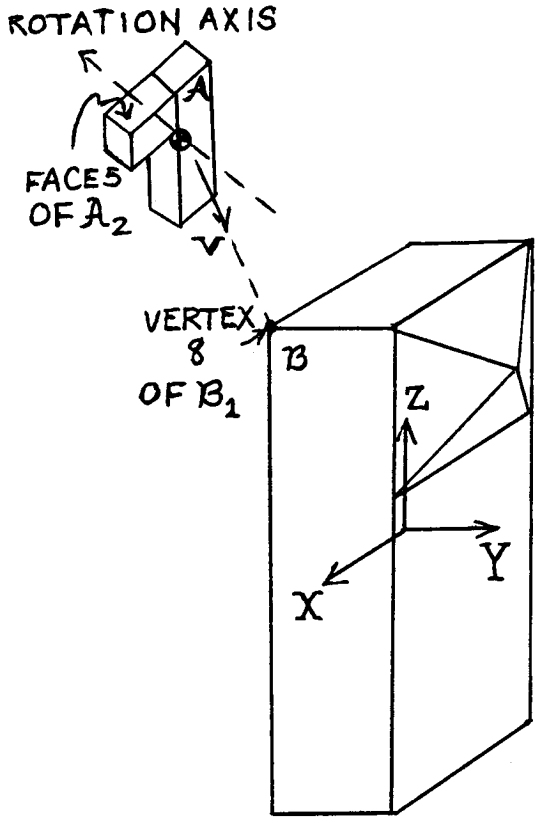
Example of a Type (a) Contact at Collision The initial configuration of $\mathcal{A}$ and $\mathcal{B}$ for this example is depicted in Figure 7.14. In this case, $\mathcal{A}$ is located above $\mathcal{B}$. The initial position of $\mathcal{A}$ is such that C^a lies in the plane passing through edges 7 and 1 of $\mathcal{B}_1$. The initial orientation of $\mathcal{A}$ is such that the Z_a axis is parallel to the Z axis but the X_a and Y_a axes are rotated $-36.87°$ from the X and Y axes. For this orientation, faces 1 and 2 of both $\mathcal{A}_1$ and $\mathcal{A}_2$ are initially perpendicular to the plane passing through edges 1 and 7 of $\mathcal{B}_1$. Faces 3 and 4 of $\mathcal{A}_1$ and $\mathcal{A}_2$ are initially parallel to this plane. This is illustrated in the view of the initial configuration of $\mathcal{A}$ and $\mathcal{B}$ in the X-Y plane included in Figure 7.14.

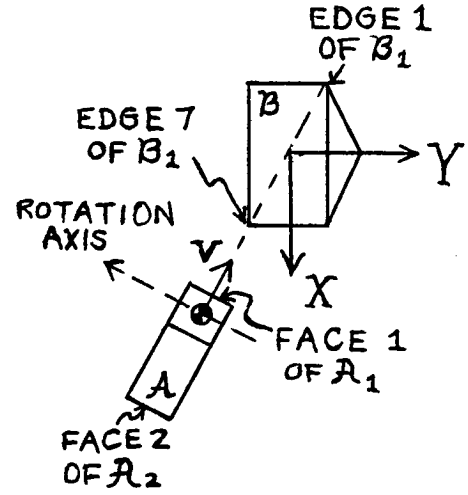
The velocity vector and the rotation axis are also indicated in Figure 7.14. The velocity vector is directed toward vertex 8 of $\mathcal{B}_1$. This is the vertex of $\mathcal{B}$ closest to $\mathcal{A}$ at the initial time. The largest component of the angular velocity vector lies along the $-Y_a$ axis so that the rotational motion of $\mathcal{A}$ is almost entirely negative pitch.

The initial configuration and the trajectory of $\mathcal{A}$ are such that the first contact is most likely to occur between vertex 8 of $\mathcal{B}_1$ and either face 1, 5, or 6 of $\mathcal{A}_1$ or face 2, 5, or 6 of $\mathcal{A}_2$. This is verified by the results of the collision detection algorithm. The first contact between $\mathcal{A}$ and $\mathcal{B}$ is found to be a type (a) contact between face 5 of $\mathcal{A}_2$ and vertex 8 of $\mathcal{B}_1$. This collision occurs in the first cycle of ωt at $T_c = 5.965$. The value of ωT_c modulo 360° is 149.60°. The approximate configuration of $\mathcal{A}$ and $\mathcal{B}$ at T_c is shown in Figure 7.15.

Example of a Type (b) Contact at Collision In this case, $\mathcal{A}$ is initially located above $\mathcal{B}$ as shown in Figure 7.16. The velocity vector is directed toward vertex 1 of $\mathcal{B}_2$, which is the apex of the pyramid. The initial orientation of $\mathcal{A}$ is such that its body axes are parallel to the reference axes. The rotation axis points into the $+X_a$, $+Y_a$, $+Z_a$ octant and passes through faces 3 and 4 of $\mathcal{A}_1$.

The collision detection algorithm finds the time of the first collision between $\mathcal{A}$ and $\mathcal{B}$ to be $T_c = 45.70$. At this time, a type (b) contact occurs between vertex 2 of $\mathcal{A}_1$ and face 2 of $\mathcal{B}_2$. Since the angular speed is relatively low, the time of first contact occurs in the first cycle of ωt. The value of ωT_c modulo 360° is 228.5°. The approximate configuration of $\mathcal{A}$ and $\mathcal{B}$ at the time of collision is shown in Figure 7.17.





View of initial configuration
in $X - Y$ plane

Figure 7.14: The initial configuration of the nonconvex polyhedra for the type (a) contact example

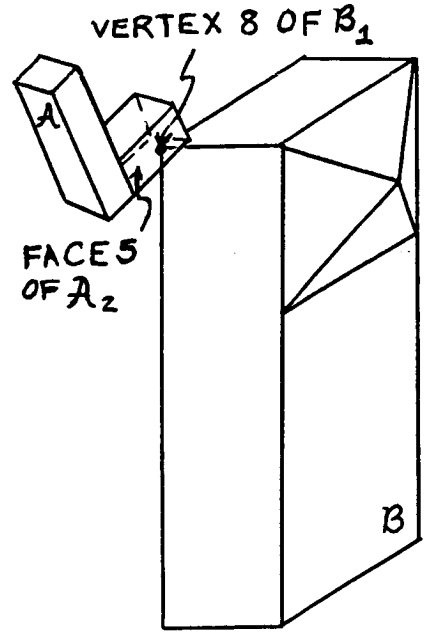


Figure 7.15: The configuration of the nonconvex polyhedra at the time of the type (a) contact

Example of a Type (c) Contact at Collision The initial configuration for this example is similar to that used above for the example of a type (a) contact. The initial position of $\mathcal{A}$ again lies in the plane passing through edges 7 and 1 of $\mathcal{B}_1$. However, C^a now lies initially in the $+X$, $-Y$, $-Z$ octant of the reference coordinate system, below $\mathcal{B}$. The initial orientation of $\mathcal{A}$ is the same as the initial orientation described above for the example of a type (a) contact at collision. The velocity vector is directed toward the midpoint of edge 7 of $\mathcal{B}_1$. This edge is the edge of $\mathcal{B}$ closest to $\mathcal{A}$ at the initial configuration. The axis of rotation is chosen to coincide with the $-Y_a$ axis so that the rotational motion of $\mathcal{A}$ is pure negative pitch. Figure 7.18 shows the initial configuration of $\mathcal{A}$ and $\mathcal{B}$ for this example.

For the chosen trajectory of $\mathcal{A}$, the first contact is expected to occur when one of edges 9, 10, or 12 of $\mathcal{A}_1$ or edges 11 or 12 of $\mathcal{A}_2$ touches edge 7 of $\mathcal{B}_1$. The result of the collision detection algorithm verifies this prediction. The algorithm finds that the first contact between $\mathcal{A}$ and $\mathcal{B}$ will take place at $T_c = 3.595$. At this time, edge 9 of $\mathcal{A}_1$ touches edge 7 of $\mathcal{B}_1$, a type (c) contact. The value of T_c falls in the first cycle of ωt and the value of ωT_c modulo 360° is 89.875°. The approximate configuration of $\mathcal{A}$ and $\mathcal{B}$ at the

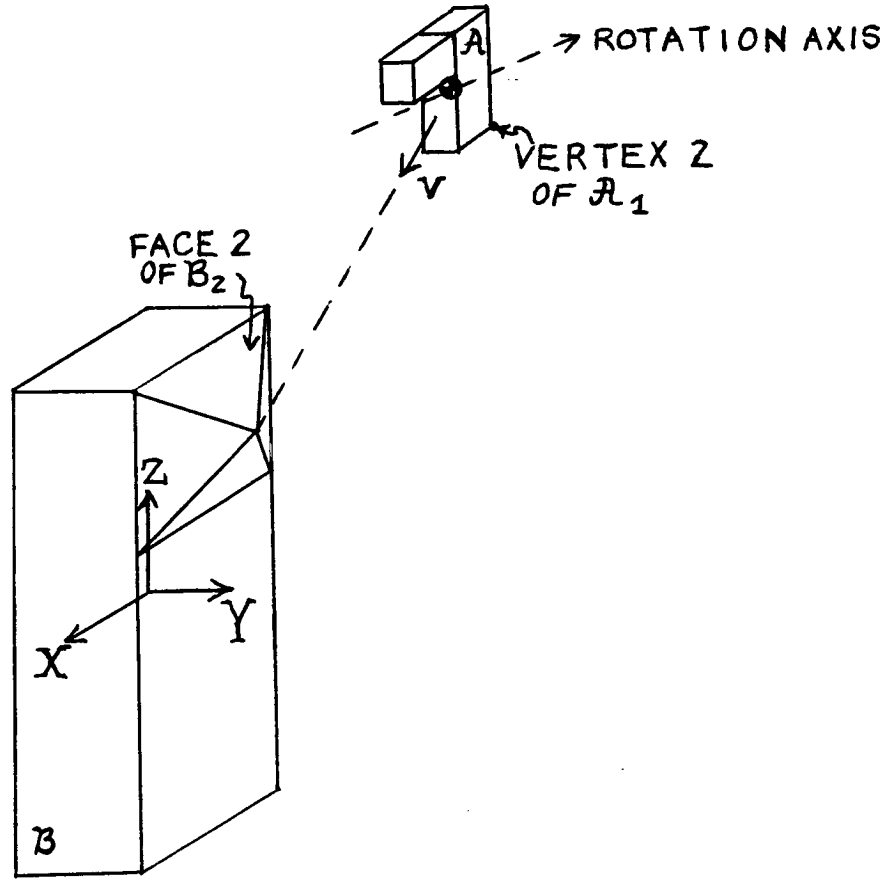


Figure 7.16: The initial configuration of the nonconvex polyhedra for the type (b) contact example

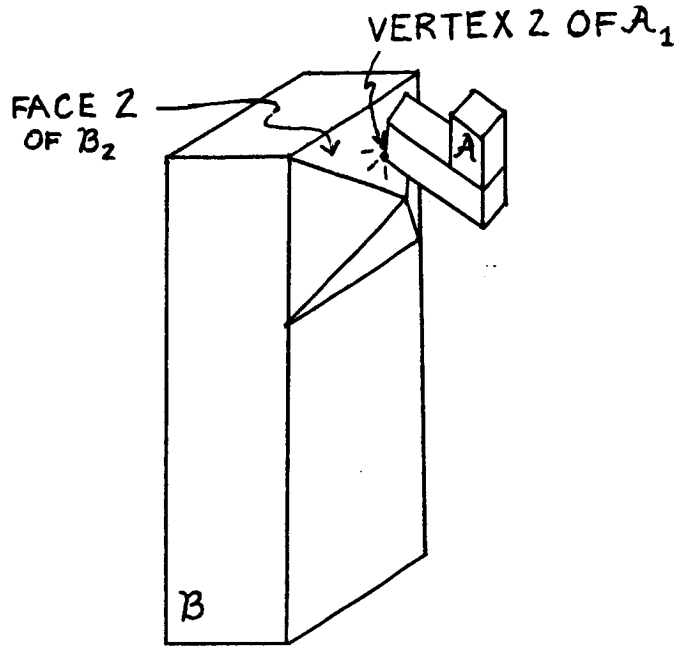


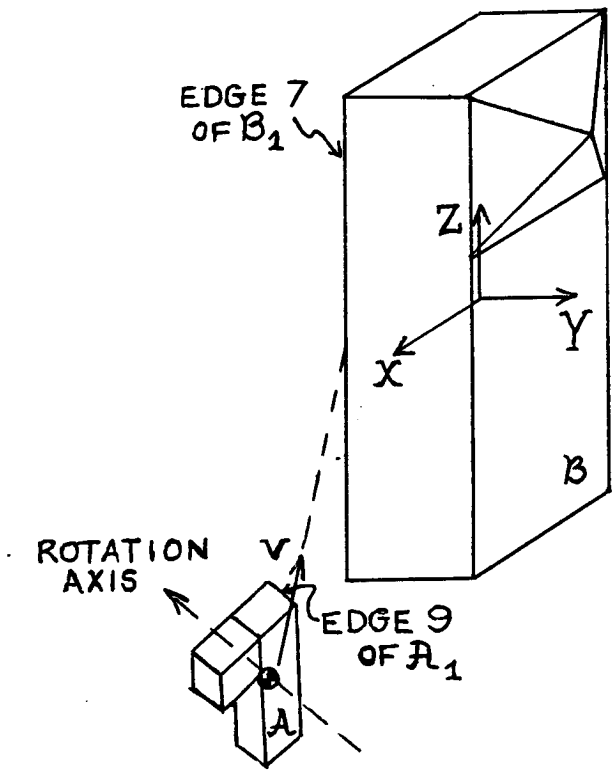
Figure 7.17: The configuration of the nonconvex polyhedra at the time of the type (b) contact

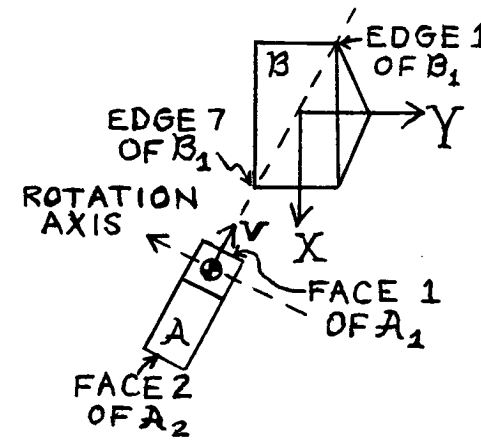
time of collision is shown in Figure 7.19.

7.4 Examples of Unexpected Types of Contact at Collision

The discussion of the examples in the previous sections indicates that, in most cases, the type of contact at collision could be predicted from the choice of the initial configuration and the direction of the linear and angular velocity vectors. This section will present three examples where the actual type of contact at collision as determined by the algorithm is not the contact that might be expected based on the choice of these quantities. The interaction between the rotation and the translation of $\mathcal{A}$ is harder to predict for these examples than for the examples in the first three sections. In these cases, it is difficult to visualize which features of $\mathcal{A}$ might be touching $\mathcal{B}$ without knowing how far into its rotation cycle $\mathcal{A}$ is located at a given position of C^a.

One example is discussed below for each of the combinations of moving





View of initial configuration
in $X - Y$ plane

Figure 7.18: The initial configuration of the nonconvex polyhedra for the type (c) contact example

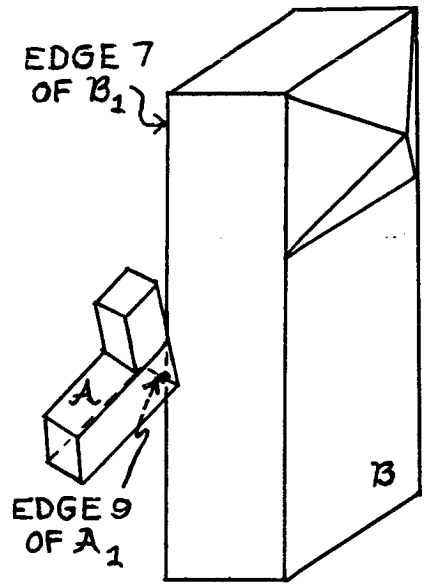


Figure 7.19: The configuration of the nonconvex polyhedra at the time of the type (c) contact

object and obstacle used in sections 7.1, 7.2, and 7.3. The initial position and orientation of $\mathcal{A}$ as well as the linear and angular velocity vectors for these examples are shown in Tables 7.4, 7.5, and 7.6, respectively.

Example of an Unexpected Type (b) Contact at Collision using the Rectangular Boxes The two rectangular boxes used in section 7.1 are also used for the first example of an unexpected type of contact at collision. For this example, the smaller rectangular box representing $\mathcal{A}$ lies in front of the larger box representing $\mathcal{B}$ as shown in Figure 7.20. Because the velocity vector is directed toward the midpoint of edge 7 of $\mathcal{B}$, it is expected that a type (c) contact will occur between this edge of $\mathcal{B}$ and some edge of $\mathcal{A}$. However, the collision detection algorithm predicts that the first contact is a type (b) contact occurring at $T_c = 43.12$ when vertex 7 of $\mathcal{A}$ touches face 2 of $\mathcal{B}$. Face 2 of $\mathcal{B}$ is one of the two faces that meet at edge 7 of $\mathcal{B}$. In this case, the rotation of $\mathcal{A}$ causes vertex 7 to come into contact with face 2 of $\mathcal{B}$ before any of the edges of $\mathcal{A}$ are close enough to touch edge 7 of $\mathcal{B}$. The approximate configuration of $\mathcal{A}$ and $\mathcal{B}$ at the time of collision is shown in Figure 7.21

$\mathbf{x}_o$ – Initial Position of C^a
$[\ +50 \quad -40 \quad -30\ ]^T$

$\mathbf{v}$ – (Constant) Linear Velocity Vector of $\mathcal{A}$
$[\ -0.63799863 \quad +0.52541064 \quad 0.56293997\ ]^T$

Initial Orientation of $\mathcal{A}$		
Yaw (°)	Pitch (°)	Roll (°)
0	0	0

$\vec{\omega}$ – (Constant) Angular Velocity Vector of $\mathcal{A}$
$[\ -6.3508528 \quad -31.754265 \quad +44.455971\ ]^T\ (°/t)$

$\|\mathbf{x}_o - \mathbf{x}^b\|$	$\|\mathbf{v}\|$	$\omega = \|\vec{\omega}\|\ (°/t)$	$r_a\omega/\|\mathbf{v}\|$
70.71	1	55	12.48

Table 7.4: The inputs for the example of an unexpected contact using the rectangular boxes

$\mathbf{x}_o$ – Initial Position of C^a
$[\ +10 \quad +30 \quad -55\ ]^T$

$\mathbf{v}$ – (Constant) Linear Velocity Vector of $\mathcal{A}$
$[\ -2.0211646 \quad -4.0423208 \quad +11.116405\ ]^T$

Initial Orientation of $\mathcal{A}$		
Yaw (°)	Pitch (°)	Roll (°)
0	45	0

$\vec{\omega}$ – (Constant) Angular Velocity Vector of $\mathcal{A}$
$[\ +11.5470 \quad +16.3299 \quad 0\ ]^T\ (°/t)$

$\lVert \mathbf{x}_o - \mathbf{x}^b \rVert$	$\lVert \mathbf{v} \rVert$	$\omega = \lVert \vec{\omega} \rVert\ (°/t)$	$r_a \omega / \lVert \mathbf{v} \rVert$
67.82	12	20	0.1454

Table 7.5: The inputs for the example of an unexpected contact using the octahedron and the pyramid

$\mathbf{x}_o$ – Initial Position of C^a
$[\ -5 \quad +42 \quad +73\ ]^T$

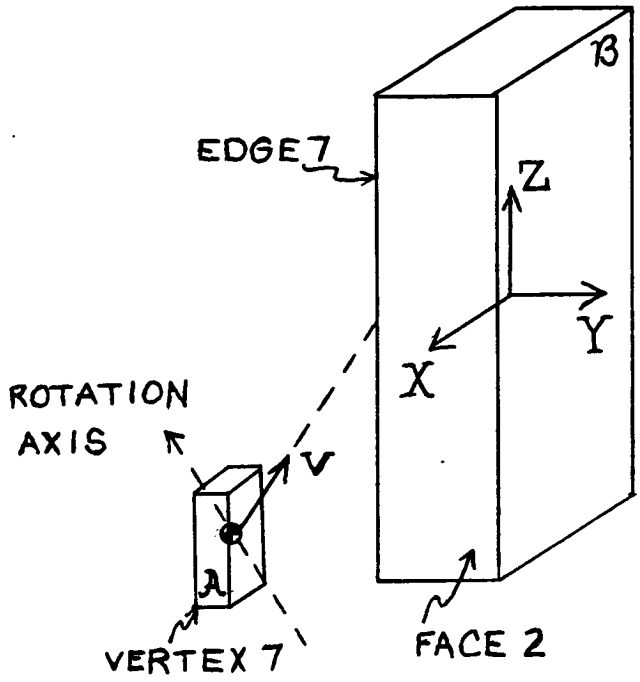
$\mathbf{v}$ – (Constant) Linear Velocity Vector of $\mathcal{A}$
$[\ +0.38100039 \quad -2.2860023 \quad -1.9050019\ ]^T$

Initial Orientation of $\mathcal{A}$		
Yaw (°)	Pitch (°)	Roll (°)
-90	0	0

$\vec{\omega}$ – (Constant) Angular Velocity Vector of $\mathcal{A}$
$[\ +14.9256 \quad 0 \quad +1.49256\ ]^T\ (°/t)$

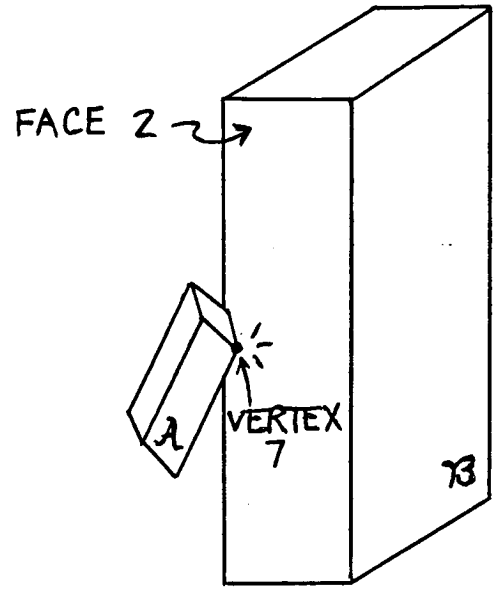
$\lVert \mathbf{x}_o - \mathbf{x}^b \rVert$	$\lVert \mathbf{v} \rVert$	$\omega = \lVert \vec{\omega} \rVert\ (°/t)$	$r_a \omega / \lVert \mathbf{v} \rVert$
81.55	3	15	1.630

Table 7.6: The inputs for the example of an unexpected contact using the two nonconvex polyhedra



A type (c) contact involving edge 7 of $\mathcal{B}$ is expected.

Figure 7.20: The initial configuration of the rectangular boxes for the example of an unexpected type (b) contact

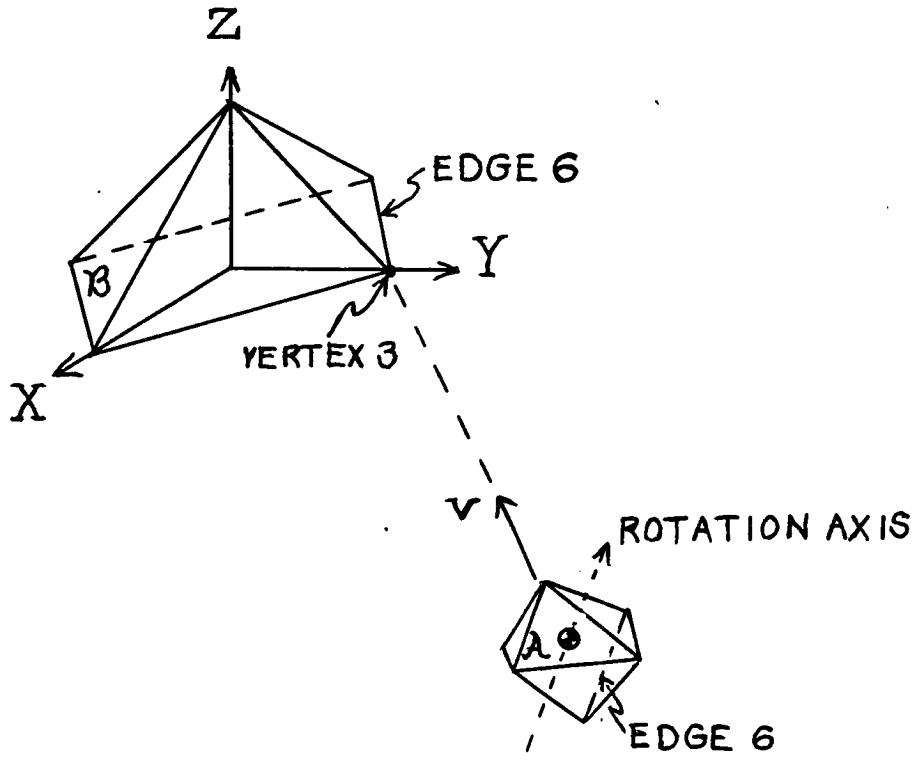


The actual collision is a type (b) contact involving face 2 of $\mathcal{B}$.

