ATTITUDE DETERMINATION
OF A
HIGH ALTITUDE BALLOON SYSTEM
PART II
DEVELOPMENT OF THE PARAMETER
DETERMINATION PROCESS

by

Marquette