Figure 7.21: The configuration of the rectangular boxes at the time of the type (b) contact

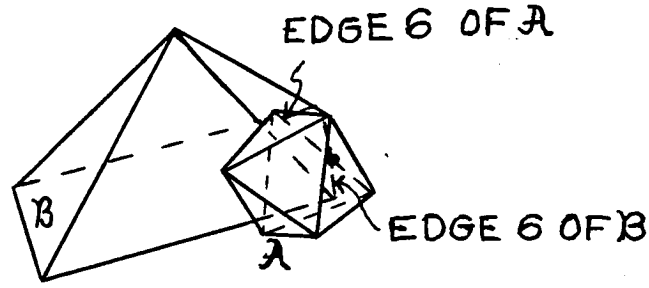
Example of an Unexpected Type (c) Contact at Collision using the Octahedron and the Pyramid The second example of an unexpected contact at collision uses the octahedron and the pyramid introduced in section 7.2. Figure 7.22 shows that the octahedron is initially located below and to the right of the pyramid for this example. This figure also shows that the the velocity vector is chosen to point toward vertex 3 of $\mathcal{B}$. Therefore, a type (a) contact is expected to occur between this vertex of $\mathcal{B}$ and some face of $\mathcal{A}$. In this case, the first contact predicted by the collision detection algorithm is a type (c) contact at $T_c = 4.69$. At this time, the orientation of $\mathcal{A}$ is such that edge 6 of $\mathcal{A}$ touches edge 6 of $\mathcal{B}$. Edge 6 of $\mathcal{B}$ is one of the edges that meet at vertex 3 of $\mathcal{B}$. The approximate configuration of $\mathcal{A}$ and $\mathcal{B}$ at the time of collision is shown in Figure 7.23

Example of an Unexpected Type (c) Contact at Collision using the Nonconvex Polyhedra The nonconvex polyhedra of section 7.3 are used for the final example of an unexpected type of contact at collision. The initial configuration of the two nonconvex polyhedra for this example is



A type (a) contact involving vertex 3 of $\mathcal{B}$ is expected.

Figure 7.22: The initial configuration of the octahedron and the pyramid for the example of an unexpected type (c) contact



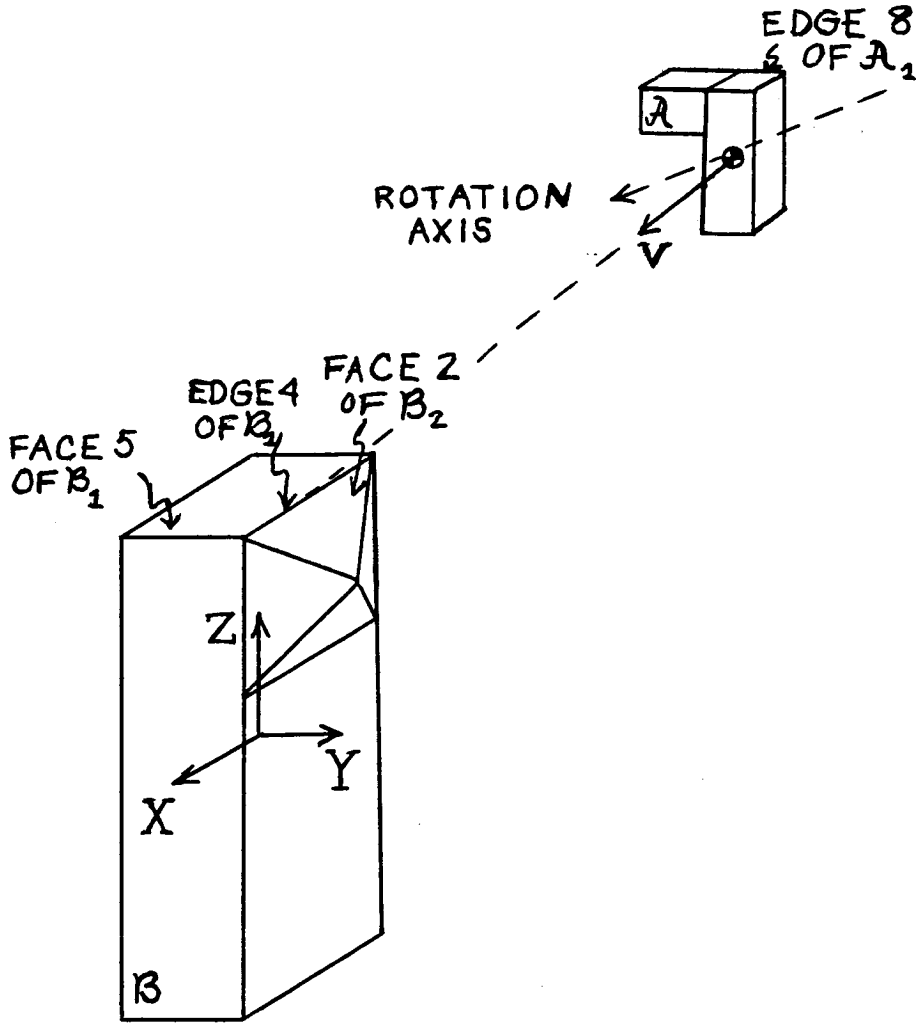
The actual collision is a type (c) contact involving edge 6 of $\mathcal{B}$.

Figure 7.23: The configuration of the octahedron and the pyramid at the time of the type (c) contact

shown in Figure 7.24. The L-shaped box representing $\mathcal{A}$ is initially located above and to the right of $\mathcal{B}$. The velocity vector points toward the midpoint of edge 4 of $\mathcal{B}_1$ which coincides with edge 6 of $\mathcal{B}_2$. $\mathcal{A}$ is initially rotated 90° about the $-Z_a$ axis and the rotation axis has its largest component along Y_a. The expected collision is therefore a type (b) contact between face 5 of $\mathcal{B}_1$ or face 2 of $\mathcal{B}_2$ and some vertex of $\mathcal{A}$. Instead, the algorithm predicts a type (c) contact at $T_c = 8.963$. The orientation of $\mathcal{A}$ at T_c is such that edge 8 of $\mathcal{A}_1$ touches edge 4 of $\mathcal{B}_1$. Figure 7.25 shows the approximate configuration of $\mathcal{A}$ and $\mathcal{B}$ at the time of the type (c) contact.

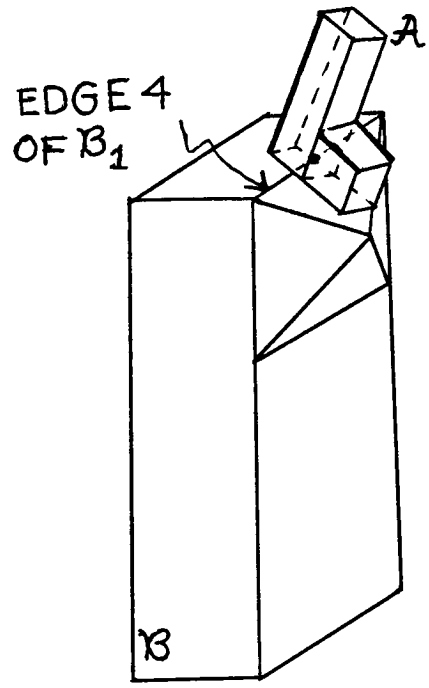
7.5 Summary of Examples

The examples discussed in sections 7.1 through 7.4 illustrate the versatility of the new collision detection algorithm. First, these examples have shown that the algorithm is capable of handling cases where the moving object and obstacle are either convex or nonconvex polyehdra. Also, it can successfully



A type (b) contact involving face 5 of $\mathcal{B}_1$ or face 2 of $\mathcal{B}_2$ is expected.

Figure 7.24: The initial configuration of the nonconvex polyhedra for the example of an unexpected type (c) contact



The actual collision is a type (c) contact involving edge 4 of $\mathcal{B}_1$.

Figure 7.25: The configuration of the nonconvex polyhedra at the time of the type (c) contact

detect collisions involving all three types of contacts. Finally, the algorithm can handle cases of both high and low angular speeds relative to the linear speed. This is confirmed by the range of values for $r_a\omega/\|\mathbf{v}\|$ from 0.0349 to 12.48 for the examples presented in these four sections. The hierarchy described in Chapter 6 allows the new algorithm to solve all these different cases with a minimum amount of searching for zeros of the C-functions.

The discussion of the examples in the preceding sections has emphasized that the new algorithm correctly solves the collision detection problem in a broad range of cases. Implementation issues such as the time needed to execute the algorithm for these examples have not been discussed. The next section will consider the issue of execution time for several different sets of examples. In particular, the execution time required for the algorithm described in Chapter 6 will be compared with the time required for a similar algorithm that uses a simpler zero search procedure.

7.6 Comparison of Two Zero Search Methods

From the description in Chapter 6, it is clear that the most complicated procedure used by the algorithm is the configuration space dynamic collision dectection procedure. The most complex part of this procedure is the examination of the C-functions defined by features of the convex polyhedra making up $\mathcal{A}$ and $\mathcal{B}$. This procedure typically requires execution of one of the numerical zero search procedures described in sections 5.1.5.2 and 5.2.2.3 for many of these C-functions. These numerical zero search procedures are also quite complex.

This section will consider a different zero search method that is much simpler than those described in Chapter 5. A collision detection algorithm is introduced below that uses this simpler zero search method. A comparision is made between the time required to execute the algorithm of Chapter 6 and the time required to execute this algorithm.

This section is organized as follows. First, in section 7.6.1, the simpler zero search method is described and compared to the search procedures of Chapter 5. Then, the sets of examples to be used in the comparison of the two search methods are presented in section 7.6.2. The resulting execution times are tabulated for each set of examples in section 7.6.3. This section also contains some comments on different aspects of the collision detection problem that influence these times for each version of the algorithm.

7.6.1 A Simpler Zero Search Method

Recall the basic outline of a procedure for finding the zeros of a function given at the beginning of section 5.1.5.2. This procedure finds the zeros in a specified bounded interval of time by sampling the value of the function at certain test points in the interval. The array of test points in a search interval selected by the procedures described in Chapter 5 consists of the beginning and end of the search interval and any points in the interval where the first or second derivative of the C-function is zero. For this reason, the zero search procedures described in Chapter 5, and used by the algorithm described in Chapter 6, will be collectively referred to as the "derivative" zero search method.

The points where the first or second derivative of the C-function equals zero are the points where the curvature changes sign or where the function has a relative maximum or minimum. These points are found by a numerical search procedure for a C-function with the form of cases 3, 4, or 5 of class Π described in section 5.2. As was explained in Chapter 5, this choice of test points insures that only a single zero of the C-function is located between two consecutive test points where the value of the C-function changes sign. Therefore, no zeros of the C-function in the given interval are overlooked by the derivative zero search method.

The simpler zero search method differs from the derivative zero search method only in the selection of the test points in a search interval. The new test points are selected from a set of points separated by a constant spacing of $\Delta\omega t$ in a cycle of ωt. This scheme is feasible because the requirement that the C-function be applicable for a valid collision restricts the final bounded zero search intervals to portions of a single cycle of ωt. This simpler zero search method will be called the "fixed $\Delta\omega t$" method.

The number of points in a cycle of ωt for the fixed $\Delta\omega t$ zero search method can be varied by changing the value of $\Delta\omega t$. These points divide a single cycle of ωt into $360°/(\Delta\omega t°)$ intervals of equal length. For example, if $\Delta\omega t = 72°$, then the fixed set consists of the four points $\omega t = \{72°, 144°, 216°, 288°\}$ and the cycle of ωt is divided into $360°/72° = 5$ equal intervals.

The test points chosen for a given search interval are the bounds of the interval and any points from the fixed set which lie in the interval. The values of t in the interval that correspond to values of ωt in the fixed set can easily be found by comparing the value of ωt at the bounds of the interval modulo 2π and the values of ωt in the fixed set. As an example, suppose that $\Delta\omega t = 72°$ and the bounds of a search interval modulo 360° are 15° and 160°, respectively. Then, the test points for that interval are $\omega t = \{15°, 72°, 144°, 160°\}$. These values can be converted back to values of t by adding the appropriate multiple of 360° and dividing the result by ω.

The steps used to find the bounded intervals for the zero search of a given C-function are the same for both versions of the collision detection algorithm. The examination of a C-function involving the checks of applicability, its physical constraint, and its analytic root regions described in section 6.2.3.1 is unchanged for the new version of the algorithm using the

fixed $\Delta\omega t$ zero search method.

The examples chosen for the comparison of the versions of the algorithm using the derivative and the fixed $\Delta\omega t$ zero search methods are described in the next section.

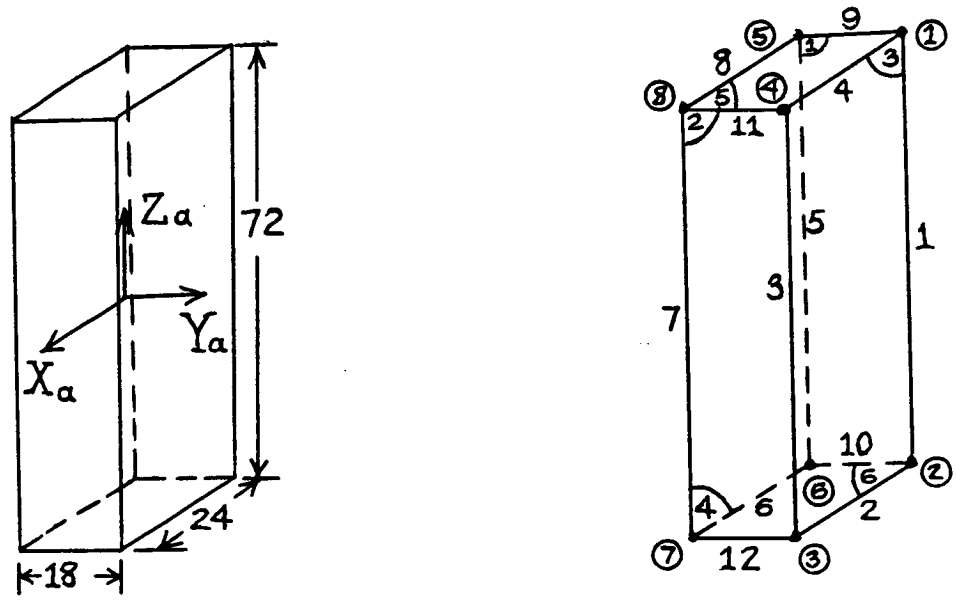
7.6.2 The Examples

Examples are considered using three different pairs of polyhedra for the moving object and obstacle. For the first two sets of examples, both $\mathcal{A}$ and $\mathcal{B}$ are represented by rectangular boxes. The boxes used for the first set of examples are the same boxes that were introduced in section 3.3.1 and used for the examples discussed in sections 6.3 and 7.1. As can be seen in Figure 3.8, the box representing $\mathcal{A}$ is much smaller than the box representing $\mathcal{B}$. In fact, the dimensions of the box representing $\mathcal{A}$ are one-fourth the dimensions of the box representing $\mathcal{B}$.

For the second set of examples, the box representing $\mathcal{A}$ is enlarged while the box representing $\mathcal{B}$ remains the same as that used for the first set. The dimensions of the two boxes for this second set are shown in Figure 7.26. In this case, the dimensions of the box representing $\mathcal{A}$ are three-fourths the dimensions of the box representing $\mathcal{B}$. In other words, the box representing $\mathcal{A}$ for this set of examples is three times larger than the box representing $\mathcal{A}$ for the first set of examples.

The rectangular boxes used to represent $\mathcal{A}$ and $\mathcal{B}$ for the first two sets of examples are convex polyhedra. For the third set of examples, $\mathcal{A}$ and $\mathcal{B}$ are represented by nonconvex polyhedra. Specifically, they are represented by the same two polyhedra used for the examples discussed in section 7.3. These two nonconvex polyhedra are shown in Figures 7.12 and 7.13.

The particular examples considered for each of these sets of polyhedra are described in the following sections. The quantities specifying the initial configuration and motion of $\mathcal{A}$ are listed in tables, as was the case for the examples discussed in sections 7.1 through 7.4. The vectors $\mathbf{x}_o$, $\mathbf{v}$, and $\vec{\omega}$ are given in reference coordinates and the initial orientation of $\mathcal{A}$ is given as a set of yaw, pitch, and roll angles. The tables also list the values of $\|\mathbf{x}_o - \mathbf{x}^b\|$, $\|\mathbf{v}\|$, ω, and $r_a\omega/\|\mathbf{v}\|$. Sketches of $\mathcal{A}$ and $\mathcal{B}$ at the initial configuration and at the time of collision, such as those given for the examples in the first four sections of this chapter, are omitted for these examples. Instead, the times of first contact and the features of $\mathcal{A}$ and $\mathcal{B}$ that are touching at first



The radius of the sphere enclosing $\mathcal{A}$ is $r_a = 39$.
Object $\mathcal{A}$

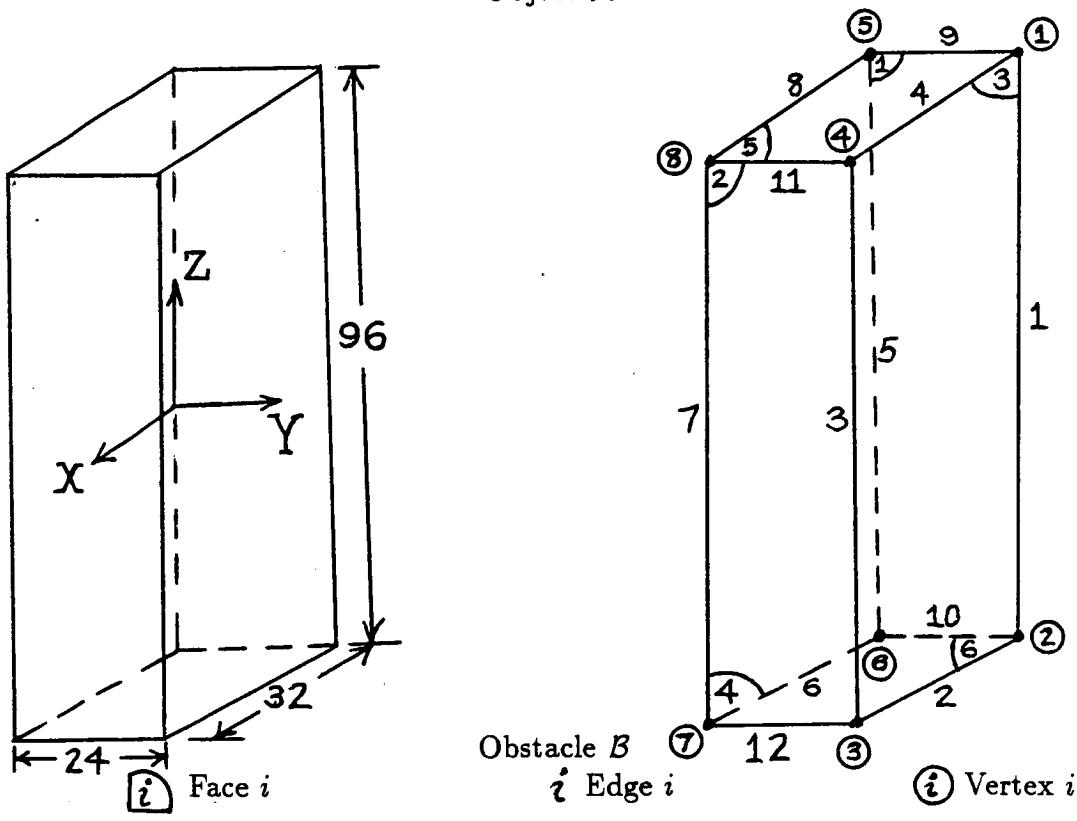


Figure 7.26: The set of rectangular boxes having the larger box for $\mathcal{A}$

contact are listed in a table.

7.6.2.1 The Two Sets of Rectangular Box Examples

Twelve examples are considered for each of the two sets of rectangular boxes. These twelve examples are divided into three subsets containing four examples of each of the three types of contacts at collision.

Tables 7.7, 7.8, 7.9, and 7.10 summarize the examples using the smaller rectangular box for $\mathcal{A}$. The vectors describing the initial configuration and motion of $\mathcal{A}$ are given in Tables 7.7, 7.8, and 7.9. These three tables list the inputs for the examples with type (a), (b), and (c) contacts at collision, respectively. The times of first contact and the features touching at collision for all twelve of these examples are given in Table 7.10. The first of these examples is the same example given for a type (a) contact at collision in section 7.1.

Similarly, the twelve examples using the larger rectangular box for $\mathcal{A}$ are summarized in Tables 7.11, 7.12, 7.13, and 7.14. The inputs for the examples with type (a), (b), and (c) contacts at collision are listed in Tables 7.11, 7.12, and 7.13, respectively. Table 7.14 lists the times of first contact and the features touching at collision for all twelve of these examples.

As can be seen from the values of $r_a\omega/\|\mathbf{v}\|$ in the tables, six of the twelve examples for each set of boxes are cases where the rotation rate is high compared to the rate of translation. For these six cases, the value of $r_a\omega/\|\mathbf{v}\|$ is much larger than one. The remaining six examples are cases where the rate of rotation is low compared to the rate of translation. The value of $r_a\omega/\|\mathbf{v}\|$ is less than one for these cases.

Initial Position of C^a	
Example #	$\mathbf{x}_o$
1,2,3,4	$[\ -32 \quad +25 \quad +100\ ]^T$

(Constant) Linear Velocity Vector of $\mathcal{A}$	
Example #	$\mathbf{v}$
1,2,3,4	$[\ +0.28603370 \quad -0.23240238 \quad -0.92960952\ ]^T$

Initial Orientation of $\mathcal{A}$			
Example #	Yaw (°)	Pitch (°)	Roll (°)
1,3	0	0	0
2,4	-36.869898	0	0

(Constant) Angular Velocity Vector of $\mathcal{A}$	
Example #	$\vec{\omega}(°/t)$
1	$[\ -6.3508528 \quad -31.754265 \quad +44.455971\ ]^T$
2	$[\ -29.2 \quad -40.6 \quad 0\ ]^T$
3	$[\ -0.11547005 \quad +0.57735027 \quad +0.80829038\ ]^T$
4	$[\ +0.632 \quad +0.776 \quad 0\ ]^T$

Example #	$\|\mathbf{x}_o - \mathbf{x}^b\|$	$\|\mathbf{v}\|$	$\omega = \|\vec{\omega}\|\,(°/t)$	$r_a\omega/\|\mathbf{v}\|$
1	107.93	1	55	12.48
2	107.93	1	50.01	11.35
3	107.93	1	1	0.227
4	107.93	1	1.001	0.227

Table 7.7: The inputs for the type (a) contact examples using the smaller rectangular box for $\mathcal{A}$

Initial Position of C^a	
Example #	$\mathbf{x}_o$
5,6,7,8	$[\ +10 \quad +100 \quad -25\ ]^T$

(Constant) Linear Velocity Vector of $\mathcal{A}$	
Example #	$\mathbf{v}$
5,6,7,8	$[\ -0.10866356 \quad -0.95623934 \quad +0.27165890\ ]^T$

Initial Orientation of $\mathcal{A}$			
Example #	Yaw (°)	Pitch (°)	Roll (°)
5,6,7,8	0	0	0

(Constant) Angular Velocity Vector of $\mathcal{A}$	
Example #	$\vec{\omega}(°/t)$
5	$[\ +5 \quad 0 \quad +49.5\ ]^T$
6	$[\ -6.3508528 \quad -31.754265 \quad +44.455971\ ]^T$
7	$[\ +0.1 \quad 0 \quad +0.99\ ]^T$
8	$[\ -0.11547005 \quad -0.57735027 \quad +0.80829038\ ]^T$

Example #	$\|\mathbf{x}_o - \mathbf{x}^b\|$	$\|\mathbf{v}\|$	$\omega = \|\vec{\omega}\|\ (°/t)$	$r_a\omega/\|\mathbf{v}\|$
5	103.56	1	49.752	11.29
6	103.56	1	55	12.48
7	103.56	1	0.995	0.226
8	103.56	1	1	0.227

Table 7.8: The inputs for the type (b) contact examples using the smaller rectangular box for $\mathcal{A}$

Initial Position of C^a	
Example #	$\mathbf{x}_o$
9,10,11,12	$[\ +100 \quad -75 \quad -30\ ]^T$

(Constant) Linear Velocity Vector of $\mathcal{A}$	
Example #	$\mathbf{v}$
9,10,11,12	$[\ -0.76921916 \quad +0.57691437 \quad +0.27472113\ ]^T$

Initial Orientation of $\mathcal{A}$			
Example #	Yaw (°)	Pitch (°)	Roll (°)
9,10,11,12	-36.869898	0	0

(Constant) Angular Velocity Vector of $\mathcal{A}$	
Example #	$\vec{\omega}(°/t)$
9	$[\ +30.8 \quad +39.4 \quad 0\ ]^T$
10	$[\ -24.133240 \quad -21.592903 \quad +44.455971\ ]^T$
11	$[\ +0.616 \quad +0.788 \quad 0\ ]^T$
12	$[\ -0.21959309 \quad -0.1962911 \quad +0.40414519\ ]^T$

Example #	$\|\mathbf{x}_o - \mathbf{x}^b\|$	$\|\mathbf{v}\|$	$\omega = \|\vec{\omega}\|\ (°/t)$	$r_a\omega/\|\mathbf{v}\|$
9	128.55	1	50.01	11.35
10	128.55	1	55	12.48
11	128.55	1	1.0002	0.227
12	128.55	1	0.5	0.113

Table 7.9: The inputs for the type (c) contact examples using the smaller rectangular box for $\mathcal{A}$

Example #	Time of First Contact	Features Touching at First Contact	
Type (a) Contacts		Face of $\mathcal{A}$	Vertex of $\mathcal{B}$
1	45.666	2	1
2	43.796	6	1
3	48.927	4	1
4	51.761	2	1
Type (b) Contacts		Vertex of $\mathcal{A}$	Face of $\mathcal{B}$
5	87.231	5	3
6	80.790	1	3
7	86.397	5	3
8	83.392	5	3
Type (c) Contacts		Edge of $\mathcal{A}$	Edge of $\mathcal{B}$
9	97.981	11	7
10	99.542	4	7
11	96.334	12	7
12	99.998	4	7

Table 7.10: The resulting collisions for the examples using the smaller rectangular box for $\mathcal{A}$

Initial Position of C^a	
Example #	$\mathbf{x}_o$
1,2,3,4	$[\ -32 \quad +25 \quad +100\ ]^T$

(Constant) Linear Velocity Vector of $\mathcal{A}$	
Example #	$\mathbf{v}$
1,2,3,4	$[\ +0.28603370 \quad -0.23240238 \quad -0.92960952\ ]^T$

Initial Orientation of $\mathcal{A}$			
Example #	Yaw (°)	Pitch (°)	Roll (°)
1,3	0	0	0
2	-36.869898	0	+90
4	-36.869898	0	0

(Constant) Angular Velocity Vector of $\mathcal{A}$	
Example #	$\vec{\omega}(°/t)$
1	$[\ -6.3508528 \quad -31.754265 \quad +44.455971\ ]^T$
2	$[\ -1.6 \quad +1.2 \quad +50\ ]^T$
3	$[\ -0.11547005 \quad +0.57735027 \quad +0.80829038\ ]^T$
4	$[\ +0.632 \quad +0.776 \quad 0\ ]^T$

Example #	$\|\mathbf{x}_o - \mathbf{x}^b\|$	$\|\mathbf{v}\|$	$\omega = \|\vec{\omega}\|\ (°/t)$	$r_a\omega/\|\mathbf{v}\|$
1	107.93	1	55	37.44
2	107.93	1	50.04	34.06
3	107.93	1	1	0.681
4	107.93	1	1.001	0.681

Table 7.11: The inputs for the type (a) contact examples using the larger rectangular box for $\mathcal{A}$

Initial Position of C^a	
Example #	$\mathbf{x}_o$
5,6,7,8	$[\ +10 \quad +100 \quad -25\ ]^T$

(Constant) Linear Velocity Vector of $\mathcal{A}$	
Example #	$\mathbf{v}$
5,6,7,8	$[\ -0.10866356 \quad -0.95623934 \quad +0.27165890\ ]^T$

Initial Orientation of $\mathcal{A}$			
Example #	Yaw (°)	Pitch (°)	Roll (°)
5,6,7,8	0	0	0

(Constant) Angular Velocity Vector of $\mathcal{A}$	
Example #	$\vec{\omega}(°/t)$
5	$[\ +5 \quad 0 \quad +49.5\ ]^T$
6	$[\ -6.3508528 \quad -31.754265 \quad +44.455971\ ]^T$
7	$[\ +0.1 \quad 0 \quad +0.99\ ]^T$
8	$[\ +0.11547005 \quad -0.57735027 \quad +0.80829038\ ]^T$

Example #	$\|\mathbf{x}_o - \mathbf{x}^b\|$	$\|\mathbf{v}\|$	$\omega = \|\vec{\omega}\|\ (°/t)$	$r_a\omega/\|\mathbf{v}\|$
5	103.56	1	49.752	33.87
6	103.56	1	55	37.44
7	103.56	1	0.995	0.677
8	103.56	1	1	0.681

Table 7.12: The inputs for the type (b) contact examples using the larger rectangular box for $\mathcal{A}$

Initial Position of C^a	
Example #	$\mathbf{x}_o$
9,10,11,12	$[\ +100 \quad -75 \quad -30\ ]^T$

(Constant) Linear Velocity Vector of $\mathcal{A}$	
Example #	$\mathbf{v}$
9,10,11,12	$[\ -0.76921916 \quad +0.57691437 \quad +0.27472113\ ]^T$

Initial Orientation of $\mathcal{A}$			
Example #	Yaw (°)	Pitch (°)	Roll (°)
9,10,11,12	-36.869898	0	0

(Constant) Angular Velocity Vector of $\mathcal{A}$	
Example #	$\vec{\omega}(°/t)$
9	$[\ +30.8 \quad +39.4 \quad 0\ ]^T$
10	$[\ -24.133240 \quad -21.592903 \quad +44.455971\ ]^T$
11	$[\ +0.616 \quad +0.788 \quad 0\ ]^T$
12	$[\ -0.43878618 \quad -0.39259821 \quad +0.80829038\ ]^T$

Example #	$\|\mathbf{x}_o - \mathbf{x}^b\|$	$\|\mathbf{v}\|$	$\omega = \|\vec{\omega}\|\,(°/t)$	$r_a\omega/\|\mathbf{v}\|$
9	128.55	1	50.01	34.04
10	128.55	1	55	37.44
11	128.55	1	1.0002	0.681
12	128.55	1	1	0.681

Table 7.13: The inputs for the type (c) contact examples using the larger rectangular box for $\mathcal{A}$

Example #	Time of First Contact	Features Touching at First Contact	
Type (a) Contacts		Face of $\mathcal{A}$	Vertex of $\mathcal{B}$
1	19.573	2	1
2	46.522	4	1
3	28.446	4	1
4	42.606	2	1
Type (b) Contacts		Vertex of $\mathcal{A}$	Face of $\mathcal{B}$
5	73.407	5	3
6	54.847	1	3
7	73.746	5	3
8	63.779	5	3
Type (c) Contacts		Edge of $\mathcal{A}$	Edge of $\mathcal{B}$
9	69.764	11	7
10	79.939	4	7
11	69.751	10	7
12	79.479	4	7

Table 7.14: The resulting collisions for the examples using the larger rectangular box for $\mathcal{A}$

7.6.2.2 The Set of Examples using Nonconvex Polyhedra

Six examples are considered for the two nonconvex polyhedra of section 7.3. This includes two examples of a type (a) contact at collision, one example of a type (b) contact at collision, and three examples of a type (c) contact at collision. The inputs for the two type (a) contact examples are listed in Table 7.15. Table 7.16 lists the inputs for the single type (b) contact example and Table 7.17 lists the inputs for the three type (c) contact examples. These three tables show that the quantity $r_a\omega/\|\mathbf{v}\|$ takes on both high and low values for these examples.

The times of first contact and the features touching at first contact are given for all six examples in Table 7.18. The examples of a type (a), (b), and (c) contact at collision described in section 7.3 correspond to examples 1, 3, and 5, respectively. The example of an unexpected contact at collision for these polyhedra given in section 7.4 corresponds to example 4.

C-4

Initial Position of C^a	
Example #	$\mathbf{x}_o$
1	$[\ +5 \quad +64 \quad +35\ ]^T$
2	$[\ +55 \quad -41.25 \quad +75\ ]^T$

(Constant) Linear Velocity Vector of $\mathcal{A}$	
Example #	$\mathbf{v}$
1	$[\ -0.12369267 \quad -0.98954139 \quad -0.074215604\ ]^T$
2	$[\ -4.8988316 \quad +3.6741237 \quad -3.3914988\ ]^T$

Initial Orientation of $\mathcal{A}$			
Example #	Yaw (°)	Pitch (°)	Roll (°)
1	0	0	0
2	-36.869898	0	0

(Constant) Angular Velocity Vector of $\mathcal{A}$	
Example #	$\vec{\omega}(°/t)$
1	$[\ +3.9777864 \quad +2.7844505 \quad +1.1933359\ ]^T$
2	$[\ -13.4 \quad -21.2 \quad 0\ ]^T$

Example #	$\|\mathbf{x}_o - \mathbf{x}^b\|$	$\|\mathbf{v}\|$	$\omega = \|\vec{\omega}\|\ (°/t)$	$r_a\omega/\|\mathbf{v}\|$
1	67.93	1	5	1.630
2	104.32	7	25.08	1.168

Table 7.15: The inputs for the type (a) contact examples using the non-convex polyhedra

Initial Position of C^a	
Example #	$\mathbf{x}_o$
3	$[\ +10 \quad +56 \quad +75\ ]^T$

(Constant) Linear Velocity Vector of $\mathcal{A}$	
Example #	$\mathbf{v}$
3	$[\ -0.18340136 \quad -0.58688434 \quad -0.78862584\ ]^T$

Initial Orientation of $\mathcal{A}$			
Example #	Yaw (°)	Pitch (°)	Roll (°)
3	0	0	0

(Constant) Angular Velocity Vector of $\mathcal{A}$	
Example #	$\vec{\omega}(°/t)$
3	$[\ +3.9777864 \quad +2.7844505 \quad +1.1933359\ ]^T$

Example #	$\|\mathbf{x}_o - \mathbf{x}^b\|$	$\|\mathbf{v}\|$	$\omega = \|\vec{\omega}\|\ (°/t)$	$r_a\omega/\|\mathbf{v}\|$
3	90.69	1	5	1.630

Table 7.16: The inputs for the type (b) contact example using the nonconvex polyhedra

Initial Position of C^a	
Example #	$\mathbf{x}_o$
4	$[\ -5 \quad +42 \quad +73\ ]^T$
5,6	$[\ +55 \quad -41.25 \quad -30\ ]^T$

(Constant) Linear Velocity Vector of $\mathcal{A}$	
Example #	$\mathbf{v}$
4	$[\ +0.38100039 \quad -2.2860023 \quad -1.9050019\ ]^T$
5	$[\ -8.1759198 \quad +6.1319399 \quad +6.2891691\ ]^T$
6	$[\ -1.3626533 \quad +1.0219900 \quad +1.0481949\ ]^T$

Initial Orientation of $\mathcal{A}$			
Example #	Yaw (°)	Pitch (°)	Roll (°)
4	-90	0	0
5,6	-36.869898	0	0

(Constant) Angular Velocity Vector of $\mathcal{A}$	
Example #	$\vec{\omega}(°/t)$
4	$[\ +14.9256 \quad 0 \quad +1.49256\ ]^T$
5,6	$[\ -15 \quad -20 \quad 0\ ]^T$

Example #	$\|\mathbf{x}_o - \mathbf{x}^b\|$	$\|\mathbf{v}\|$	$\omega = \|\vec{\omega}\|\,(°/t)$	$r_a\omega/\|\mathbf{v}\|$
4	81.55	3	15	1.630
5	78.47	12	25	0.679
6	78.47	2	25	4.076

Table 7.17: The inputs for the type (c) contact examples using the nonconvex polyhedra

Example #	Time of First Contact	Features Touching at First Contact	
Type (a) Contacts		Face of $\mathcal{A}$	Vertex of $\mathcal{B}$
1	36.15	1 of $\mathcal{A}_1$	1 of $\mathcal{B}_2$
2	5.965	5 of $\mathcal{A}_2$	8 of $\mathcal{B}_1$
Type (b) Contacts		Vertex of $\mathcal{A}$	Face of $\mathcal{B}$
3	45.70	2 of $\mathcal{A}_1$	2 of $\mathcal{B}_2$
Type (c) Contacts		Edge of $\mathcal{A}$	Edge of $\mathcal{B}$
4	8.963	8 of $\mathcal{A}_1$	4 of $\mathcal{B}_1$
5	3.595	9 of $\mathcal{A}_1$	7 of $\mathcal{B}_1$
6	18.92	11 of $\mathcal{A}_2$	7 of $\mathcal{B}_1$

Table 7.18: The resulting collisions for the six examples using the nonconvex polyhedra

7.6.3 Comparison of the Execution Times for the Two Zero Search Methods

The CPU time required to execute the versions of the collision detection algorithm using the two different zero search methods was measured for each of the examples described in section 7.6.2. Both versions of the algorithm were implemented in FORTRAN on a VAX 11/780 computer. Access to this computer's memory is shared among many users so that the computers resources were not completely dedicated to running the collision detection programs. Also, the FORTRAN versions of the algorithm are experimental versions primarily intended to verify the concepts presented in the previous chapters of this thesis. Relatively little effort was devoted to optimizing this code. For these reasons, the CPU times listed in the tables below are presented only for the purpose of comparing the two zero search methods. These times should not be interpreted as absolute measures of the computational efficiency of the algorithm.

The value of $\Delta\omega t$ to be used for the fixed $\Delta\omega t$ zero search was specified as an additional input to the new version of the collision detection algorithm. The execution time required for the new version of the algorithm was measured for three different values of $\Delta\omega t$. Thus, each of the examples was run four times. The first run used the collision detection algorithm with the derivative zero search method and the last three runs used the collision detection algorithm with the fixed $\Delta\omega t$ zero search method with $\Delta\omega t$ set to three different values. The three values of $\Delta\omega t$ that were used and the corresponding number of divisions per cycle of ωt are shown below:

$\Delta\omega t$	# of divisions per cycle
60°	6
36°	10
18°	20

The execution times measured for the twelve examples using the smaller rectangular box for $\mathcal{A}$ are shown in Table 7.19. The execution times for the examples using the larger rectangular box for $\mathcal{A}$ are shown in Table 7.20.

There are four cases shown in these two tables for which the algorithm using the fixed $\Delta\omega t$ zero search method fails to predict the correct time of first contact. These are indicated by an asterisk (*) in the tables. The reasons for these failures will be discussed in section 7.6.3.1.

The differences in execution time among the examples using the same set of boxes will be discussed in section 7.6.3.2. This section also contrasts the execution times required for similar examples using the two different sets of boxes.

Table 7.21 lists the execution times for the six examples using the two nonconvex polyhedra for $\mathcal{A}$ and $\mathcal{B}$. These results are discussed in section 7.6.3.3. The examples having different types of contacts at collision are compared. In addition, the execution times for the examples using the two nonconvex polyhedra are compared with those for the rectangular box examples.

Section 7.6.3.4 summarizes the results of the comparison of the execution times for the versions of the algorithm using the fixed $\Delta\omega t$ zero search method and the derivative zero search method.

Example #	Type of Collision	$r_a\omega/\|\mathbf{v}\|$	CPU Time (sec)			
			Derivative Zero Search	Fixed $\Delta\omega t$ Zero Search (# of Divisions per Cycle)		
				6	10	20
High Rotation Rate Examples						
1	a	12.5	1.080	1.026	1.040	1.070
2	a	11.3	0.991	*	*	1.006
5	b	11.3	0.761	0.743	0.750	0.751
6	b	12.5	0.847	0.831	0.831	0.833
9	c	11.3	1.051	1.006	1.012	1.030
10	c	12.5	1.072	1.037	1.042	1.058
Low Rotation Rate Examples						
3	a	0.227	0.939	0.906	0.907	0.907
4	a	0.227	0.881	0.869	0.867	0.870
7	b	0.226	0.757	0.743	0.745	0.743
8	b	0.227	0.852	0.826	0.827	0.832
11	c	0.227	0.835	0.819	0.814	0.820
12	c	0.113	0.926	0.903	0.900	0.901

* The version of the algorithm using the fixed $\Delta\omega t$ zero search method did not find the correct first contact for these cases.

Table 7.19: Execution times for the examples using the smaller rectangular box for $\mathcal{A}$

Example #	Type of Collision	$r_a\omega/\|\mathbf{v}\|$	CPU Time (sec)			
			Derivative Zero Search	Fixed $\Delta\omega t$ Zero Search (# of Divisions per Cycle)		
				6	10	20
High Rotation Rate Examples						
1	a	37.4	1.272	1.201	1.232	1.314
2	a	34.1	1.891	1.669	1.725	1.869
5	b	33.9	1.185	1.089	*	1.126
6	b	37.4	1.044	1.010	1.028	1.067
9	c	34.0	1.532	*	1.477	1.552
10	c	37.4	1.420	1.330	1.410	1.457
Low Rotation Rate Examples						
3	a	0.681	0.962	0.937	0.941	0.948
4	a	0.681	0.902	0.901	0.903	0.911
7	b	0.677	0.806	0.799	0.800	0.800
8	b	0.681	0.909	0.904	0.906	0.910
11	c	0.681	0.882	0.860	0.862	0.864
12	c	0.681	0.944	0.938	0.942	0.943

* The version of the algorithm using the fixed $\Delta\omega t$ zero search method did not find the correct first contact for these cases.

Table 7.20: Execution times for the examples using the larger rectangular box for $\mathcal{A}$

Example #	Type of Collision	$r_a\omega/\|\|\mathbf{v}\|\|$	CPU Time (sec)			
			Derivative Zero Search	Fixed $\Delta\omega t$ Zero Search (# of Divisions per Cycle)		
				6	10	20
1	a	1.630	4.119	4.084	4.088	4.102
2	a	1.168	2.854	2.837	2.840	2.844
3	b	1.630	3.054	3.021	3.023	3.034
4	c	1.630	2.888	2.850	2.856	2.859
5	c	0.679	2.622	2.629	2.630	2.633
6	c	4.076	2.711	2.701	2.690	2.712

Table 7.21: Execution times for the examples using the nonconvex polyhedra

7.6.3.1 Failure of the Fixed $\Delta\omega t$ Zero Search Method to Find the Correct First Contact

The first example for which the new version of the algorithm gave an incorrect answer for the first contact was example 7 using the smaller rectangular box for $\mathcal{A}$. The failure occurred for the runs using 6 and 10 divisions per cycle of ωt. This corresponds to the two larger of the three fixed values of $\Delta\omega t$, 60° and 36°. For this example, the C-function determining first contact, $f^a_{6,1}$, has a relative maximum at $t = 43.518$ in the search interval $[43.191, 43.935]$. There are two zeros of $f^a_{6,1}$ in this interval, one that is not allowable between $t = 43.191$ and $t = 43.518$ and an allowable zero at $t = 43.796$ that represents the first contact between $\mathcal{A}$ and $\mathcal{B}$.

The derivative zero search method used the point where the maximum occurs, $t = 43.518$, as one of the search points and correctly interpolated between $t = 43.518$ and $t = 43.935$ for the allowable zero at $t = 43.796$. However, the relative maximum of $f^a_{6,1}$ was ignored by the fixed $\Delta\omega t$ zero search method. The value of $f^a_{6,1}$ is negative for all of the test points used by this method in the interval $[43.191, 43.935]$. Therefore, the new version of the algorithm concluded that $f^a_{6,1}$ had no zeros in this interval. For $\Delta\omega t = 36°$ and $\Delta\omega t = 60°$, the spacing between test points was too large for the new version of the algorithm to find the zero of $f^a_{6,1}$ representing the first contact of $\mathcal{A}$ and $\mathcal{B}$. Hence, it erroneously concluded that a zero of a different C-function represented the first contact.

The two examples for which the new version of the algorithnm failed to find the correct first contact when the larger rectangular box was used for $\mathcal{A}$ were examples 3 and 5. For example 3, the incorrect answer was obtained for the run using 10 divisions per cycle of ωt where $\Delta\omega t = 36°$. In this case, the C-function determining first contact, $f^b_{5,3}$, has a relative minimum at $t = 73.634$ in the search interval $[73.231, 74.168]$. As was the case for $f^a_{6,1}$ in the example discussed above, $f^b_{5,3}$ has two zeros in this interval, an allowable one at $t = 73.407$ that is the first contact and one that is not allowable at $t = 73.863$.

The derivative zero search used the point where the minimum occurs, $t = 73.634$, as one of the search points and correctly interpolated between $t = 73.321$ and $t = 73.634$ for the allowable zero at $t = 73.407$. The fixed $\Delta\omega t$ zero search used a search point at $t = 73.806$ that was slightly beyond the exact minimum. This caused the iteration routine using the secant

method to converge to the second zero at $t = 73.863$, the one that is not allowable. This zero lies outside the interval $[73.321, 73.806]$ used to start the secant method, but the algorithm did not recognize this. It concluded that the zero was in that interval and, hence, that it was allowable. Thus, for this example, the new version of the algorithm correctly identified the C-function whose zero represents the first contact of $\mathcal{A}$ and $\mathcal{B}$. However, the behavior of the iteration routine caused it to predict that another zero of this function represented the first contact.

For example 5 using the larger rectangular box of $\mathcal{A}$, the new version of the algorithm failed to correctly identify the first contact using 6 divisions per cycle of ωt where $\Delta\omega t = 60°$. The C-function determining first contact, $f^c_{11,7}$, has a relative minimum at $t = 69.850$ in the search interval $[69.722, 70.186]$. It has positive orientation in this interval. There are two zeros of $f^c_{11,7}$ in this interval, an allowable one at $t = 69.764$ that is the first contact and one that is not allowable between $t = 69.850$ and $t = 70.186$.

The derivative zero search used the point where the minimum occurs, $t = 69.850$, as one of the search points and correctly interpolated between $t = 69.722$ and $t = 69.850$ for the allowable zero at $t = 69.764$. The fixed $\Delta\omega t$ zero search used only two search points — the beginning and end of the interval. The value of $f^c_{11,7}$ is positive at both of these points, so the algorithm concluded that this C-function had no zeros in the interval. Thus, it also concluded that a zero of a different C-function represented the first contact of $\mathcal{A}$ and $\mathcal{B}$.

From the discussion in the preceding paragraphs, it is clear that the fixed $\Delta\omega t$ zero search method is sensitive to the value of $\Delta\omega t$. If the choice of this spacing is too large, it can fail to detect some zeros of the C-functions. This will cause the algorithm either to predict an incorrect time of first contact or to fail to detect a collision when one does occur. This is more likely for cases with high rotation rates relative to the linear speed because the C-functions change more rapidly in these cases.

The sensitivity to the choice of $\Delta\omega t$ is a disadvantage of the fixed $\Delta\omega t$ zero search method. This problem does not occur when using the derivative zero search method because the test points are chosen to insure that no zeros of a C-function are overlooked.

7.6.3.2 Comparison of the Execution Times for the Two Sets of Rectangular Box Examples

Tables 7.19 and 7.20 show that the examples using the larger box for $\mathcal{A}$ generally require slightly greater execution times using either algorithm than the examples using the smaller rectangular box for $\mathcal{A}$. This can be attributed to the relative sizes of $\mathcal{A}$ and $\mathcal{B}$. The volumes of the regions where the physical constraints are satisfied are increased by enlarging the box representing $\mathcal{A}$. Therefore, the trajectory of C^a is more likely to penetrate one of these regions when $\mathcal{A}$ is represented by the larger box. Most of the C-functions will satisfy their physical constraints for these examples. The physical constraints are much less likely to be satisfied when $\mathcal{A}$ is represented by the smaller box. Many more C-functions are eliminated from the zero search by the physical constraints for these examples than for the examples using the larger rectangular box for $\mathcal{A}$.

The high rotation rate examples have longer running times than the low rotation rate examples because of the nature of the C-functions. As explained in the beginning of this chapter, these functions are more oscillatory and have many relative maxima and minima for high rotation rates. This means that they probably have more allowable zeros, which the collision detection algorithm must find and check for valid collisions. For lower rotation rates, the C-functions are less oscillatory. They do not change as quickly and may have no zeros or only one zero in the interval of interest. This means that the iteration routine for finding the zeros and the check of the third collision condition are executed less often for examples having low rotation rates than for examples having high rotation rates relative to the rate of translation.

Type (a) and (c) collisions take longer to detect than type (b) collisions. This is expected because the type (a) and (c) C-functions have more complex behavior as functions of time and require a more complicated zero search procedure than the type (b) C-functions. Also, both versions of the algorithm search for zeros of the type (b) C-functions before the type (a) and (c) C-functions are examined. When a valid collision is found for a type (b) C-function, the updated zero search interval for the remaining C-functions is smaller than when no valid collision is found for any type (b) C-function.

In comparing the execution time for the two zero search methods for

the two sets of rectangular box examples, the general trend is that the fixed $\Delta\omega t$ zero search method takes slightly less time than the derivative zero search method. On the average, 2-4% less CPU time is used for the runs specifying 6-10 divisions per cycles and only 1% less CPU time is used for the runs specifying 20 divisions per cycle. In most of the cases, the execution times are not significantly different for the versions of the algorithm using the two zero search methods.

7.6.3.3 Comparison of the Examples using Nonconvex Polyhedra

An important factor that influences the execution time required by the collision detection algorithm when $\mathcal{A}$ are $\mathcal{B}$ are nonconvex polyhedra is the number of pairs of convex polyhedra that must be considerd separately. Each additional pair of convex polyhedra defines another set of C-functions that must be examined for allowable zeros that may be valid collisions of $\mathcal{A}$ and $\mathcal{B}$. For this set of examples, there are four pairs of convex polyhedra to be checked because each of the nonconvex polyhedra representing $\mathcal{A}$ and $\mathcal{B}$ has two convex pieces. There are three additional pairs of convex polyhedra to be considered beyond the single pair for the rectangular box examples considered above. This is the reason for the larger values of the execution times listed in Table 7.21 relative to the values listed in Tables 7.19 and 7.20. These examples take $2-4$ times longer than the examples using either of the two sets of rectangular boxes for $\mathcal{A}$ and $\mathcal{B}$.

The type of contact at collision was one factor mentioned in the previous section that influences the amount of time required to execute the collision detection algorithm. When $\mathcal{A}$ or $\mathcal{B}$, or both, is a nonconvex polyhedron, another factor that influences the execution time is which pair of convex polyhedra $\mathcal{A}_i$ and $\mathcal{B}_j$ first come into contact. For examples 1 and 3, the first contact occurs between $\mathcal{A}_1$ and $\mathcal{B}_1$ Since this is the first pair of convex polyhedra to be considered by the algorithm, these examples take the least time of all the examples for the two nonconvex polyhedra. The reason for this is that the upper bound of the zero search interval is decreased to the time of first contact found for the first pair of polyhedra when the other three pairs of polyhedra are considered. For examples 2, 4, 5, and 6, the upper bound of the search interval is unchanged after the examination of $\mathcal{A}_1$ and $\mathcal{B}_1$. Hence, the algorithm must examine the C-functions defined by

the remaining three pairs of polyhedra in a larger interval than is the case for examples 1 and 3.

The first contact occurs between $\mathcal{A}_1$ and $\mathcal{B}_2$ for examples 2 and 4. This is the second pair of convex polyhedra to be considered by the algorithm. Also, the trajectory for these two examples is such that most of the C-functions defined by all four pairs of polyhedra satisfy their physical constraints. Therefore, the physical constraints do not eliminate many C-functions from the zero search. For these reasons, these examples take the most time of all the examples considered for the two nonconvex polyhedra.

For examples 5 and 6, the first contact occurs betweeen $\mathcal{A}_2$ and $\mathcal{B}_1$. This is the third pair of convex polyhedra considered by the algorithm. However, the trajectories for these examples are such that the physical constraints eliminate most of the C-functions involving features of $\mathcal{B}_2$ from the zero search. This causes these two examples to take less time than examples 2 and 4, even though the first contact is not found until the third pair of polyhedra is considered.

The results given in Table 7.21 show that there is a negligible reduction in the execution time for the version of the algorithm using the fixed $\Delta\omega t$ zero search method relative to the version using the derivative zero search method. For the six examples using the nonconvex polyhedra, the version of the algorithm using the fixed $\Delta\omega t$ zero search method took an average of 0.3 - 0.7% less time than the version using the derivative zero search method.

7.6.3.4 Summary of the Comparison of the Two Zero Search Methods

No significant difference has been found in the execution times for the version of the collision detection algorithm using the fixed $\Delta\omega t$ zero search method and the version using the derivative zero search method for any of the three sets of examples considered in this section. The cases in section 7.6.3.1 show that using the fixed $\Delta\omega t$ zero search method can sometimes cause the algorithm to give an incorrect answer to the collision detection problem. The chances of missing zeros of some C-functions can be reduced by choosing smaller values of $\Delta\omega t$, but the execution time increases as the value of $\Delta\omega t$ decreases and any savings compared with the version of the algorithm using the derivative zero search method are reduced. The

loss of reliability for the new version of the algorithm far outweighs the benefit of reduced execution time. In almost all cases, the algorithm described in Chapter 6 will give the correct answer to the collision detection problem because it cannot overlook any zeros of the C-functions. This algorithm is clearly superior to the version that uses the fixed $\Delta\omega t$ zero search method because it provides provides a much higher degree of reliability using a comparable amount of CPU time.

Chapter 8

Conclusions and Recommendations

8.1 Conclusions

As stated in Chapter 1, the goal of this thesis was to develop a reliable collision detection algorithm to be used when two spacecraft are operating in proximity. The previous chapters of this thesis have described the development of the new collision detection algorithm based on the C-function formulation of the configuration space approach. Both spacecraft are modelled as nonconvex polyhedra. The case where one or both of the spacecraft is represented by a single convex polyhedron is included as a special case.

The new algorithm is designed to handle the case where the moving craft translates with constant linear velocity and rotates with constant angular velocity relative to the other spacecraft. The algorithm makes special provisions for the cases where the moving vehicle is translating with constant linear velocity but not rotating and where it is rotating with constant angular velocity but not translating.

The assumption of constant linear and angular velocities results in forms for the C-functions as functions of time that have transcendental terms. Three major contributions have been made in this thesis that provide the new collision detection algorithm with the ability to handle these transcendental functions. The first contribution is the development of methods to bound the intervals of time that must be searched for zeros of the C-functions. An overall search interval for each C-function is found using simple linear and quadratic inequalities when the moving object is translating with constant linear velocity. Then, the zero search is restricted to periodic subintervals of this overall interval by taking advantage of the geometry of the objects and the direction of the constant angular velocity vector. Only zeros of the C-functions within these subintervals can represent potential collisions.

The second contribution of this thesis is the development of methods to locate the exact zeros of the C-functions in the given bounded search intervals. This was accomplished by analyzing the form of the generalized

C-function $f(t)$. First, criteria determining the analytic root intervals of $f(t)$ were derived. Although these intervals are unbounded in most cases, they can be used in conjunction with the methods mentioned above to further limit the zero search intervals. Then, analytic formulas or numerical procedures for locating the zeros of $f(t)$ were found for each of twelve cases depending on the values of its six coefficients.

The numerical zero search procedures were designed to take advantage of the special forms of the first and second derivatives of $f(t)$. These special forms allow all of the points where these derivatives equal zero to be found. These points are used in the search for the zeros of the function itself. Including the points where the first derivative is zero, which are the relative maxima and minima of the C-function, guarantees that none of the zeros in the search regions for a C-function is ignored by the algorithm. This means that the new algorithm can find the exact time of first contact within the accuracy of the computer's calculations.

This property of the new algorithm makes it more reliable than the method for collision detection used by the system described in reference [16]. In the latter algorithm, collisions are found by checking the C-functions at discrete values of time separated by a fixed time step. This approach is limited to the resolution of the chosen step size and can "miss" collisions that take place between two discrete check points. A similar problem can occur when the fixed $\Delta\omega t$ zero search method is substituted for the "derivative" zero search method in the new algorithm, as was demonstrated in Chapter 7.

The final contribution of this thesis is the design of the hierarchy for the collision detection algorithm that limits the search for zeros of the C-functions as much as possible. The level of detail in modelling the interaction of the two polyhedra increases as the algorithm progresses to the later levels of its hierarchy. The algorithm is structured as a series of successively more complicated and more accurate procedures, as described in Chapter 6. The most complex procedure used by the algorithm is the configuration space dynamic collision detection procedure which examines each of the C-functions individually. The hierarchy for the examination of a single C-function specifies the order in which the various methods for bounding the zero search intervals are considered.

The examples in Chapter 7 have shown that the new algorithm successfully predicts the time of the first collision for all three types of contacts. It

also correctly identifies the features of the moving and stationary polyhedra that are touching at the time of first contact.

The new collision detection algorithm is the first algorithm that can solve the collision detection problem exactly for the case where the moving object has constant linear and angular velocities. The only other algorithm that can find an exact answer to the collision detection problem is the one presented in reference [17]. However, that algorithm can only be applied to cases where the trajectory of the moving object can be written as a straight line in the configuration space. Modifications are also described for the algorithm in reference [17] that allow it to handle cases of polynomial paths in the configuration space. The assumption of constant angular velocity used by the new algorithm means that the trajectory for the moving object cannot be represented as a polynomial in the configuration space. The new collision detection algorithm can therefore be regarded as a generalization of the algorithm described in reference [17] because it assumes a more difficult, but more realistic, path for the configuration of the moving object in $C_{space}(\mathcal{A})$.

8.2 Recommendations

8.2.1 Improving the Collision Detection Algorithm

The collision detection algorithm can be improved by finding ways to minimize the detrimental effects of the accumulation of numerical errors in the special cases discussed in section 6.4. The choice of the tolerance on the zero value and its use in the algorithm should be further investigated. It may be beneficial to have different values for this tolerance, one that is used when checking the values of the C-functions and the ACFs and another that is used when checking the third collision condition. Also, additional code may be needed to recognize the special orientations when more than one C-function or ACF is exactly equal to zero.

Further improvement in the performance of the algorithm may result from changing the order in which it considers the pairs of convex polyhedra forming the moving object and the obstacle and the C-functions defined by a single pair of these polyhedra. The current version of the algorithm considers the pairs of convex polyhedra and the sets of different types of C-functions defined by each of these pairs in the order in which they are

numbered in the database. This numbering is arbitrarily chosen when the data are entered. The examples in Chapter 7 have shown that less computation time is required for cases where the collision occurs between one of the first pairs of convex polyhedra to be checked by the algorithm and when the time of first contact is a zero of one of the first C-functions to be examined by the algorithm. This is because the upper bound of the interval to be searched for zeros of the remaining C-functions is truncated to reflect the value of this valid collision.

The possibility of developing criteria for ordering the convex polyhedra and features of these polyhedra should be studied. The criteria involving the parts of the stationary obstacle might depend on the direction of the velocity vector. For example, the faces of $\mathcal{B}$ could be ordered according to the values of time when the linear trajectory of the center of rotation of $\mathcal{A}$ intersected the planes containing these faces. This would mean that the type (b) C-functions defined using a face of the obstacle that lies behind another face as seen from the initial position of the moving object would be checked after those defined using the face of the obstacle that is nearest to the initial position.

Similar criteria for ordering the convex polyhedra and their features for the moving object may be more difficult to formulate because the moving object changes its orientation relative to the obstacle. One possibility is to examine the applicability regions of all the C-functions. The current version of the algorithm stores an applicability timeline for each of the C-functions defined by the features of two convex polyhedra that are parts of the moving object and obstacle. An alternative to this would be to store an overall applicability timeline defining the intervals in a single cycle of ωt where the set of applicable C-functions remains fixed. The same features of the obstacle will identify the C-functions in each interval, but the features of the moving object defining the applicable C-functions will change from interval to interval.

The criteria for ordering the convex polyhedra and their features for both $\mathcal{A}$ and $\mathcal{B}$ would be combined to determine the final order in which the algorithm would examine the C-functions. However, the ordering criteria would have to be kept relatively simple to yield the greatest possible reduction in computation time. Criteria that were difficult to implement could use more time than is saved by finding the first contact early in the search for zeros of the C-functions.

Currently, the algorithm does not use any information about how the convex polyhedra are connected to form the overall nonconvex polyhedron representing $\mathcal{A}$ or $\mathcal{B}$. It considers each pair of convex polyhedra in isolation from the other pairs. It may be possible to use information about the configuration of the convex polyhedra to improve the collision detection algorithm. For example, certain features of one of the convex polyhedra may be covered by features of another convex polyhedron. Also, it may not be possible for two features to touch without prior contact between features of another pair of convex polyhedra. In either of these cases, the algorithm need not consider the C-functions involving these features because they cannot be involved in a first contact between $\mathcal{A}$ and $\mathcal{B}$. Again, the rules for eliminating these C-functions from the zero search must be kept relatively simple.

The collision detection algorithm as implemented for the thesis can handle any polyhedral model for the two spacecraft. The performance of a collision detection procedure to be used for a particular set of spacecraft may be improved by exploiting the shapes of these vehicles. Any symmetries or other special properties of the polyhedra representing the two spacecraft could be used to derive tests similar to the dynamic spherical approximation. Special relationships between the vectors representing the features of $\mathcal{A}$ and $\mathcal{B}$ may simplify the applicability tests. Also, it may be possible to formulate the exact rules determining the regions where the set of applicable C-functions is fixed.

The new algorithm can also handle any combination of initial configuation and linear and angular velocity vectors. Any operational constraints for a particular pair of spacecraft that restrict these quantities may be used to simplify the collision detection procedure. The types of trajectories along which one spacecraft is allowed to approach the other may be limited. This type of constraint could restrict the allowable directions of the linear and angular velocity vectors at a particular configuration of the moving spacecraft. Constraints on the the magnitudes of the linear or angular velocity vectors will influence the observed behavior of the C-functions. This is especially true if the moving spacecraft is limited to relatively low rotation rates compared to its translation rates. In these cases, the C-functions will be nearly linear and less complex zero search procedures can safely be used.

The collision detection algorithm developed for this thesis will provide the exact solution only when one spacecraft is moving with constant linear

and angular velocities relative to the other spacecraft. One way to use this algorithm for trajectories with changing linear or angular velocity vectors is to execute the algorithm at equally spaced steps along the actual trajectory. The position and orientation at each step along with the current values of the linear and angular velocity vectors would be computed for the actual trajectory of the moving vehicle and used as input to the collision detection algorithm at each step. This procedure will be feasible when the linear and angular velocity vectors are slowly varying with time, as is the case between jet firings for a spacecraft using a reaction control system. The algorithm could be executed after each jet firing to get a new prediction of any collisions using the updated velocity vectors.

Executing the collision detection algorithm at selected steps along the trajectory may be prohibitively expensive in terms of computation time when the linear and angular accelerations are high. In these cases, the spacing between steps must be decreased so that the algorithm must be run more frequently. Some computation time may be saved by using the start of the next time step as the upper bound of the zero search interval for the C-functions. Then, the algorithm will only predict any contact that will occur in the current time step. Any predictions of contact beyond the current time step would be inaccurate because of the change in the values of the linear and angular velocity vectors.

A better approach for solving the collision detection problem for trajectories that are not characterized by constant linear and angular velocities is to design a new algorithm based on the equations for the new trajectory. One simple extension of the model for the relative motion of the two spacecraft is to assume translation with constant linear *acceleration* while retaining the assumption of rotation with constant angular velocity. A better model for the relative translational motion of two spacecraft uses Hill's equations or the Clohessy-Wiltshire equations. These equations take into account the effects of orbital mechanics on the motion of the two vehicles. Another model for the rotational motion of the moving spacecraft could be derived by assuming that the spacecraft experiences no external torques. The equations for the torque-free rotation of the rigid vehicle can be solved in closed form. This model for the rotational motion specifies the angular velocity vector as a function of time.

When the magnitude and direction of the angular velocity vector are both time-varying, the forms of the ACFs will be more complex than the

shifted sinusoids derived in Chapter 3. This could make the task of finding the applicability regions for the C-functions more difficult. Also, the forms of the C-functions as functions of time for any of these different assumptions for the relative motion of the spacecraft would be considerably more complex than those found in Chapter 4. The greatest difficulty in designing a new collision detection algorithm using these assumptions would be to find a zero search procedure that would apply to the new forms of the C-functions. The new forms would have to be analyzed for any special characteristics that could be used to design the search procedure. The efficiency of the zero search procedure using these characteristics would have to be compared to that of a simple procedure that samples the function at equally spaced time intervals. This simple procedure may be the only alternative if the new forms of the functions have no special properties that can be used to guide the zero search.

The C-function formulation of the configuration space approach can be used to design a collision detection algorithm as long as the two spacecraft are modelled as polyhedra. The shape of any spacecraft can be approximated to any desired degree of accuracy by using more complex nonconvex polyhedra. However, there is a penalty of increased storage space and computation time when the collision detection algorithm must work with more complex polyhedra. A more versatile model for the shapes of the vehicles would involve using curved surfaces. If this model were chosen, the C-function formulation may not be the most practical approach because the mathematics used to characterize the curved surfaces and, therefore, to define the C-functions may become quite complicated.

8.2.2 Collision Avoidance Issues

This thesis has concentrated on the *detection* of potential collisions. However, this is only the first phase in the broader problem of collision *avoidance*. The algorithm developed in this thesis can function as a part of an overall collision avoidance system for spacecraft proximity operations.

A collision avoidance system can be used as a tool for mission planning. In this case, the system is implemented as part of a simulation of the spacecraft in its orbital environment. It would aid mission planners in choosing trajectories that accomplished the mission objectives subject to the vehicle's operational constraints and the contraints imposed in order to avoid

unintentional contact with other spacecraft. For example, suppose that one of the mission objectives is to use a spacecraft's reaction control system to perform maneuvers to dock with another spacecraft. The docking process typically involves a series of jet firings separated by intervals where the two spacecraft are coasting relative to one another. A simple collision avoidance system uses the collision detection algorithm to iteratively determine an acceptable set of jet firings. The algorithm is run after each simulated jet firing to determine if any unwanted collisions take place before the next firing. The assumption of constant linear and angular velocities is a reasonable model for the motion of the spacecraft in the intervals between jet firings. A new firing is suggested for any unacceptable firings and the algorithm is then executed to check the new set of firings.

A collision avoidance system can also be implemented as part of the flight software for a spacecraft. In this case, the amount of memory and CPU time available for the system may be limited. The collision detection algorithm as well as the other parts of the system should be optimized to execute in minimal time. This is not as critical when the system is used as a planning tool because it is not required to run in real time and greater computer resources will usually be available.

Information on the shapes of the different spacecraft and their relative motion will be required whether the collision avoidance system is implemented as flight software or as part of a simulation. The database describing the relevant spacecraft must be available before any of the routines in the collision avoidance system are executed. The information describing the current configuration and relative motion of the vehicles will be based on either real or simulated sensor measurements. Typically, a navigation routine uses the data from these measurements to produce estimates of the translational and rotational states of the spacecraft with some associated errors. The question of how the errors in these estimates propagate through the collision detection algorithm should be investigated. The loss of accuracy in the predicted time of first contact due to these errors should be quantified to determine if the prediction is still acceptable.

The prediction of a collision by the first phase of the collision avoidance system will be used in the second phase. As mentioned in Chapter 1, this phase determines a strategy for avoiding the potential collision. One major issue in this second phase is the desired amount of automation in selecting this strategy. One possibility is that the collision avoidance system is im-

plemented on an autonomous vehicle. There may be a limited capability for commanding the vehicle but normally it will operate independently. In this case, the flight software must make all decisions with no human intervention. The rules for selecting a strategy to avoid a collision must be codified as part of this software.

Another possibility is that the collision avoidance system is to be used for a spacecraft that is either partially or completely controlled by a human operator. The operator may be the pilot of a manned vehicle such as the shuttle or the ground controller of an unmanned vehicle such as the OMV. The decision to take any action to prevent collisions is made by the operator. The collision avoidance system is designed to aid the operator in making this decision. A simple system may only notify the operator of an impending collision. A more sophisticated system could also suggest a possible strategy for avoiding a predicted collision. The operator would then have the option of implementing this strategy or selecting one of his or her own. It may also be desirable to have the system check any strategy selected by the operator before it is implemented to verify that it will avoid collisions.

Two other factors to be considered in designing an interactive collision avoidance system are the amount of information to be conveyed to the operator and the format for presenting this information. All essential information should be presented in a format that can be quickly and easily interpreted by the operator. For example, it may be useful to display a sketch of the two vehicles indicating which parts are predicted to collide as well as a readout of the predicted time to collision. If the collision can be avoided by a pure translation, this could be suggested on the sketch of the vehicles by an arrow indicating the appropriate direction of the velocity vector.

The second phase of the collision avoidance system must first decide when it is necessary to take some action to avoid a predicted collision. This may involve setting a bound on the value of the predicted time to collision above which no action will be taken. This bound will depend on the control authority available from the spacecraft's jets or other actuators. It may also depend on some of the operational constraints for the two spacecraft. Limitiations on the control authority may also impose another bound on the value of the predicted time to collision below which the collision is unavoidable. This bound represents the minimum time necessary for the

spacecraft to complete a maneuver to avoid the collision.

Once it has been determined that some action must be taken to avoid a collision, a particular type of action must be chosen. One strategy is to change the relative trajectory of the two spacecraft. An evasive maneuver can be performed by either one or both of the spacecraft. First, consider the case where only one of the spacecraft will perform evasive maneuvers in response to a potential collision. This active spacecraft can be identified with the moving object and the passive spacecraft can be identified with the obstacle when using the collision detection algorithm. Because the algorithm is based on the configuration space approach, it provides some information in addition to the time of first contact that may be useful in selecting an evasive maneuver. First, it can predict the features of the two polyhedra that are touching at the time of the potential collision. This information could be used to select the direction of a translation and/or rotation such that these features move farther apart instead of closer together. Second, the values of the C-functions near the time of first contact could be used to suggest a new direction for the trajectory of the configuration of the moving object that moves away from the boundary of the configuration space obstacle. This would take into account the relative motion of the other features of the two polyhedra besides the ones involved in the predicted contact. The values of the C-functions have been used in a similar manner to generate collision-free paths in systems such as the one described in reference [16].

Now, consider the possibility that both spacecraft are allowed to perform maneuvers in response to potential collisions. In this case, the two spacecraft can cooperate to determine the best way to modify their relative motion. This requires communication between the computers or operators controlling each vehicle. There is a greater amount of freedom in choosing the maneuver(s) for both spacecraft.

Another strategy for avoiding collisions is to change the shape of one or both of the spacecraft. This changes the configuration space obstacle so that configurations that were formerly in the obstacle now lie in free space. This strategy may be feasible if one of the spacecraft has movable parts such as solar panels or antennas. The knowledge of the features that touch at the time of first contact is also useful in this case to verify that the shapes can be changed to prevent the collision. Once the shape of a spacecraft has been changed, the database describing it must be modified

to reflect this change.

In the third phase, the collision avoidance system implements the strategy selected in the second phase. The implementation must be monitored to confirm that collisions have been successfully avoided. One important problem in this final phase is to determine when the threat of a particular collision is past. At this point, the system is no longer concerned with that collision.

The collision avoidance system discussed above is only used to modify an existing trajectory for a spacecraft such that potential collisions are prevented. The existing trajectory is initially planned by mission designers or operators to achieve certain goals. The next step beyond a collision avoidance system would be a high-level system for planning spacecraft proximity operations to achieve these mission goals. This system would be similar to the task planners now being designed for robots. The mission goals would be described to the planner in simple terms. The planner would then integrate rules about the operation of the spacecraft to determine the detailed maneuvers required to achieve the goals. These rules could include criteria such as avoiding collisions between the spacecraft or minimizing fuel usage for each maneuver.

References

[1] Young, K. and Bell, J., "Rendezvous and Proximity Operations – Mission Operations — Flight Design Perspective," in *Proc. of the Rendezvous and Proximity Operations Workshop*, NASA Johnson Space Center, February 1985.

[2] "Autonomous Operation," presentation by MDAC at meetings on Space Station Work Package 2, December 1985.

[3] Smith, W. and Cox, K., "Role of Rendezvous and Proximity Operations in Integrated Orbital Operations," in *Proc. of the Rendezvous and Proximity Operations Workshop*, NASA Johnson Space Center, February 1985.

[4] Henderson, D., "Formation Flying Techniques," in *Proc. of the Rendezvous and Proximity Operations Workshop*, NASA Johnson Space Center, February 1985.

[5] Kayton, M. and Fried, W. (eds.), **Avionics Navigation Systems**, John Wiley and Sons, New York, 1969, pp. 605 – 608.

[6] Boyse, J. W., " Interference Detection among Solids and Surfaces," *Comm. of the ACM*, vol. 22, no. 1, January 1979, pp 3 – 9.

[7] Lozano-Perez, T., *Spatial Planning with Polyhedral Models*, PhD thesis, MIT, 1980.

[8] Schwartz, J. and Sharir, M., *On the Piano Movers Problem I: The Case of a Two-Dimensional Rigid Polygonal Body Moving amidst Polygonal Barriers*, Courant Institute of Mathematical Sciences Report # 39, 1981.

[9] Ahuja, N., Chien, R. T., Yen, R., and Bridwell, N., "Interference Detection and Collision Avoidance among Three-Dimensional Objects," in *Proc. of 1st Annual Conference on Artificial Intelligence*, Stanford University, 1980, pp. 44 – 48.

[10] Cameron, S., "A Study of the Clash Detection Problem in Robotics," in *Proc. of IEEE Conference on Robotics and Automation*, 1985, pp. 488 – 493.

[11] Mortenson, M. E., **Geometric Modeling**, John Wiley and Sons, New York, 1985.

[12] Lozano-Perez, T. and Wesley, M. A., "An Algorithm for Planning Collision-Free Paths among Polyhedral Obstacles," *Comm. of the ACM*, vol. 22, no. 10, October 1979, pp. 560 – 570.

[13] Esterling, D. M., and VanRosendale, J., "An Intersection Algorithm for Moving Parts," in *Proc. of the NASA Symposium on Computer-Aided Geometric Modelling*, NASA Langley, March 1984, pp. 119 – 123.

[14] Lozano-Perez, T., "Spatial Planning: A Configuration Space Approach," *IEEE Trans. on Computers*, vol. C-32, no. 2, February 1983, pp. 108 – 120.

[15] Udupa, S., "Collision Detection and Avoidance in Computer Controlled Manipulators," in *Proc. of the 5th International Joint Conference on Artificial Intelligence*, MIT, 1977.

[16] Donald, B. R., *Motion Planning with Six Degrees of Freedom*, MIT Artificial Intelligence Laboratory TR – 791, 1984.

[17] Canny, J., " Collision Detection for Moving Polyhedra," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 2, March 1986, pp. 200 – 209.

[18] Arnold, V. I., **Mathematical Methods of Classical Mechanics**, Springer-Verlag, New York, 1978, *Chapter 6 Rigid Bodies*, pp. 123 – 159.

[19] Hamilton, W. R., **Elements of Quaternions**, Chelsea, New York, 1969.

[20] Branets, V. N. and Shmyglevskiy, I. P., **Application of Quaternions to Rigid Body Rotation Problems**, NASA-TT-F-15414, July 1974.

[21] Ebert, W. L., *Kinematics, Dynamics, and Estimation of Rigid Body Motion using Euler Parameters (Quaternions)*, Johns Hopkins University Applied Physics Laboratory, JHU/APL TG 1329, November 1981.

[22] Holt, R., personal communication, August 1986.

[23] Megiddo, N., " Linear Time Algorithm for Linear Programming in $\Re^3$ and Related Problems," *SIAM Journal of Computation*, vol. 12, no. 4, November 1983, pp. 759 – 776.

[24] Baer, A., Eastman, C., and Henrion, M., " Geometric Modelling - A Survey," *Computer-Aided Design*, vol. 11, no. 5, September 1979, pp. 253 – 272.

[25] Requicha, A. A. G. and Voelcker, H. B., " Solid Modeling: A Historical Summary and Contemporary Assessment," *IEEE Computer Graphics and Applications*, March 1982, pp. 9 – 24.

[26] Ostrowski, A. M., **Solution of Equations in Euclidean and Banach Spaces**, Academic Press, New York, 1973, *Chapter 3 Method of False Position* (*Regula Falsi*), pp. 27 – 37.

[27] Householder, A. S., **The Numerical Treatment of a Single Nonlinear Equation**, McGraw-Hill, New York, 1970, *Chapter 4 Local Methods* pp. 149 – 153.

[28] Ralston, A., **A First Course in Numerical Analysis**, McGraw-Hill, New York, 1965, *Chapter 8 The Solution of Nonlinear Equations* pp 318 – 328.

Appendix A

The Intervals where the Second Derivative is Zero for Case 5 C-Functions of Class II

As shown in section 5.2.2.2, the second derivative of the C-function $f(t)$ for case 5 of class II when $d_\phi \neq 0$ can be written as

$$\frac{d^2f}{dt^2} = \omega \left\{ F_2(t) \cos(\omega t + \phi_2(t)) \right\}$$

where

$$F_2(t) = \sqrt{\left[-\omega m_2 t + (2m_3 - \omega l_2)\right]^2 + \left[\omega m_3 t + (2m_2 + \omega l_3)\right]^2}$$

and

$$\begin{aligned} \cos\phi_2(t) &= \left(-\omega m_2 t + (2m_3 - \omega l_2)\right)/F_2(t) \\ \sin\phi_2(t) &= \left(+\omega m_3 t + (2m_2 + \omega l_3)\right)/F_2(t) \end{aligned}$$

This appendix identifies sixteen possible cases for the behavior of the phase angle $\phi_2(t)$. The quantities β_1, β_2, and α, defined in sections 5.2.2.2 and 5.2.2.3, are given for each case. In addition, tighter bounds on the location of the zeros of d^2f/dt^2 are derived.

A.1 Choosing the values of β_1 and β_2

The phase angle $\phi_2(t)$ is either increasing or decreasing between two constant angles β_1 and β_2 as t goes from $-\infty$ to $+\infty$. The definitions of β_1 and β_2 from section 5.2.2.2 are

$$\begin{aligned} \cos\beta_1 &= \frac{m_2}{\sqrt{m_2^2 + m_3^2}} \\ \sin\beta_1 &= \frac{-m_3}{\sqrt{m_2^2 + m_3^2}} \end{aligned} \quad \Rightarrow \quad \tan\beta_1 = \frac{-m_3}{m_2}$$

and

$$\begin{aligned} \cos\beta_2 &= \frac{-m_2}{\sqrt{m_2^2+m_3^2}} \\ \sin\beta_2 &= \frac{m_3}{\sqrt{m_2^2+m_3^2}} \end{aligned} \quad \Rightarrow \tan\beta_2 = \frac{m_3}{-m_2}$$

The quadrants containing β_1 and β_2 depend on the values of m_2 and m_3. The eight possible combinations for the values of the two coefficients m_2 and m_3 for class II are numbered as follows:

$$\begin{array}{ll} 1 & m_2 > 0\,,\ m_3 > 0 \\ 2 & m_2 > 0\,,\ m_3 < 0 \\ 3 & m_2 < 0\,,\ m_3 > 0 \\ 4 & m_2 < 0\,,\ m_3 < 0 \\ 5 & m_2 = 0\,,\ m_3 > 0 \\ 6 & m_2 = 0\,,\ m_3 < 0 \\ 7 & m_2 > 0\,,\ m_3 = 0 \\ 8 & m_2 < 0\,,\ m_3 = 0 \end{array}$$

It is always possible to choose the values of β_1 and β_2 such that the curve of $\phi_2(t)$ is continuous between them. This choice for the bounds of the quadrants containing β_1 and β_2 will depend on the value of d_ϕ as well as m_2 and m_3. The sign of the quantity $d_\phi = 2(m_2^2+m_3^2)+\omega\,(m_2 l_3 - m_3 l_2)$ determines whether $\phi_2(t)$ is increasing or decreasing. Since d_ϕ can be either positive or negative for each of the eight possible combinations of m_2 and m_3, there are a total of sixteen cases to be examined.

The cases where $d_\phi > 0$ will be denoted by a $+$ superscript. For these "+" cases, the phase angle $\phi_2(t)$ increases monotonically from β_1 to $\beta_2 = \beta_1 + \pi$. As an example, consider case 1^+ where $m_2 > 0$, and $m_3 > 0$. For this case, β_1 is located in the fourth quadrant and β_2 is located in the second quadrant. The phase angle $\phi_2(t)$ increases from β_1 to β_2, passing through values of 0 and $\pi/2$. For $\phi_2(t)$ to be continuous from β_1 to β_2, β_1 must be in the interval $(-\pi/2, 0)$ and β_2 in the interval $(\pi/2, \pi)$.

The cases where $d_\phi < 0$ will be denoted by a $-$ superscript. For these "$-$" cases, the phase angle $\phi_2(t)$ decreases monotonically from β_1 to $\beta_2 = \beta_1 - \pi$. An example of a "$-$" case is case 3^- where m_2 is negative and m_3 is positive. The two angles β_1 and β_2 are located in the third and first quadrants, respectively. The phase angle $\phi_2(t)$ decreases from β_1 through

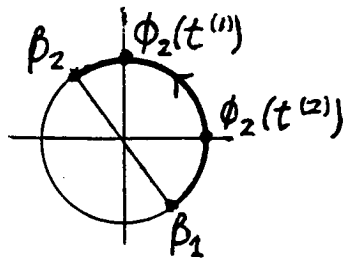
Case	β_1	β_2
1^+	$(-\pi/2, 0)$	$(\pi/2, \pi)$
2^+	$(0, \pi/2)$	$(\pi, 3\pi/2)$
3^+	$(-\pi, -\pi/2)$	$(0, \pi/2)$
4^+	$(\pi/2, \pi)$	$(3\pi/2, 2\pi)$
5^+	$-\pi/2$	$\pi/2$
6^+	$\pi/2$	$3\pi/2$
7^+	0	π
8^+	π	2π
1^-	$(3\pi/2, 2\pi)$	$(\pi/2, \pi)$
2^-	$(0, \pi/2)$	$(-\pi, -\pi/2)$
3^-	$(\pi, 3\pi/2)$	$(0, \pi/2)$
4^-	$(\pi/2, \pi)$	$(-\pi/2, 0)$
5^-	$3\pi/2$	$\pi/2$
6^-	$\pi/2$	$-\pi/2$
7^-	0	$-\pi$
8^-	π	0

Table A.1: The intervals and special values for β_1 and β_2

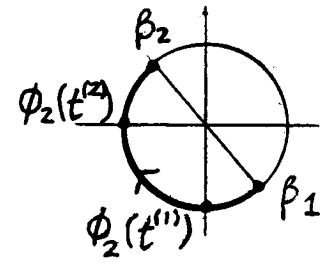
π and $\pi/2$ to β_2. If the curve of $\phi_2(t)$ is to be continuous, then β_1 must be in the interval $(\pi, 3\pi/2)$ and β_2 must be in the interval $(0, \pi/2)$.

Arguments similar to those given for cases 1^+ and 3^- govern the choice of the intervals containing β_1 and β_2 for the remaining cases where m_2 and m_3 are nonzero. The quadrants containing β_1 and β_2 are the same for the corresponding $+$ and $-$ cases. These quadrants are shown in Figure A.1 for cases $1^+ - 4^+$ and $1^- - 4^-$. This figure also the range of values of $\phi_2(t)$ as arcs of the unit circle. For the $+$ cases, the arrows pointing counterclockwise on these arcs indicate that $\phi_2(t)$ is increasing. Similarly, the arrows pointing clockwise on the arcs for the $-$ cases indicate that $\phi_2(t)$ is decreasing. The resulting intervals containing β_1 and β_2 for continuous $\phi_2(t)$ for these eight cases are shown in Table A.1.

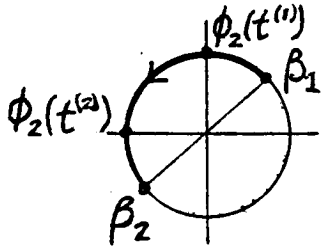
For cases 5^+, 5^-, 6^+, and 6^-, $m_2 = 0$ and $\cos\beta_1 = \cos\beta_2 = 0$. For cases 5^+ and 5^-, where $m_3 > 0$, $\sin\beta_1 = -1$ and $\sin\beta_2 = +1$. This



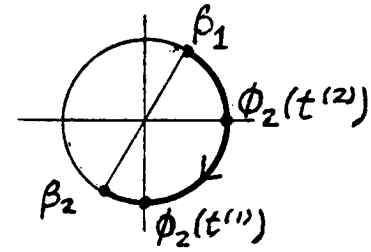
Case 1^+



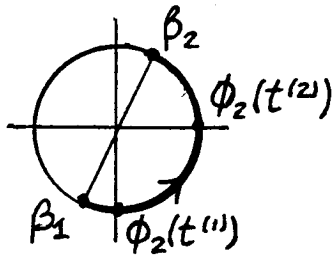
Case 1^-



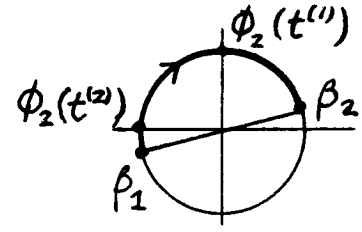
Case 2^+



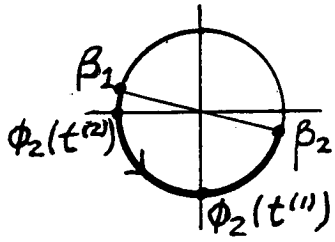
Case 2^-



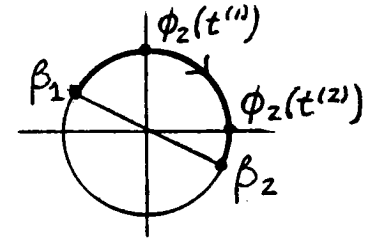
Case 3^+



Case 3^-



Case 4^+



Case 4^-

Figure A.1: The range of values of $\phi_2(t)$ for cases $1^+ - 4^+$ and $1^- - 4^-$

means that for case 5^+ $\phi_2(t)$ must increase from $\beta_1 = -\pi/2$, through 0, to $\beta_2 = \pi/2$. Conversely, for case 5^-, $\phi_2(t)$ must decrease from $\beta_1 = 3\pi/2$, through π, to $\beta_2 = \pi/2$. For cases 6^+ and 6^-, where $m_3 < 0$, $\sin\beta_1 = +1$ and $\sin\beta_2 = -1$. Therefore, for case 6^+, $\phi_2(t)$ increases from $\beta_1 = \pi/2$, through π, to $\beta_2 = 3\pi/2$. For case 6^-, $\phi_2(t)$ decreases from $\beta_1 = \pi/2$, through 0, to $\beta_2 = -\pi/2$.

Now, $\sin\beta_1 = \sin\beta_2 = 0$ for cases 7^+, 7^-, 8^+, and 8^- because $m_3 = 0$. The value of m_2 is positive for cases 7^+ and 7^- so that $\cos\beta_1 = +1$, and $\cos\beta_2 = -1$. In case 7^+, $\phi_2(t)$ increases from $\beta_1 = 0$, through $\pi/2$, to $\beta_2 = \pi$. It decreases from $\beta_1 = 0$, through $-\pi/2$, to $\beta_2 = -\pi$ in case 7^-. The coefficient m_2 is negative for cases 8^+ and 8^-. This means that $\cos\beta_1 = -1$ and $\cos\beta_2 = +1$. Thus, $\phi_2(t)$ increases from $\beta_1 = \pi$, through $3\pi/2$, to $\beta_2 = 2\pi$ for case 8^+ and decreases from $\beta_1 = \pi$, through $\pi/2$, to $\beta_2 = 0$ for case 8^-.

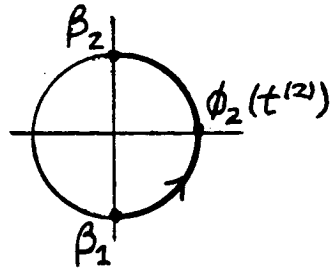
The special locations of β_1 and β_2 on the unit circle for cases 5^+ - 8^+ and cases 5^- - 8^- are shown in Figure A.2. The range of values of $\phi_2(t)$ for these cases are shown as arcs of the unit circle in this figure. As was the case for Figure A.1, the arrows pointing counterclockwise on the arcs for the + cases indicate that $\phi_2(t)$ is increasing and the arrows pointing clockwise on the arcs for the − cases indicate that $\phi_2(t)$ is decreasing. The resulting special values of β_1 and β_2 for continuous $\phi_2(t)$ in these cases are also listed in Table A.1.

A.2 Choosing the value of α

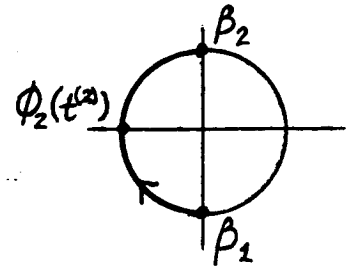
In section 5.2.2.2, the points where the second derivative of $f(t)$ is zero were shown to be the points where

$$\phi_2(t) = -\omega t + \frac{k'\pi}{2}$$

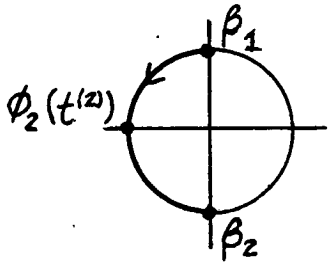
for k' an odd integer. These are the points where the curve $\phi_2(t)$ intersects one of the family of lines $y(t) = -\omega t + k'\pi/2$. The segments of these lines which lie between $y(t) = \beta_1$ and $y(t) = \beta_2 = \beta_1 \pm \pi$ divide the ωt axis into intervals of length π. A new phase angle α was introduced in section 5.2.2.3 so that the zero search could be conducted on a periodic basis. This phase angle must be chosen so that two consecutive intervals containing segments



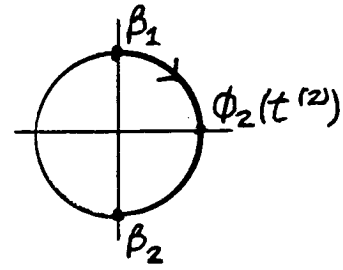
Case 5^+



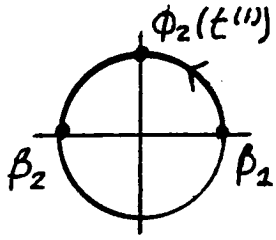
Case 5^-



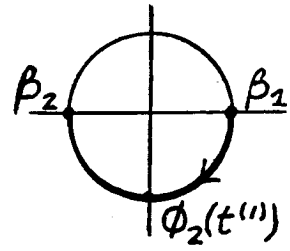
Case 6^+



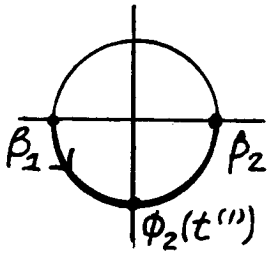
Case 6^-



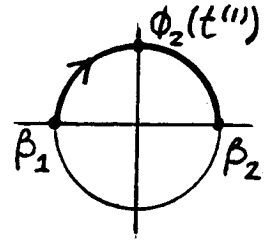
Case 7^+



Case 7^-



Case 8^+



Case 8^-

Figure A.2: The range of values of $\phi_2(t)$ for cases 5^+ – 8^+ and 5^- – 8^-

of $y(t)$ form a single cycle where the value of $\omega t + \alpha$ modulo 2π is contained in the interval $[0, 2\pi)$. The relevant intervals for locating the zeros of the second derivative of $f(t)$ then become intervals of the form $[n'\pi, (n'+1)\pi]$ in $\omega t + \alpha$. The criteria for choosing the value of α are given below.

The phase angle α is defined by considering one of the intervals containing the segments of the lines $y(t)$ for a particular odd integer k_2'. This integer is chosen to be the odd integer whose value is such that the line $y(t) = -\omega t + k_2'\pi/2$ has its y-intercept between β_1 and β_2. That is, the value of $y(0) = k_2'\pi/2$ must be between β_1 and β_2. Then, α is computed by setting $\omega t + \alpha$ equal to zero at the lower bound of the interval containing the line $y(t) = -\omega t + k_2'\pi/2$.

When ϕ_2 is increasing, the intervals of ωt containing segments of the lines $y(t)$ have the form $[-\beta_1 - \pi + k'\pi/2, -\beta_1 + k'\pi/2]$. The lower bound of the interval containing $y(t) = -\omega t + k_2'\pi/2$ is $\omega t = -\beta_1 - \pi + k_2'\pi/2$. Substituting this value of ωt in the equation $\omega t + \alpha = 0$ gives $\alpha = \beta_1 + \pi - k_2'\pi/2$.

When ϕ_2 is decreasing, the intervals of ωt containing segments of the lines $y(t)$ have the form $[-\beta_1 + k'\pi/2, -\beta_1 + \pi + k'\pi/2]$ Therefore, the lower bound of the interval containing $y(0) = -k_2'\pi/2$ is $\omega t = -\beta_1 + k_2'\pi/2$ and $\alpha = \beta_1 - k_2'\pi/2$. For these values of α, the new intervals of $\omega t + \alpha$ become $[(k' - k_2')\pi/2, (k' - k_2')\pi/2 + \pi]$. Since k' and k_2' are both odd integers, their difference $k' - k_2'$ is an even integer and the new intervals of $\omega t + \alpha$ are simply $[n'\pi, (n'+1)\pi]$ where $n' = (k' - k_2')/2$ is an integer.

The choice of k_2' as described above is ambiguous for the four cases 5^+, 5^-, 6^+, and 6^- where $m_2 = 0$. This is because the values of β_1 and β_2 are equal to $k'\pi/2$ for two different values of $k' \in \{\ldots, -3, -1, +1, +3, \ldots\}$. There are two lines $y(t) = -\omega t + k'\pi/2$ whose y-intercepts are exactly equal to β_1 and β_2, respectively. In these cases, the lines $y(t)$ divide the orginal ωt axis into intervals of the form $[n'\pi, (n'+1)\pi]$ where n' is an integer. Since the purpose of introducing the phase angle α is to convert to intervals of this form in $\omega t + \alpha$, the value of α can be set to zero in these cases.

The values of k_2' and α for the sixteen cases are listed in Table A.2.

Case	k_2'	α
1^+	1	$\beta_1 + \pi/2$
2^+	1	$\beta_1 + \pi/2$
3^+	-1	$\beta_1 + 3\pi/2$
4^+	3	$\beta_1 - \pi/2$
5^+	-	0
6^+	-	0
7^+	1	$\pi/2$
8^+	3	$\pi/2$
1^-	3	$\beta_1 - 3\pi/2$
2^-	-1	$\beta_1 + \pi/2$
3^-	1	$\beta_1 - \pi/2$
4^-	1	$\beta_1 - \pi/2$
5^-	-	0
6^-	-	0
7^-	-1	$\pi/2$
8^-	1	$\pi/2$

Table A.2: The values of k_2' and the new phase angle α

A.3 New Bounds on the Intervals where the Second Derivative is Zero

The analysis in section 5.2.2.2 showed that in most cases there is a single zero of d^2f/dt^2 in each interval $[n'\pi, (n'+1)\pi]$ of $\omega t + \alpha$. The bounds of each interval could be used as the starting values for the secant method, but it can be shown that the zero of d^2f/dt^2 must lie in a smaller part of each interval. This will yield better starting values for the iteration. These new starting values are found by considering the values of t where $\cos\phi_2(t) = 0$ and where $\sin\phi_2(t) = 0$.

Recall the definitions of $\cos\phi_2(t)$ and $\sin\phi_2(t)$ from section 5.2.2.2:

$$\begin{aligned}
\cos\phi_2(t) &= \frac{-\omega m_2 t + (2m_3 - \omega l_2)}{\sqrt{[-\omega m_2 t + (2m_3 - \omega l_2)]^2 + [\omega m_3 t + (2m_2 + \omega l_3)]^2}} \\
\sin\phi_2(t) &= \frac{\omega m_3 t + (2m_2 + \omega l_3)}{\sqrt{[-\omega m_2 t + (2m_3 - \omega l_2)]^2 + [\omega m_3 t + (2m_2 + \omega l_3)]^2}}
\end{aligned}$$

When $m_2 \neq 0$, the cosine of $\phi_2(t)$ is zero for

$$t^{(1)} = \frac{2m_3 - \omega l_2}{\omega m_2}$$

Evaluating the sine of $\phi_2(t)$ at $t = t^{(1)}$ gives

$$\sin\phi_2(t^{(1)}) = \frac{\omega m_3 t^{(1)} + 2m_2 + \omega l_3}{\left|\omega m_3 t^{(1)} + 2m_2 + \omega l_3\right|} = \pm 1$$

Thus, the value of $\phi_2(t^{(1)})$ must be equal to $k'\pi/2$ where k' is an odd integer. Recall from the definition of k_2' that $k_2'\pi/2$ is between β_1 and β_2. Since $\phi_2(t^{(1)})$ must also lie between β_1 and β_2, $\phi_2(t^{(1)}) = k_2'\pi/2$. When $m_2 = 0$, $t^{(1)}$ is undefined. In these cases, $\phi_2(t)$ approaches $k'\pi/2$ for $t \to \pm\infty$. This is confirmed by the values of β_1 and β_2 listed in Table A.1 for cases 5^+, 5^-, 6^+, and 6^-.

A similar analysis reveals that the sine of $\phi_2(t)$ is zero for

$$t^{(2)} = \frac{-2m_2 - \omega l_3}{\omega m_3}$$

when $m_3 \neq 0$. Therefore, the cosine of $\phi_2(t)$ at $t = t^{(2)}$ becomes

$$\cos\phi_2(t^{(2)}) \;=\; \frac{-\omega m_2 t^{(2)} + 2m_3 - \omega l_2}{\left|-\omega m_2 t^{(2)} + 2m_3 - \omega l_2\right|} \;=\; \pm 1$$

The value of $\phi_2(t^{(2)})$ must be equal to $n_2'\pi$ where n_2' is an integer chosen so that $n_2'\pi$ lies between β_1 and β_2. Consider the quantity $(k_2' \pm 1)\pi/2$. Since k_2' is an odd integer, $k_2' \pm 1$ is an even integer and $(k_2' \pm 1)\pi/2$ is an integral multiple of π. One of the choices of the sign will yield a value which lies between β_1 and β_2. Therefore, $\phi_2(t^{(2)})$ can be set equal to either $(k_2' + 1)\pi/2$ or $(k_2' - 1)\pi/2$. In cases 1^+, 4^+, 1^-, and 4^- where β_1 is in the fourth quadrant and β_2 is in the second quadrant, n_2' is equal to $(k_2' - 1)/2$. When β_1 is in the first quadrant and β_2 is in the third quadrant, n_2' equals $(k_2' + 1)/2$. This corresponds to cases 2^+, 3^+, 2^-, and 3^-. The quantity $t^{(2)}$ is undefined when $m_3 = 0$. In these cases, $\phi_2(t)$ approaches integral multiples of π as $t \to \pm\infty$, as verified by the values of β_1 and β_2 listed in Table A.1 for cases 7^+, 7^-, 8^+, and 8^-.

The actual values of $\phi_2(t^{(1)})$ and $\phi_2(t^{(2)})$ depend on the interval $[\beta_1, \beta_2]$. These values are shown in Table A.3. The points corresponding to $\phi_2(t^{(1)})$ and $\phi_2(t^{(2)})$ on the unit circle are indicated in Figures A.1 and A.2.

When both m_2 and m_3 are nonzero, $t^{(1)}$ and $t^{(2)}$ are well-defined and the curve of $\phi_2(t)$ must pass through both of these points. It is easy to determine which of them will occur first. From their definitions,

$$\omega m_2 t^{(1)} \;=\; 2m_3 - \omega l_2 \;\;,\;\; \omega m_3 t^{(2)} \;=\; -2m_2 - \omega l_3$$

Multiplying the equation on the left by m_3 and the equation on the right by m_2 and subtracting the resulting two equations gives

$$\omega(m_2 m_3)(t^{(1)} - t^{(2)}) = 2\left(m_2^2 + m_3^2\right) + \omega\left(m_2 l_3 - m_3 l_2\right) = d_\phi$$

Therefore, $t^{(1)} > t^{(2)}$ when $(m_2 m_3) > 0$ and $d_\phi > 0$ or when $(m_2 m_3) < 0$ and $d_\phi < 0$. These conditions correspond to cases 1^+, 4^+, 2^-, and 3^-. Conversely, $t^{(1)} < t^{(2)}$ when $(m_2 m_3) < 0$ and $d_\phi > 0$ or when $(m_2 m_3) > 0$ and $d_\phi < 0$. These are the conditions defining cases 2^+, 3^+, 1^-, 4^-. These predictions are confirmed by the locations of $\phi_2(t^{(1)})$ and $\phi_2(t^{(2)})$ on the circles in Figure A.1.

Now, $\phi_2(t)$ is continuously increasing or decreasing between β_1 and β_2. Therefore, the time axis can be divided into three intervals using $t^{(1)}$ and

Case	$\phi_2(t^{(1)}) = k_2'\pi/2$	$\phi_2(t^{(2)}) = n_2'\pi$
1^+	$\pi/2$	0
2^+	$\pi/2$	π
3^+	$-\pi/2$	0
4^+	$3\pi/2$	π
5^+	–	0
6^+	–	π
7^+	$\pi/2$	–
8^+	$3\pi/2$	–
1^-	$3\pi/2$	π
2^-	$-\pi/2$	0
3^-	$\pi/2$	π
4^-	$\pi/2$	0
5^-	–	π
6^-	–	0
7^-	$-\pi/2$	–
8^-	$\pi/2$	–

Table A.3: The values of $\phi_2(t^{(1)})$ and $\phi_2(t^{(2)})$

$t^{(2)}$. There are corresponding restrictions on the value of $\phi_2(t)$ in these intervals. First, consider cases 1^+, 4^+, 2^-, and 3^- where $t^{(2)} < t^{(1)}$. In these cases, the value of $\phi_2(t)$ lies between β_1 and $\phi_2(t^{(2)}) = n_2'\pi$ for $t < t^{(2)}$. The value of $\phi_2(t)$ lies between $\phi_2(t^{(2)}) = n_2'\pi$ and $\phi_2(t^{(1)}) = k_2'\pi/2$ for $t^{(2)} < t < t^{(1)}$. The value of $\phi_2(t)$ lies between $\phi_2(t^{(1)}) = k_2'\pi/2$ and β_2 for $t > t^{(1)}$. Next, consider the cases 1^-, 4^-, 2^+, and 3^+ where $t^{(1)} < t^{(2)}$. The value of $\phi_2(t)$ lies between β_1 and $\phi_2(t^{(1)}) = k_2'\pi/2$ for $t < t^{(1)}$. The value of $\phi_2(t)$ lies between $\phi_2(t^{(1)}) = k_2'\pi/2$ and $\phi_2(t^{(2)}) = n_2'\pi$ for $t^{(1)} < t < t^{(2)}$. The value of $\phi_2(t)$ lies between $\phi_2(t^{(2)}) = n_2'\pi$ and β_2 for $t > t^{(2)}$.

When $m_2 = 0$, $t^{(1)}$ is undefined. There are only two conditions on the value of $\phi_2(t)$ in this case. The value of $\phi_2(t)$ is between β_1 and $\phi_2(t^{(2)}) = n_2'\pi$ when $t < t^{(2)}$. The value of $\phi_2(t)$ is between $\phi_2(t^{(2)}) = n_2'\pi$ and β_2 when $t > t^{(2)}$. Similarly, there are only two conditions on the value of $\phi_2(t)$ when $m_3 = 0$ and $t^{(2)}$ is undefined. The value of $\phi_2(t)$ is between β_1 and $\phi_2(t^{(1)}) = k_2'\pi/2$ when $t < t^{(1)}$. It is between $\phi_2(t^{(1)}) = k_2'\pi/2$ and β_2 when $t > t^{(1)}$.

The bounds on the value of $\phi_2(t)$ generated using $t^{(1)}$ and $t^{(2)}$ can be combined with bounds on the value of $y(t)$ in a single interval of $\omega t + \alpha$ to find the part of the interval where $\phi_2(t)$ can intersect $y(t)$. Cases 1^+ - 4^+ and 1^- - 4^- will be discussed in section A.3.1. Two sets of restrictions are presented for these cases. The first set is derived using only the conditions imposed on $\phi_2(t)$ and $y(t)$ by $t^{(1)}$, $\phi_2(t^{(1)})$. The value of $y(t)$ passes through $\phi_2(t^{(1)}) = k_2'\pi/2$ at $\omega t = (k' - k_2')\pi/2$. By the definition of n', this is the point where $\omega t + \alpha = n'\pi + \alpha$. The interval $[n'\pi, (n'+1)\pi]$ can be divided into two parts at the point $n'\pi + \alpha$. The conditions on the value of $y(t)$ in these two parts of the interval are compared to the conditions on the value of $\phi_2(t)$ to give new limits on the location of the intersection point of $\phi_2(t)$ and $y(t)$.

The second set of restrictions is derived by using the conditions imposed on $\phi_2(t)$ and $y(t)$ by *both* $t^{(1)}$, $\phi_2(t^{(1)})$ *and* $t^{(2)}$, $\phi_2(t^{(2)})$. The segment of the line $y(t)$ passes through $\phi_2(t^{(2)}) = n_2'\pi$ at $\omega t = (k' - k_2' \pm 1)\pi/2$. This is the point where $\omega t + \alpha = n'\pi + \alpha \pm \pi/2$. The interval can be subdivided into three parts at the two points $n'\pi + \alpha$ and $n'\pi + \alpha \pm \pi/2$. A comparison of the conditions for the value of $y(t)$ in these three parts of the interval and the conditions on the value of $\phi_2(t)$ results in another set of limits on the part of an interval where the intersection of $\phi_2(t)$ and $y(t)$ can occur. These limits are tighter bounds on the location of the intersection point than the

limits that are derived using only $t^{(1)}$, $\phi_2(t^{(1)})$.

The special cases 5^+, 5^-, 6^+, and 6^- where $m_2 = 0$ and 7^+, 7^-, 8^+, and 8^- where $m_3 = 0$ are discussed in section A.3.2. In these cases, only $t^{(1)}$, $\phi_2(t^{(1)})$ or $t^{(2)}$, $\phi_2(t^{(2)})$ are available to derive the new bounds on the location of the zero of d^2f/dt^2. When $m_2 = 0$, $\alpha = 0$ and β_1 and β_2 are equal to consecutive odd multiples of $\pi/2$. Therefore, the segment of the line $y(t) = -\omega t + k'\pi/2$ between β_1 and β_2 reaches the value of $n'_2\pi$ at the midpoint of the interval $[n'\pi, (n'+1)\pi]$ where $\omega t = n'\pi + \pi/2$. For the cases where $m_3 = 0$, β_1 and β_2 are equal to consecutive integral multiples of π and the new phase angle α is equal to $\pi/2$. The segment of the line $y(t) = -\omega t + k'\pi/2$ between β_1 and β_2 passes through $k'_2\pi/2$ when $\omega t + \alpha = n'\pi + \alpha = n'\pi + \pi/2$. This is also the midpoint of the interval $[n'\pi, (n'+1)\pi]$. Thus, the intervals are divided into two equal parts. The conditions on the value of $y(t)$ in these two parts are compared with the conditions on the value of $\phi_2(t)$ to generate restrictions on the location of the point of intersection between $\phi_2(t)$ and $y(t)$.

The final collision detection algorithm implemented for this thesis uses the first set of restrictions based on $t^{(1)}$ and $\phi_2(t^{(1)})$ for cases $1^+ - 4^+$ and $1^- - 4^-$. This simplified the programming by allowing the same routine to be used to find the zeros of the second derivative of $f(t)$ in these cases and in cases 7^+, 7^-, 8^+, and 8^- . This routine is also be used for cases 5^+, 5^-, 6^+, and 6^- by substituting the starting values based on $t^{(2)}$ and $\phi_2(t^{(2)})$ for those based on $t^{(1)}$ and $\phi_2(t^{(1)})$. If the second set of restrictions based on both $t^{(1)}$, $\phi_2(t^{(1)})$ and $t^{(2)}$, $\phi_2(t^{(2)})$ were implemented for cases $1^+ - 4^+$ and $1^- - 4^-$, different routines would be required for handling these cases and the special cases 5^+, 5^-, 6^+, 6^-, 7^+, 7^-, 8^+, and 8^- .

The discussion in sections A.3.1 and A.3.2 assumes that only one intersection of the $\phi_2(t)$ curve and the line $y(t)$ takes place in an interval $[n'\pi, (n'+1)\pi]$ of $\omega t + \alpha$. The conditions under which multiple intersections can take place in a single interval and the starting values for the iteration in this interval were given in section 5.2.2.2.

The format is identical for describing each different set of starting values. First, the criteria that determine when the set may be used are stated. Next, the behavior of the line $y(t)$ in a single interval $[n'\pi, (n'+1)\pi]$ of $\omega t + \alpha$ is described. Then, the rules governing the choice of the starting values in this interval are enumerated. A figure is included that shows the behavior of $y(t)$ and $\phi_2(t)$ for each case. The intervals of $\omega t + \alpha$ in the

figures are marked with numbers corresponding to the different rules for the intersection point of $\phi_2(t)$ and $y(t)$ for that case.

A.3.1 The Cases Where $m_2 \neq 0$ and $m_3 \neq 0$

A.3.1.1 The New Bounds Using $t^{(1)}$ Alone

This section derives the starting values for the iteration to find the zero of d^2f/dt^2 from the restrictions imposed by $t^{(1)}$ and $\phi_2(t^{(1)})$ alone. As explained above, these starting values are used by the collision detection algorithm implemented for this thesis. The different starting values for the cases where $d_\phi > 0$ and $d_\phi < 0$ are listed below.

Cases $1^+ - 4^+$ For these four cases, $\phi_2(t)$ is increasing. The line segment $y(t)$ decreases from $\beta_2 = \beta_1 + \pi$ at $\omega t + \alpha = n'\pi$, through $k_2'\pi/2$ at $\omega t + \alpha = n'\pi + \alpha$, to β_1 at $\omega t + \alpha = (n'+1)\pi$. The limits on the location of the zero of d^2f/dt^2 in the interval $[n'\pi, (n'+1)\pi]$ of $\omega t + \alpha$ are:

1. If $\omega t^{(1)} + \alpha > n'\pi + \alpha$, then the intersection point of $\phi_2(t)$ and $y(t)$ must occur when $\omega t + \alpha \in \left(n'\pi + \alpha, \min[\omega t^{(1)} + \alpha, (n'+1)\pi]\right)$. Between these two points, the curve of $\phi_2(t)$ is increasing from β_1 to $k_2'\pi/2$ and the line $y(t)$ is decreasing from $k_2'\pi/2$ to β_1.

2. If $\omega t^{(1)} + \alpha < n'\pi + \alpha$, then the intersection point of $\phi_2(t)$ and $y(t)$ must occur when $\omega t + \alpha \in \left(\max[n'\pi, \omega t^{(1)} + \alpha], n'\pi + \alpha\right)$. Between these two points, the curve of $\phi_2(t)$ is increasing from $k_2'\pi/2$ to β_2 and the line $y(t)$ is decreasing from β_2 to $k_2'\pi/2$.

3. If $\omega t^{(1)} + \alpha = n'\pi + \alpha$, then the intersection of $\phi_2(t)$ and $y(t)$ occurs at $t = t^{(1)}$. Both $\phi_2(t^{(1)})$ and $y(t^{(1)})$ are equal to $k_2'\pi/2$.

An example of the behavior of $\phi_2(t)$ for these cases is shown in Figure A.3.

Cases $1^- - 4^-$ For these four cases, $\phi_2(t)$ is decreasing. The line segment $y(t)$ decreases from β_1 at $\omega t + \alpha = n'\pi$, through $k_2'\pi/2$ at $\omega t + \alpha = n'\pi + \pi/2$, to $\beta_2 = \beta_1 - \pi$ at $\omega t + \alpha = (n'+1)\pi$. The limits on the location of the zero of d^2f/dt^2 in the interval $[n'\pi, (n'+1)\pi]$ of $\omega t + \alpha$ are:

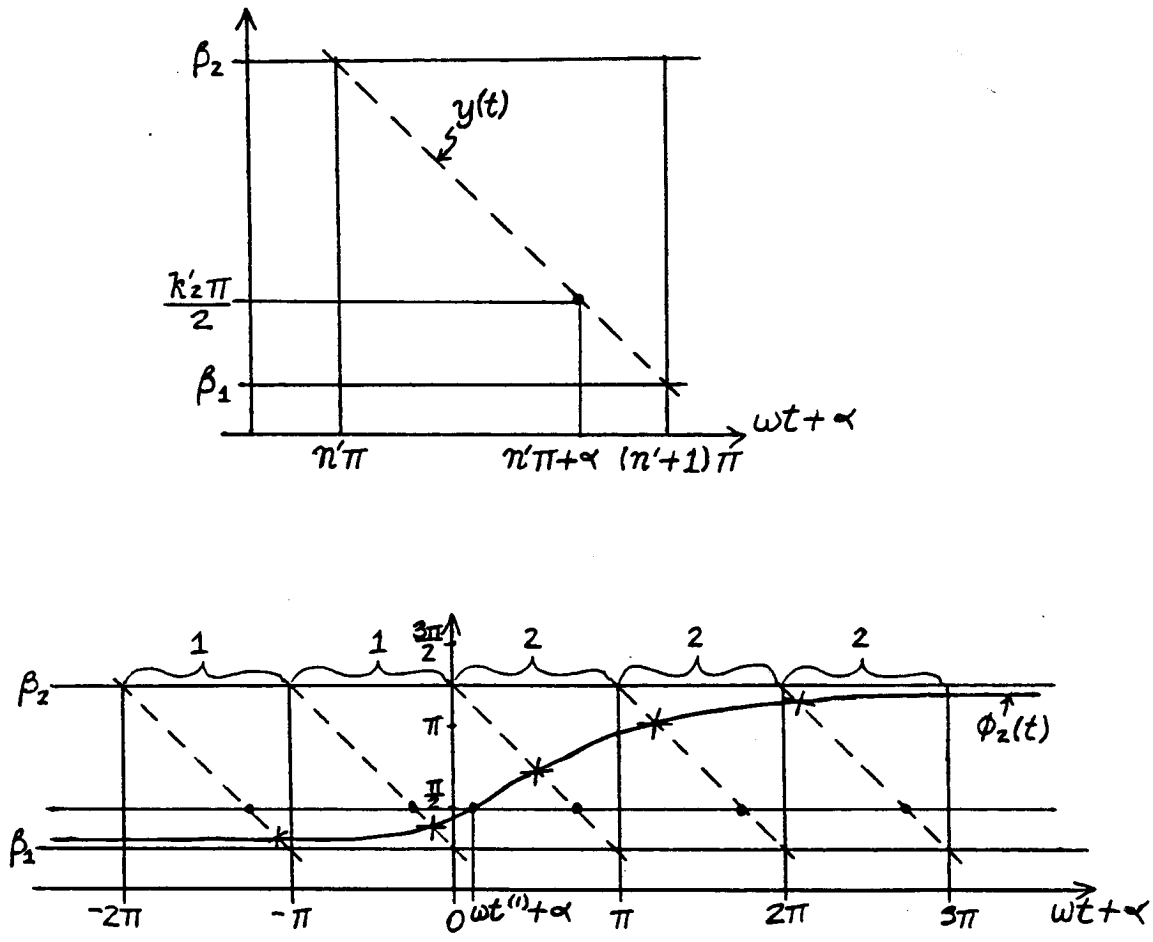


Figure A.3: The starting values for cases $1^+ - 4^+$ using only $t^{(1)}$, $\phi_2(t^{(1)})$

1. If $\omega t^{(1)} + \alpha > n'\pi + \alpha$, then the intersection point of $\phi_2(t)$ and $y(t)$ must occur when $\omega t + \alpha \in (n'\pi, n'\pi + \alpha)$. Between these two points, the curve of $\phi_2(t)$ and the line $y(t)$ are decreasing from β_1 to $k_2'\pi/2$.

2. If $\omega t^{(1)} + \alpha < n'\pi + \alpha$, then the intersection point of $\phi_2(t)$ and $y(t)$ must occur when $\omega t + \alpha \in (n'\pi + \alpha, (n'+1)\pi)$. Between these two points, the curve of $\phi_2(t)$ and the line $y(t)$ are decreasing from $k_2'\pi/2$ to β_2.

3. If $\omega t^{(1)} + \alpha = n'\pi + \alpha$, the intersection $\phi_2(t)$ and $y(t)$ occurs at $t = t^{(1)}$. Both $\phi_2(t^{(1)})$ and $y(t^{(1)})$ are equal to $k_2'\pi/2$.

An example of the behavior of $\phi_2(t)$ for these cases is shown in Figure A.4.

A.3.1.2 The New Bounds Using $t^{(1)}$ and $t^{(2)}$

Now, consider the restrictions imposed by using both $t^{(1)}$, $\phi_2(t^{(1)})$, and $t^{(2)}$, $\phi_2(t^{(2)})$. By adding the conditions on the value of $\phi_2(t)$ and $y(t)$ arising from $t^{(2)}$, $\phi_2(t^{(2)})$, tighter bounds can be placed on the location of the zero of d^2f/dt^2. The zero can now be confined to one of three parts of each interval $[n'\pi, (n'+1)\pi]$ of $\omega t + \alpha$ as opposed to the two parts that were obtained using $t^{(1)}$ and $\phi_2(t^{(1)})$ alone.

There are four possibilities depending on the values of d_ϕ, $t^{(1)}$, and $t^{(2)}$. The starting values for the iteration for the zero of d^2f/dt^2 for each of these possibilities are listed in the following paragraphs. As mentioned previously, these starting values were not used by the algorithm implemented for this thesis for simplicity of programming.

Cases 1^+ and 4^+ For these two cases, $\phi_2(t)$ is increasing and $t^{(1)} > t^{(2)}$. In the interval $[n'\pi, (n'+1)\pi]$, the line segment $y(t)$ decreases from $\beta_2 = \beta_1 + \pi$ at $\omega t + \alpha = n'\pi$, through $k_2'\pi/2$ at $\omega t + \alpha = n'\pi + \alpha$ and $n_2'\pi = (k_2' - 1)\pi/2$ at $\omega t + \alpha = n'\pi + \alpha + \pi/2$, to β_1 at $\omega t + \alpha = (n'+1)\pi$. The following limits can be placed on the location of the zero of d^2f/dt^2 in this interval:

1. If $\omega t^{(2)} + \alpha > n'\pi + \alpha + \pi/2$, then the intersection point of $\phi_2(t)$ and $y(t)$ must occur when $\omega t + \alpha \in \left(n'\pi + \alpha + \pi/2, \max[(n'+1)\pi, \omega t^{(2)} + \alpha]\right)$. Between these two points, the curve of $\phi_2(t)$ is increasing from β_1 to $n_2'\pi$ and the line $y(t)$ is decreasing from $n_2'\pi$ to β_1.

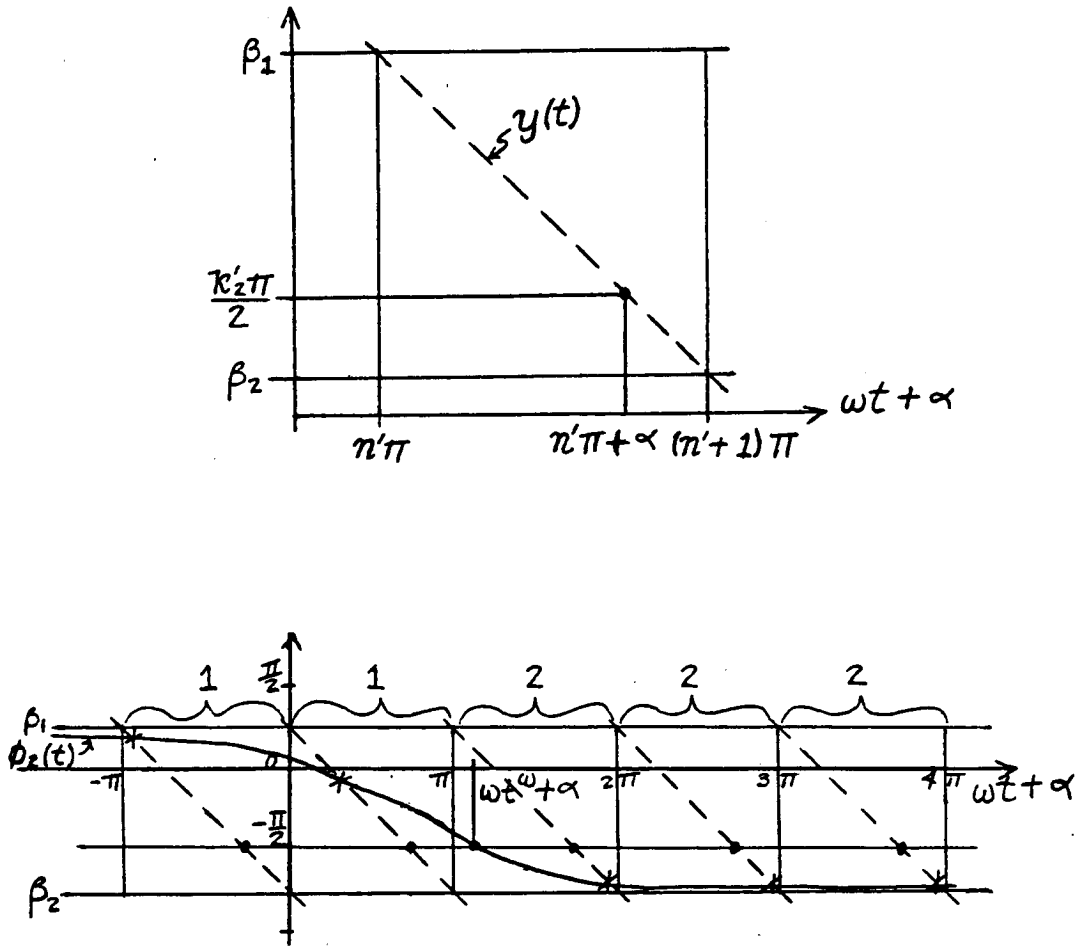


Figure A.4: The starting values for cases $1^- - 4^-$ using only $t^{(1)}$, $\phi_2(t^{(1)})$

2. If $\omega t^{(1)} + \alpha < n'\pi + \alpha$, then the intersection point of $\phi_2(t)$ and $y(t)$ must occur when $\omega t + \alpha \in \left(\max[n'\pi, \omega t^{(1)} + \alpha], n'\pi + \alpha\right)$. Between these two points, the curve of $\phi_2(t)$ is increasing from $k_2'\pi/2$ to β_2 and the line $y(t)$ is decreasing from β_2 to $k_2'\pi/2$.

3. If $\omega t^{(2)} + \alpha < n'\pi + \alpha + \pi/2$ and $\omega t^{(1)} + \alpha > n'\pi + \alpha$, then the intersection point of $\phi_2(t)$ and $y(t)$ must occur when

$$\omega t + \alpha \in \left(\max[n'\pi + \alpha, \omega t^{(2)} + \alpha], \min[n'\pi + \alpha + \pi/2, \omega t^{(1)} + \alpha]\right)$$

   Between these two points, the curve of $\phi_2(t)$ is increasing from $n_2'\pi$ to $k_2'\pi/2$ and the line $y(t)$ is decreasing from $k_2'\pi/2$ to $n_2'\pi$.

4. If $\omega t^{(1)} + \alpha = n'\pi + \alpha$, then the intersection point of $\phi_2(t)$ and $y(t)$ is $t = t^{(1)}$. Both $\phi_2(t^{(1)})$ and $y(t^{(1)})$ are equal to $k_2'\pi/2$.

5. If $\omega t^{(2)} + \alpha = n'\pi + \alpha + \pi/2$, then the intersection point of $\phi_2(t)$ and $y(t)$ is $t = t^{(2)}$. Both $\phi_2(t^{(2)})$ and $y(t^{(2)})$ are equal to $n_2'\pi$.

An example of the behavior of $\phi_2(t)$ for these cases is shown in Figure A.5.

Cases 2^+ and 3^+ For these two cases, $\phi_2(t)$ is increasing and $t^{(2)} > t^{(1)}$. In the interval $[n'\pi, (n'+1)\pi]$, the line segment $y(t)$ decreases from $\beta_2 = \beta_1 + \pi$ at $\omega t + \alpha = n'\pi$, through $n_2'\pi = (k_2' + 1)\pi/2$ at $\omega t + \alpha = n'\pi + \alpha - \pi/2$ and $k_2'\pi/2$ at $\omega t + \alpha = n'\pi + \alpha$, to β_1 at $\omega t + \alpha = (n'+1)\pi$. In these two cases, the possible limits for the location of the zero of d^2f/dt^2 in a single interval are:

1. If $\omega t^{(1)} + \alpha > n'\pi + \alpha$, then the intersection point of $\phi_2(t)$ and $y(t)$ must occur when $\omega t + \alpha \in \left(n'\pi + \alpha, \min[(n'+1)\pi, \omega t^{(1)} + \alpha]\right)$. Between these two points, the curve of $\phi_2(t)$ is increasing from β_1 to $k_2'\pi/2$ and the line $y(t)$ is decreasing from $k_2'\pi/2$ to β_1.

2. If $\omega t^{(2)} + \alpha < n'\pi + \alpha - \pi/2$, then the intersection point of $\phi_2(t)$ and $y(t)$ must occur when $\omega t + \alpha \in \left(\max[n'\pi, \omega t^{(2)} + \alpha], n'\pi + \alpha - \pi/2\right)$. Between these two points, the curve of $\phi_2(t)$ is increasing from $n_2'\pi$ to β_2 and the line $y(t)$ is decreasing from β_2 to $n_2'\pi$.

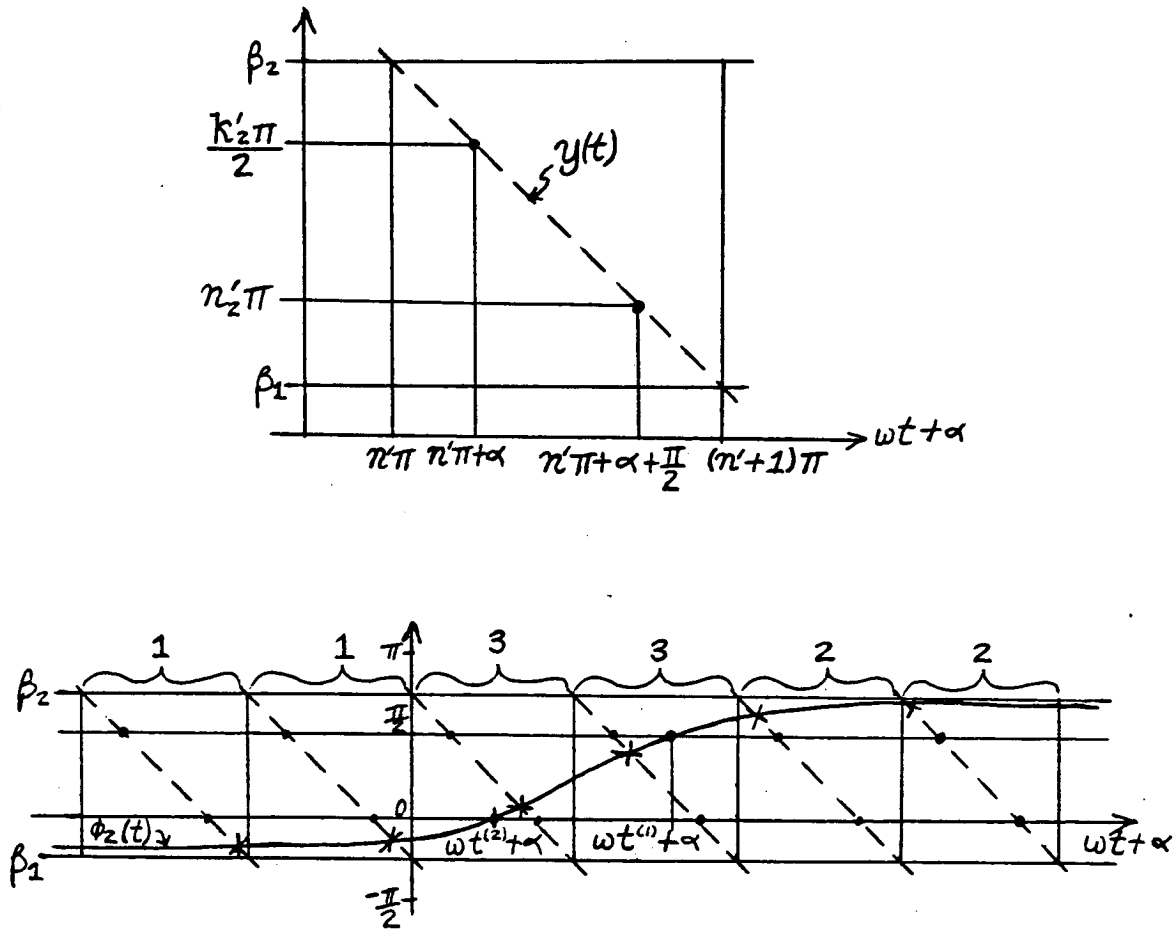


Figure A.5: The starting values for cases 1^+ and 4^+ using $t^{(1)}$, $\phi_2(t^{(1)})$ and $t^{(2)}$, $\phi_2(t^{(2)})$

3. If $\omega t^{(1)} + \alpha < n'\pi + \alpha$ and $\omega t^{(2)} + \alpha > n'\pi + \alpha - \pi/2$, then the intersection point of $\phi_2(t)$ and $y(t)$ must occur when

$$\omega t + \alpha \in \left(\max[n'\pi + \alpha - \pi/2, \omega t^{(1)} + \alpha], \min[n'\pi + \alpha, \omega t^{(2)} + \alpha]\right)$$

Between these two points, the curve of $\phi_2(t)$ is increasing from $k_2'\pi/2$ to $n_2'\pi$ and the line $y(t)$ is decreasing from $n_2'\pi$ to $k_2'\pi/2$.

4. If $\omega t^{(1)} + \alpha = n'\pi + \alpha$, then the intersection point of $\phi_2(t)$ and $y(t)$ is $t = t^{(1)}$. Both $\phi_2(t^{(1)})$ and $y(t^{(1)})$ are equal to $k_2'\pi/2$.

5. If $\omega t^{(2)} + \alpha = n'\pi + \alpha - \pi/2$, then the intersection point of $\phi_2(t)$ and $y(t)$ is $t = t^{(2)}$. Both $\phi_2(t^{(2)})$ and $y(t^{(2)})$ are equal to $n_2'\pi$.

An example of the behavior of $\phi_2(t)$ for these cases is shown in Figure A.6.

Cases 1^- and 4^- For these two cases, $\phi_2(t)$ is decreasing and $t^{(2)} > t^{(1)}$. In the interval $[n'\pi, (n'+1)\pi]$, the line segment $y(t)$ decreases from β_1 at $\omega t + \alpha = n'\pi$, through $k_2'\pi/2$ at $\omega t + \alpha = n'\pi + \alpha$ and $n_2'\pi = (k_2' - 1)\pi/2$ at $\omega t + \alpha = n'\pi + \alpha + \pi/2$, to $\beta_2 = \beta_1 - \pi$ at $\omega t + \alpha = (n'+1)\pi$. The limits for the location of the zero of d^2f/dt^2 in a single interval are:

1. If $\omega t^{(1)} + \alpha > n'\pi + \alpha$, then the intersection point of $\phi_2(t)$ and $y(t)$ must occur when $\omega t + \alpha \in (n'\pi, n'\pi + \alpha)$ Between these two points, the curve of $\phi_2(t)$ and the line $y(t)$ are decreasing from β_1 to $k_2'\pi/2$.

2. If $\omega t^{(2)} + \alpha < n'\pi + \alpha + \pi/2$, then the intersection point of $\phi_2(t)$ and $y(t)$ must occur when $\omega t + \alpha \in (n'\pi + \alpha + \pi/2, (n'+1)\pi)$. Between these two points, the curve of $\phi_2(t)$ and the line $y(t)$ are decreasing from $n_2'\pi$ to β_2.

3. If $\omega t^{(1)} + \alpha < n'\pi + \alpha$ and $\omega t^{(2)} + \alpha > n'\pi + \alpha + \pi/2$, then the intersection point of $\phi_2(t)$ and $y(t)$ must occur when

$$\omega t + \alpha \in (n'\pi + \alpha, n'\pi + \alpha + \pi/2)$$

Between these two points, the curve of $\phi_2(t)$ and the line $y(t)$ are decreasing from $k_2'\pi/2$ to $n_2'\pi$.

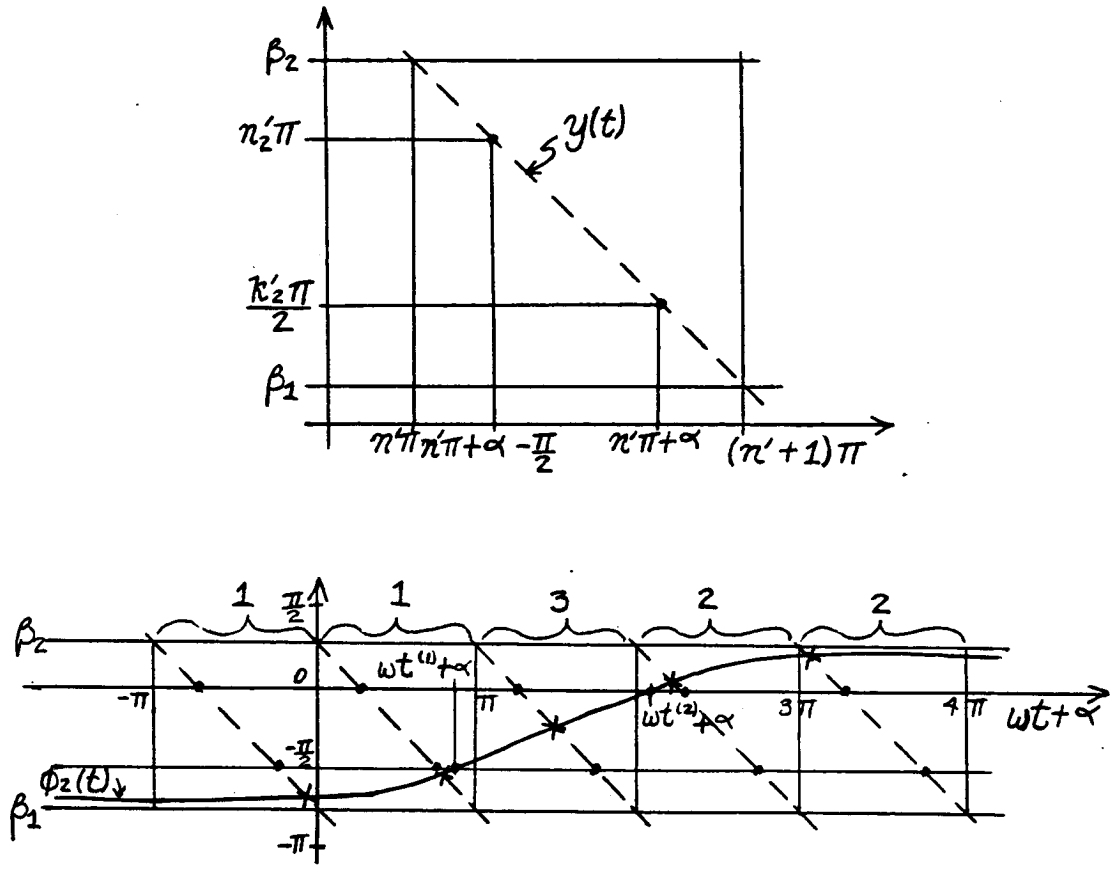


Figure A.6: The starting values for cases 2^+ and 3^+ using $t^{(1)}$, $\phi_2(t^{(1)})$ and $t^{(2)}$, $\phi_2(t^{(2)})$

4. If $\omega t^{(1)} + \alpha = n'\pi + \alpha$, then the intersection point of $\phi_2(t)$ and $y(t)$ is $t = t^{(1)}$. Both $\phi_2(t^{(1)})$ and $y(t^{(1)})$ are equal to $k_2'\pi/2$.

5. If $\omega t^{(2)} + \alpha = n'\pi + \alpha + \pi/2$, then the intersection point of $\phi_2(t)$ and $y(t)$ is $t = t^{(2)}$. Both $\phi_2(t^{(2)})$ and $y(t^{(2)})$ are equal to $n_2'\pi$.

An example of the behavior of $\phi_2(t)$ for these cases is shown in Figure A.7.

Cases 2^- and 3^- For these cases, $\phi_2(t)$ is decreasing and $t^{(1)} > t^{(2)}$. In the interval $[n'\pi, (n'+1)\pi]$, the line segment $y(t)$ decreases from β_1 at $\omega t + \alpha = n'\pi$, through $n_2'\pi = (k_2'+1)\pi/2$ at $\omega t + \alpha = n'\pi + \alpha - \pi/2$ and $k_2'\pi/2$ at $\omega t + \alpha = n'\pi + \alpha$, to $\beta_2 = \beta_1 - \pi$ at $\omega t + \alpha = (n'+1)\pi$. In these two cases, the possible limits for the zero of d^2f/dt^2 in a single interval are:

1. If $\omega t^{(2)} + \alpha > n'\pi + \alpha - \pi/2$, then the intersection point of $\phi_2(t)$ and $y(t)$ must occur when $\omega t + \alpha \in (n'\pi, n'\pi + \alpha - \pi/2)$. Between these two points, the curve of $\phi_2(t)$ and the line $y(t)$ are decreasing from β_1 to $n_2'\pi$.

2. If $\omega t^{(1)} + \alpha < n'\pi + \alpha$, then the intersection point of $\phi_2(t)$ and $y(t)$ must occur when $\omega t + \alpha \in (n'\pi + \alpha, (n'+1)\pi)$. Between these two points, the curve of $\phi_2(t)$ and the line $y(t)$ are decreasing from $k_2'\pi/2$ to β_2.

3. If $\omega t^{(2)} + \alpha < n'\pi + \alpha - \pi/2$ and $\omega t^{(1)} + \alpha > n'\pi + \alpha$, then the intersection point of $\phi_2(t)$ and $y(t)$ must occur when

$$\omega t + \alpha \in (n'\pi + \alpha - \pi/2, n'\pi + \alpha)$$

   Between these two points, the curve of $\phi_2(t)$ and the line $y(t)$ are decreasing from $n_2'\pi$ to $k_2'\pi/2$.

4. If $\omega t^{(1)} + \alpha = n'\pi + \alpha$, then the intersection point of $\phi_2(t)$ and $y(t)$ is $t = t^{(1)}$. Both $\phi_2(t^{(1)})$ and $y(t^{(1)})$ are equal to $k_2'\pi/2$.

5. If $\omega t^{(2)} + \alpha = n'\pi + \alpha - \pi/2$, then the intersection point of $\phi_2(t)$ and $y(t)$ is $t = t^{(2)}$. Both $\phi_2(t^{(2)})$ and $y(t^{(2)})$ are equal to $n_2'\pi$.

An example of the behavior of $\phi_2(t)$ for these cases is shown in Figure A.8.

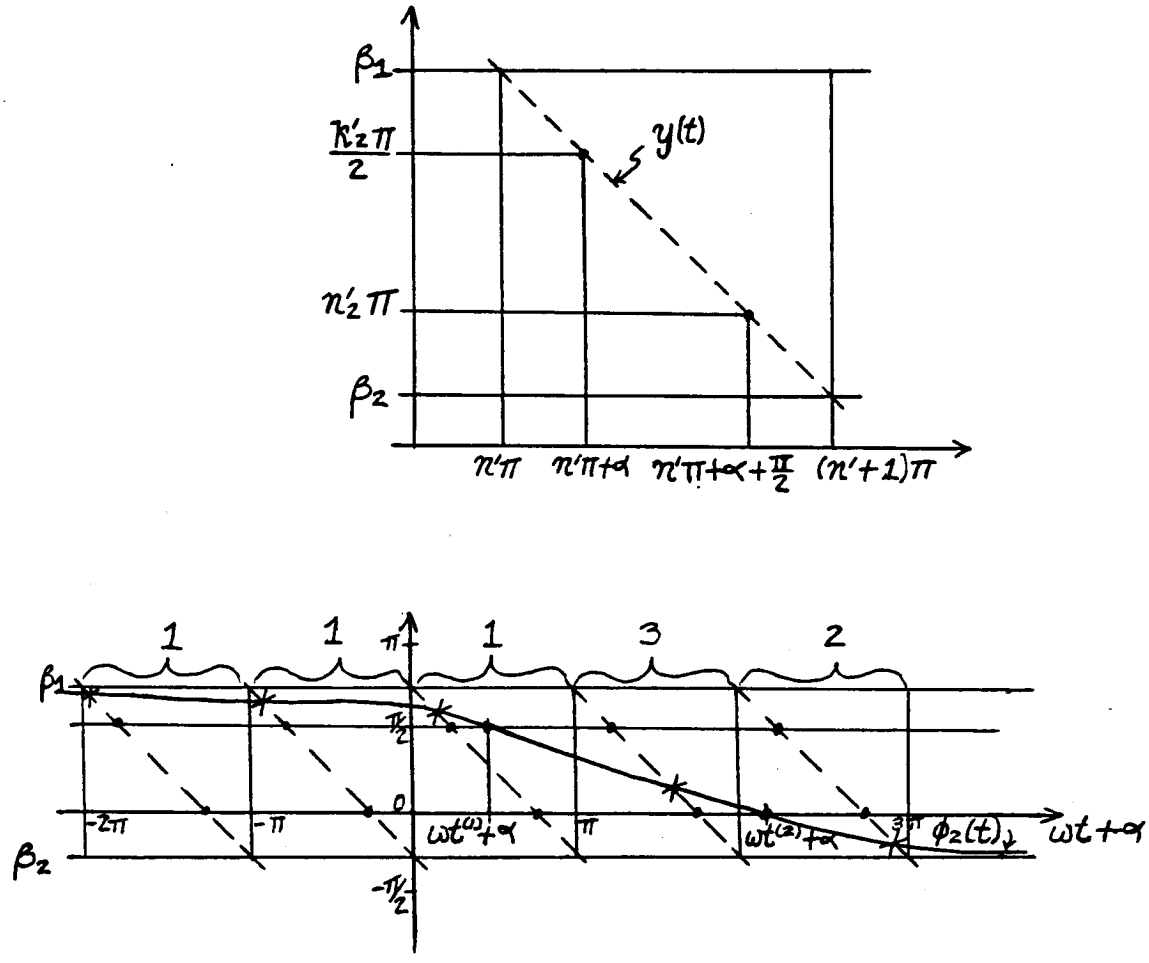


Figure A.7: The starting values for cases 1^- and 4^- using $t^{(1)}$, $\phi_2(t^{(1)})$ and $t^{(2)}$, $\phi_2(t^{(2)})$

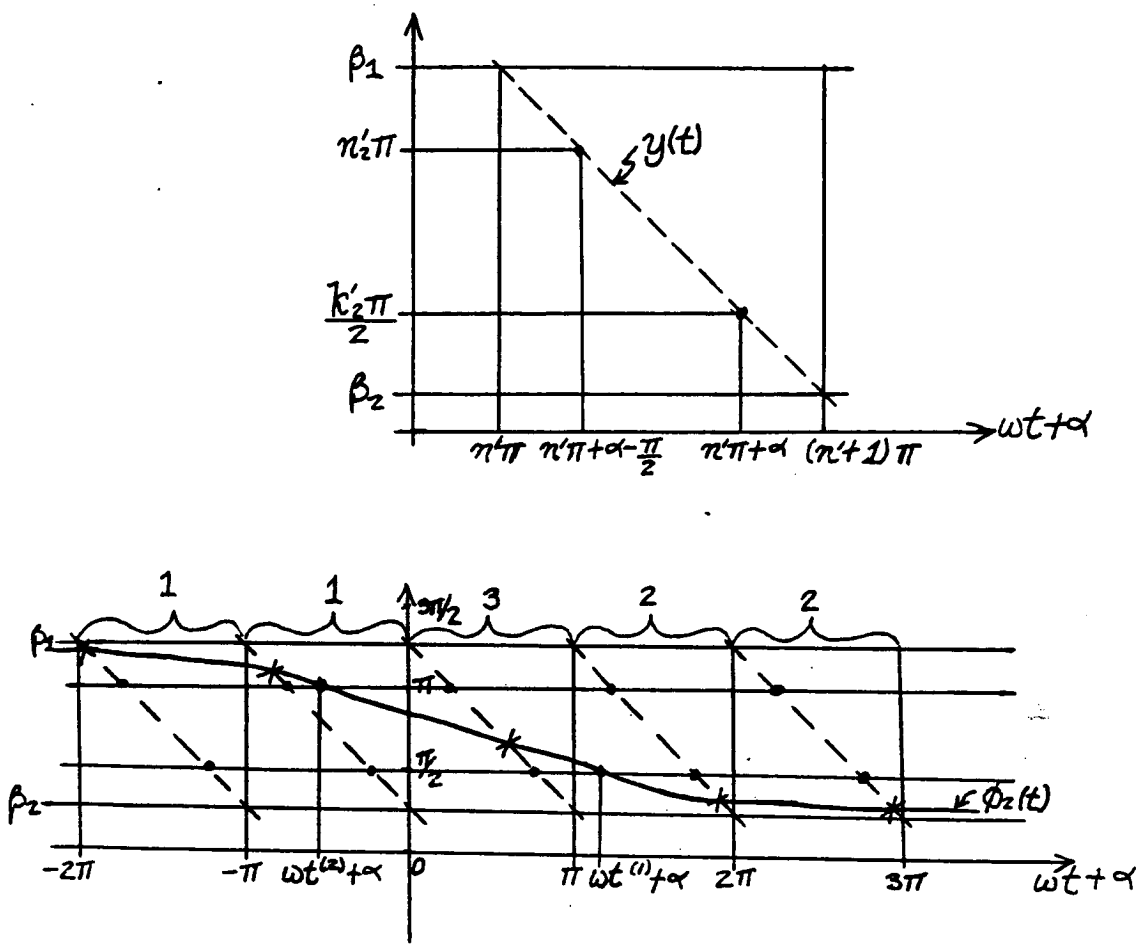


Figure A.8: The starting values for cases 2^- and 3^- using $t^{(1)}$, $\phi_2(t^{(1)})$ and $t^{(2)}$, $\phi_2(t^{(2)})$

A.3.2 The Cases where $m_2 = 0$ or $m_3 = 0$

For cases 5^+, 5^-, 6^+, and 6^-, only $t^{(2)}$ and $\phi_2(t^{(2)}) = n_2'\pi$ are available to limit the part of an interval where the zero of d^2f/dt^2 can be located. In addition, $\alpha = 0$ and the intervals become intervals of ωt. For cases 7^+, 7^-, 8^+, and 8^-, only $t^{(1)}$ and $\phi_2(t^{(1)}) = k_2'\pi/2$ are available to limit the part of an interval where the zero of d^2f/dt^2 can be located and $\alpha = \pi/2$. These cases are presented in the following sections according to whether $\phi_2(t)$ is increasing or decreasing. The limits listed in these sections are similar to those presented in section A.3.1.1.

A.3.2.1 Cases 5^+ and 6^+

For the two cases 5^+ and 6^+ where $\phi_2(t)$ is increasing, the line segment $y(t)$ decreases from $\beta_2 = \beta_1 + \pi$ at $\omega t = n'\pi$, through $n_2'\pi$ at $\omega t = n'\pi + \pi/2$, to β_1 at $\omega t = (n'+1)\pi$. The limits on the location of the zero of d^2f/dt^2 in the interval $[n'\pi, (n'+1)\pi]$ of ωt are:

1. If $\omega t^{(2)} > n'\pi + \pi/2$, then the intersection point of $\phi_2(t)$ and $y(t)$ must occur when $\omega t \in \left(n'\pi + \pi/2, \min[\omega t^{(2)}, (n'+1)\pi]\right)$. Between these two points, the curve of $\phi_2(t)$ is increasing from β_1 to $n_2'\pi$ and the line $y(t)$ is decreasing from $n_2'\pi$ to β_1.

2. If $\omega t^{(2)} < n'\pi + \pi/2$, then the intersection point of $\phi_2(t)$ and $y(t)$ must occur when $\omega t \in \left(\max[n'\pi, \omega t^{(2)}], n'\pi + \pi/2\right)$. Between these two points, the curve of $\phi_2(t)$ is increasing from $n_2'\pi$ to β_2 and the line $y(t)$ is decreasing from β_2 to $n_2'\pi$.

3. If $\omega t^{(2)} = n'\pi + \pi/2$, then the intersection point of $\phi_2(t)$ and $y(t)$ is $t = t^{(2)}$. Both $\phi_2(t^{(2)})$ and $y(t^{(2)})$ are equal to $n_2'\pi$.

An example of the behavior of $\phi_2(t)$ for these cases is shown in Figure A.9.

A.3.2.2 Cases 5^- and 6^-

For the two cases 5^- and 6^- where $\phi_2(t)$ is decreasing, the line segment $y(t)$ decreases from β_1 at $\omega t = n'\pi$, through $n_2'\pi$ at $\omega t = n'\pi + \pi/2$, to $\beta_2 = \beta_1 - \pi$ at $\omega t = (n'+1)\pi$. The limits on the location of the zero of d^2f/dt^2 in the interval $[n'\pi, (n'+1)\pi]$ of ωt are:

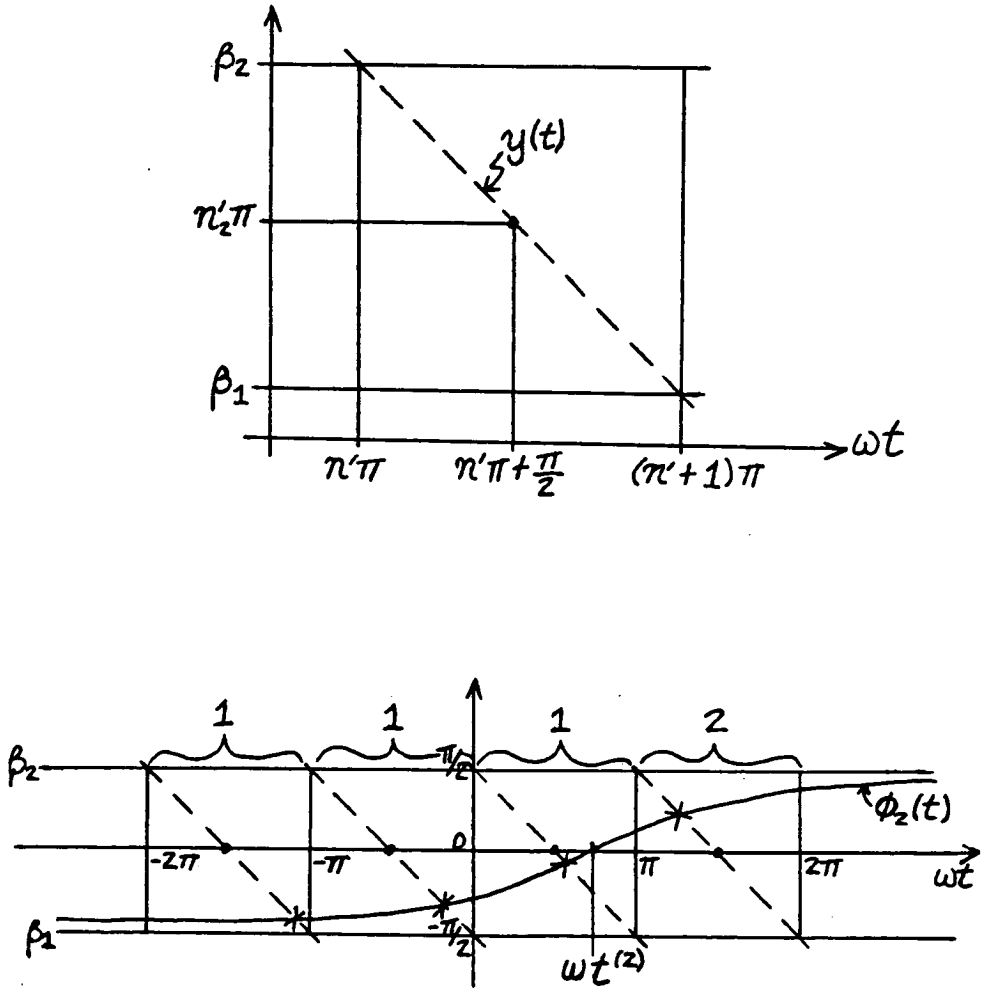


Figure A.9: The starting values for cases 5^+ and 6^+

1. If $\omega t^{(2)} > n'\pi + \pi/2$, then the intersection point of $\phi_2(t)$ and $y(t)$ must occur in the interval $\omega t \in (n'\pi, n'\pi + \pi/2)$. Between these two points, the curve of $\phi_2(t)$ and the line $y(t)$ are decreasing from β_1 to $n'_2\pi$.

2. If $\omega t^{(2)} < n'\pi + \pi/2$, then the intersection point of $\phi_2(t)$ and $y(t)$ must occur when $\omega t \in (n'\pi + \pi/2, (n'+1)\pi)$. Between these two points, the curve of $\phi_2(t)$ and the line $y(t)$ are decreasing from $n'_2\pi$ to β_2.

3. If $\omega t^{(2)} = n'\pi + \pi/2$, then the intersection point of $\phi_2(t)$ and $y(t)$ is $t = t^{(2)}$. Both $\phi_2(t^{(2)})$ and $y(t^{(2)})$ are equal to $n'_2\pi$.

An example of the behavior of $\phi_2(t)$ for these cases is shown in Figure A.10.

A.3.2.3 Cases 7^+ and 8^+

For the two cases 7^+ and 8^+ where $\phi_2(t)$ is increasing, the line segment $y(t)$ decreases from $\beta_2 = \beta_1 + \pi$ at $\omega t + \pi/2 = n'\pi$, through $k'_2\pi/2$ at $\omega t + \pi/2 = n'\pi + \pi/2$, to β_1 at $\omega t + \pi/2 = (n'+1)\pi$. The limits on the location of the zero of d^2f/dt^2 in the interval $[n'\pi, (n'+1)\pi]$ of $\omega t + \pi/2$ are:

1. If $\omega t^{(1)} + \pi/2 > n'\pi + \pi/2$, then the intersection point of $\phi_2(t)$ and $y(t)$ must occur when $\omega t + \pi/2 \in \left(n'\pi + \pi/2, \min[\omega t^{(1)} + \pi/2, (n'+1)\pi]\right)$. Between these two points, the curve of $\phi_2(t)$ is increasing from β_1 to $k'_2\pi/2$ and the line $y(t)$ is decreasing from $k'_2\pi/2$ to β_1.

2. If $\omega t^{(1)} + \pi/2 < n'\pi + \pi/2$, then the intersection point of $\phi_2(t)$ and $y(t)$ must occur when $\omega t + \pi/2 \in \left(\max[n'\pi, \omega t^{(1)} + \pi/2], n'\pi + \pi/2\right)$. Between these two points, the curve of $\phi_2(t)$ is increasing from $k'_2\pi/2$ to β_2 and the line $y(t)$ is decreasing from β_2 to $k'_2\pi/2$.

3. If $\omega t^{(1)} + \pi/2 = n'\pi + \pi/2$, then the intersection point of $\phi_2(t)$ and $y(t)$ is $t = t^{(1)}$. Both $\phi_2(t^{(1)})$ and $y(t^{(1)})$ are equal to $k'_2\pi/2$.

An example of the behavior of $\phi_2(t)$ for these cases is shown in Figure A.11.

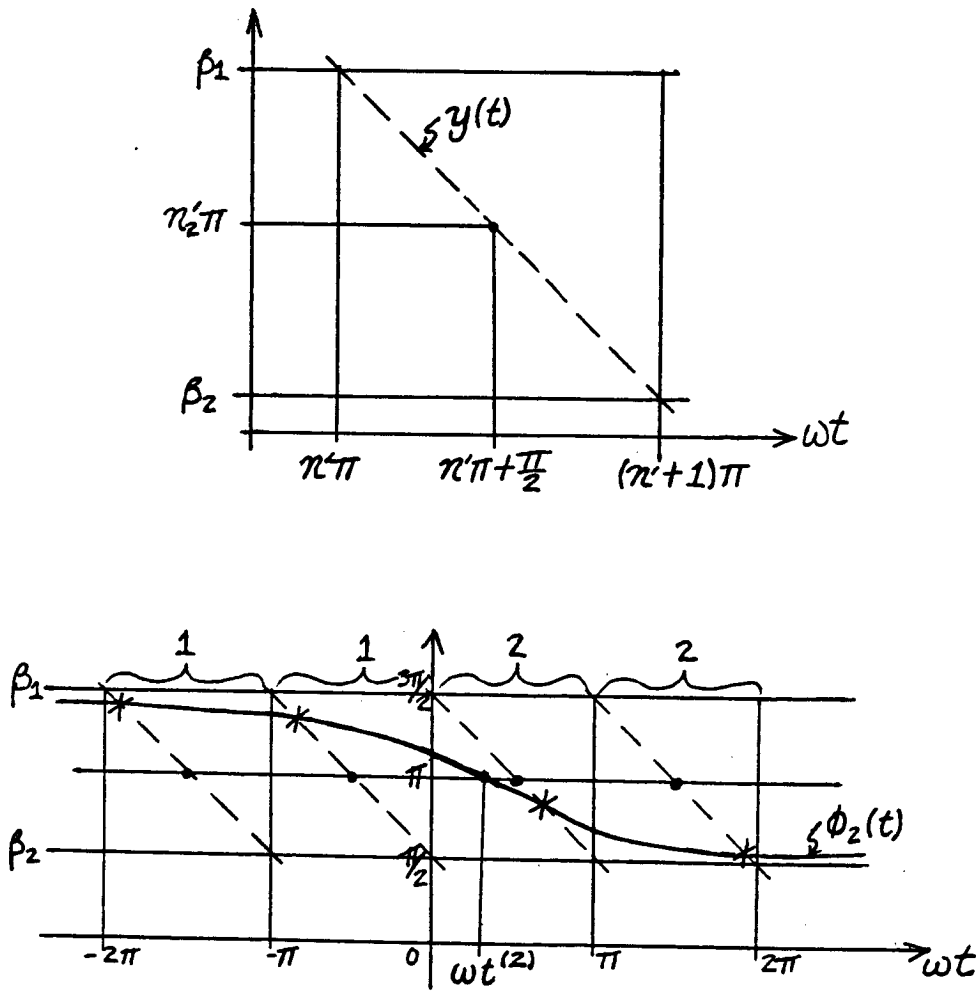


Figure A.10: The starting values for cases 5^- and 6^-

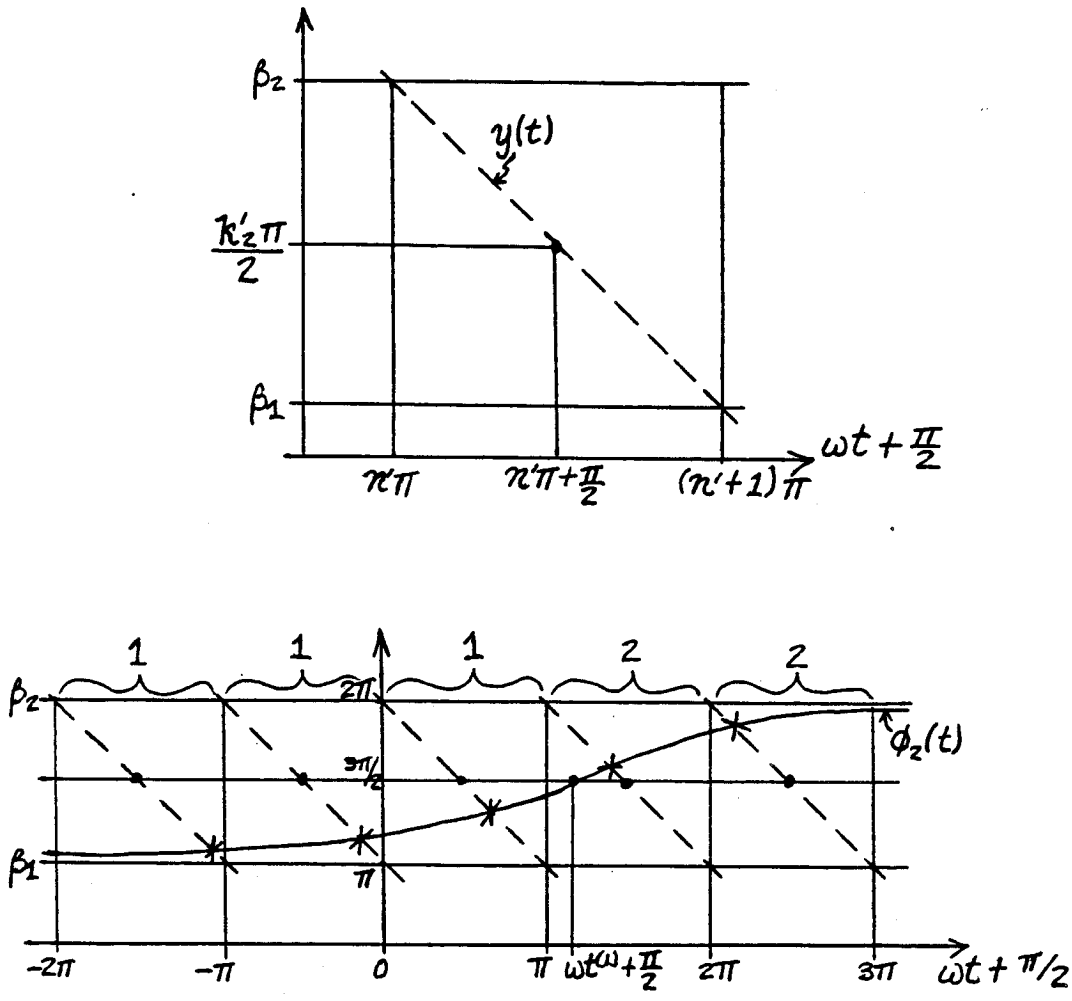


Figure A.11: The starting values for cases 7^+ and 8^+

A.3.2.4 Cases 7^- and 8^-

For the two cases 7^- and 8^- where $\phi_2(t)$ is decreasing, the line segment $y(t)$ decreases from β_1 at $\omega t + \pi/2 = n'\pi$, through $k_2'\pi/2$ at $\omega t + \pi/2 = n'\pi + \pi/2$, to $\beta_2 = \beta_1 - \pi$ at $\omega t + \pi/2 = (n'+1)\pi$. The limits on the location of the zero of d^2f/dt^2 in the interval $[n'\pi, (n'+1)\pi]$ of $\omega t + \pi/2$ are:

1. If $\omega t^{(1)} + \pi/2 > n'\pi + \pi/2$, then the intersection point of $\phi_2(t)$ and $y(t)$ must occur when $\omega t + \pi/2 \in (n'\pi, n'\pi + \pi/2)$. Between these two points, the curve of $\phi_2(t)$ and the line $y(t)$ are decreasing from β_1 to $k_2'\pi/2$.

2. If $\omega t^{(1)} + \pi/2 < n'\pi + \pi/2$, then the intersection point of $\phi_2(t)$ and $y(t)$ must occur when $\omega t + \pi/2 \in (n'\pi + \pi/2, (n'+1)\pi)$. Between these two points, the curve of $\phi_2(t)$ and the line $y(t)$ are decreasing from $k_2'\pi/2$ to β_2.

3. If $\omega t^{(1)} + \pi/2 = n'\pi + \pi/2$, then the intersection point of $\phi_2(t)$ and $y(t)$ is $t = t^{(1)}$. Both $\phi_2(t^{(1)})$ and $y(t^{(1)})$ are equal to $k_2'\pi/2$.

An example of the behavior of $\phi_2(t)$ for these cases is shown in Figure A.12.

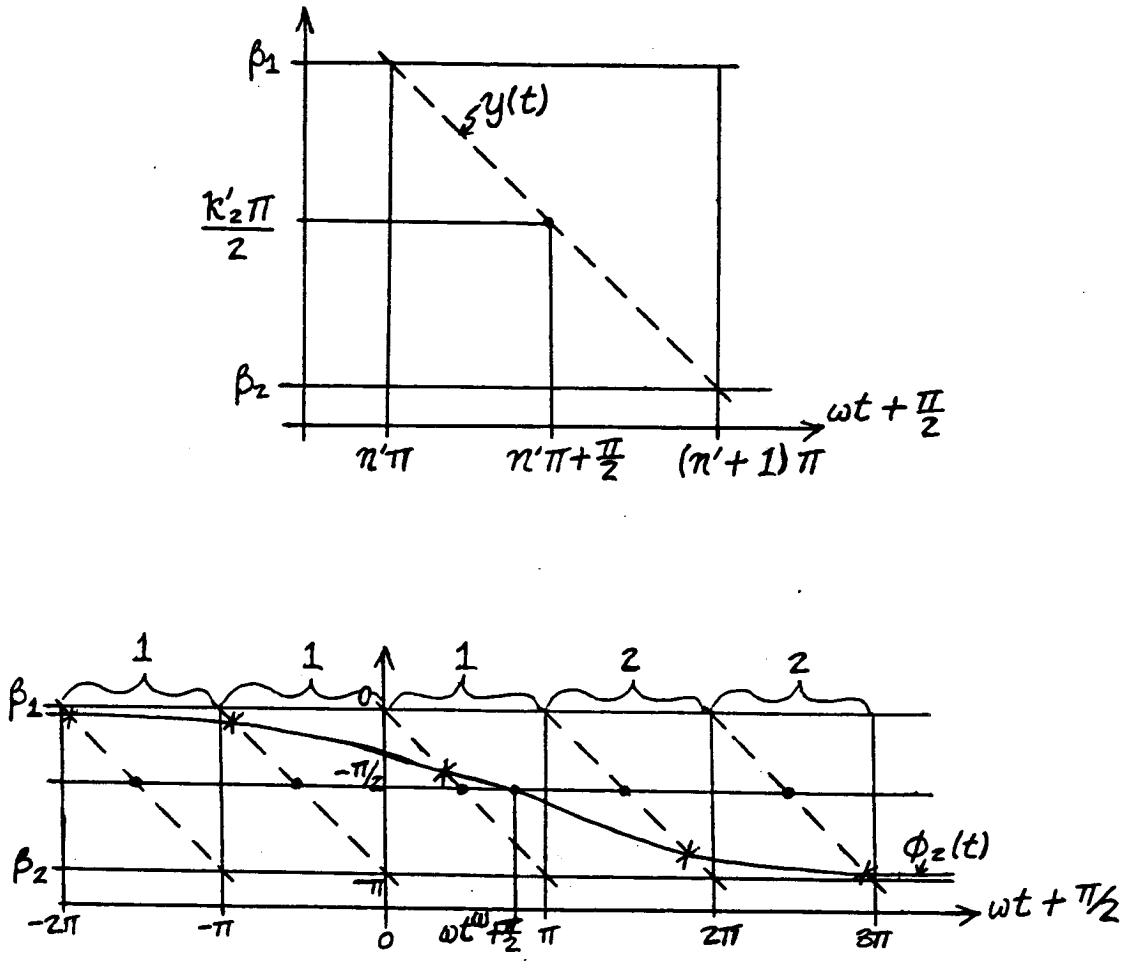


Figure A.12: The starting values for cases 7^- and 8^-

Appendix B

Two Iterative Methods for Finding the Zeros of a Function

This appendix gives a brief discussion of standard linear interpolation and its variant, the secant method. More detailed descriptions of these methods can be found in references [26,27,28].

B.1 Linear Interpolation

Linear interpolation is one of the oldest and simplest iterative methods for finding a zero of a function. It is also called the method of false position or *regula falsi*. As its name implies, this method finds a zero of a function by approximating the original function by a series of linear functions and solving for the zeros of each of these linear functions. At each step of the iteration, only one new value of the function itself is evaluated. No derivatives of the function are used for this method.

Let $f(t)$ be the function whose zero is to be found and let $t_0 < t_1$ be two distinct values of t such that the product $f(t_0)f(t_1)$ is negative. Then, there must be at least one zero of $f(t)$ in the interval $[t_0, t_1]$. Assume that there is only one zero of $f(t)$ in this interval. As explained in Chapter 5, this will always be the case in this thesis. To find the zero using linear interpolation, $f(t)$ is approximated by the line passing through $f(t_0)$ at t_0 and through $f(t_1)$ at t_1. Thus,

$$f(t) \approx \frac{(t - t_0)f(t_1) - (t - t_1)f(t_0)}{t_1 - t_0}$$

Setting the above equation equal to zero gives

$$t_2 = \frac{t_0 f(t_1) - t_1 f(t_0)}{f(t_1) - f(t_0)}$$

where t_2 is the first approximation for the zero of $f(t)$. Now, $f(t_2)$ is evaluated and compared to $f(t_0)$ and $f(t_1)$. Assuming that $f(t_2) \neq 0$, the

actual zero of $f(t)$ lies in one of the intervals $[t_0, t_2]$ or $[t_2, t_1]$ depending on whether $f(t_0)f(t_2) < 0$ or $f(t_1)f(t_2) < 0$. If $f(t_0)f(t_2) < 0$, the next approximation for the zero of $f(t)$, t_3, is found from

$$t_3 = \frac{t_0 f(t_2) - t_2 f(t_0)}{f(t_2) - f(t_0)}$$

If $f(t_1)f(t_2) < 0$, then t_0 and $f(t_0)$ are replaced by t_1 and $f(t_1)$, respectively, in the above equation for t_3. This process is repeated until the approximate value of the zero, t_n, becomes sufficiently close to the actual zero of $f(t)$. Either t_0 and $f(t_0)$ or t_1 and $f(t_1)$ is used at each step of the iteration. (As discussed in the next section, this is not the case for the secant method which uses the previous two iterates to generate the next approximation to the zero of the function.)

The equation for t_{n+1} given t_n and $f(t_n)$ is

$$t_{n+1} = \frac{t_i f(t_n) - t_n f(t_i)}{f(t_n) - f(t_i)} \text{ for } n = 1, 2, \ldots$$

where $i = 0$ if $f(t_0)f(t_n) < 0$ and $i = 1$ if $f(t_1)f(t_n) < 0$. Two alternate forms of this equation are

$$\begin{aligned} t_{n+1} &= t_i - f(t_i)\frac{t_n - t_i}{f(t_n) - f(t_i)} \\ &= t_n - f(t_n)\frac{t_n - t_i}{f(t_n) - f(t_i)} \end{aligned}$$

Figure B.1 illustrates the process of standard linear interpolation. For the graph of $f(t)$ shown in this figure, $i = 0$ and t_0 and $f(t_0)$ are used at each step in the iteration. The new approximation for the zero, t_{n+1}, is obtained from the old approximation, t_n, and t_0 using the equation

$$t_{n+1} = \frac{t_0 f(t_n) - t_n f(t_0)}{f(t_n) - f(t_0)}$$

The series $t_2, t_3, \ldots, t_n, \ldots$ generated from repeated use of the equation for t_{n+1} can be shown to converge to the value of the zero for any continuous function $f(t)$. Let the error at the $(n+1)$th step in the iteration be defined

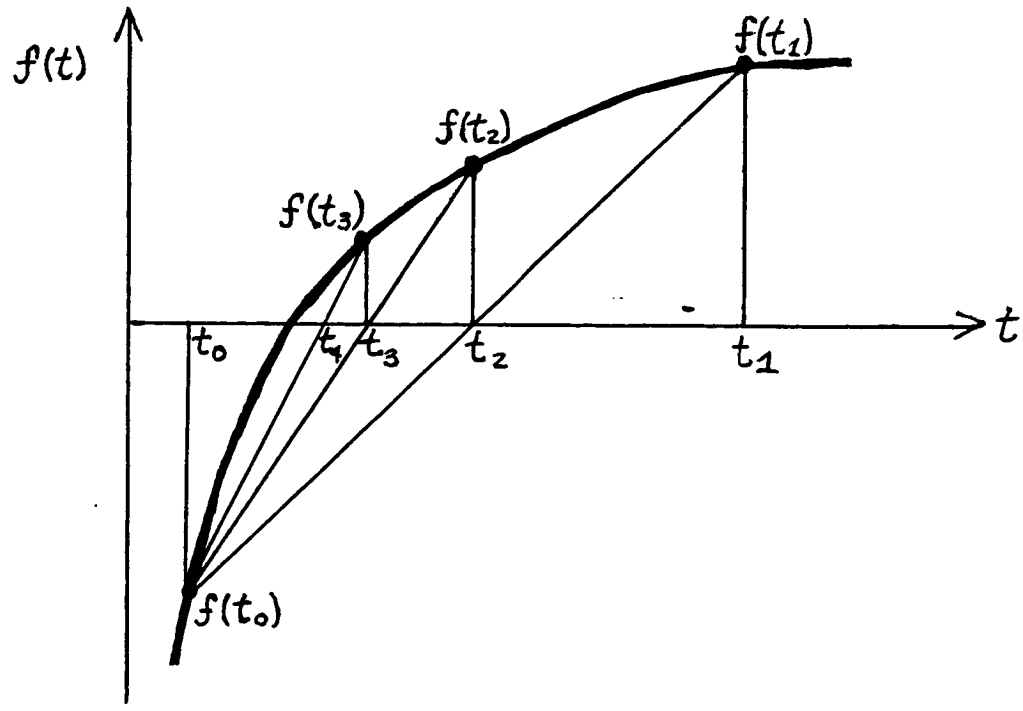


Figure B.1: The method of linear interpolation or *regula falsi*

as the absolute value of the difference between t_{n+1} and the actual value of the zero of $f(t)$. When linear interpolation is applied in a region where the function is convex, the error at each step in the iteration is proportional to the error at the previous step where the constant of proportionality is less than one. Because the error at each step is proportional to the error at the previous step raised to the *first* power, this method is said to have *linear* convergence in a region where $f(t)$ is convex.

B.2 The Secant Method

A variation of the linear interpolation method that has improved convergence is the secant method. For this method, the previous two points in the series are used to generate the next approximation to the zero of $f(t)$. This contrasts with standard linear interpolation where one of the starting points is always used at each step of the iteration. The secant method generally requires fewer steps to find the zero of $f(t)$ to some specified accuracy than the linear interpolation method. For this reason, it is used by the numerical zero search procedures implemented in this thesis.

The first two iteration steps for the secant method are the same as

for standard linear interpolation. The first step uses the starting points $t_0, f(t_0)$ and $t_1, f(t_1)$ to generate t_2. The second step uses $t_2, f(t_2)$ and either $t_0, f(t_0)$ or $t_1, f(t_1)$ to generate t_3. The equations for t_2 and t_3 are given in the preceding section. The third approximation to the zero of $f(t)$, t_4, is generated using $t_2, f(t_2)$ and $t_3, f(t_3)$. Similarly, at each subsequent step, the two previous points $t_{n-1}, f(t_{n-1})$ and $t_n, f(t_n)$ are used to generate t_{n+1}. The equation for t_{n+1} when $n = 3, 4, \ldots$ is

$$t_{n+1} = \frac{t_{n-1}f(t_n) - t_n f(t_{n-1})}{f(t_n) - f(t_{n-1})}$$

This equation can also be written in the two forms shown below:

$$\begin{aligned} t_{n+1} &= t_{n-1} - f(t_{n-1})\frac{t_n - t_{n-1}}{f(t_n) - f(t_{n-1})} \\ &= t_n - f(t_n)\frac{t_n - t_{n-1}}{f(t_n) - f(t_{n-1})} \end{aligned}$$

When t_0 and t_1 are sufficiently close to the actual zero of $f(t)$, the secant method has *superlinear* convergence. This means that the error at each step in the iteration is on the order of the error at the previous step raised to a power *greater than one*. In particular, it can be shown that the error at each step in the iteration is on the order of the error at the previous step raised to the power $(1 + \sqrt{5})/2 \approx 1.618$. This is a substantial improvement over standard linear interpolation.

The conditions determining when t_0 and t_1 are "sufficiently" close to the zero of $f(t)$ can be found in references [26,28]. Reference [28] also gives an example using both linear interpolation and the secant method that demonstrates the more rapid convergence of the secant method.

The stopping criterion for the iteration using the secant method is such that the value of the zero is computed to single precision accuracy. The relative error is defined to be $|(t_{n+1} - t_n)/t_n|$, except for the special case where $t_n = 0$ where it is defined to be $|(t_{n+1} - t_n)/t_{n+1}|$. The iteration is terminated when the relative error is less than some tolerance denoted by ϵ. For single precision computations, ϵ is set equal to 10^{-6}.

A second stopping criterion is also included that tests the total number of iteration steps. For this thesis, the iteration was stopped after twenty

steps regardless of the value of the relative error. This maximum value of twenty iteration steps was chosen arbitrarily. The test on the number of iteration steps is primarily a safety measure to insure that the iteration is stopped at some point. It was never needed for any of the examples run for this thesis. The secant method typically converged to within the desired accuracy for the value of the zero in three or four steps.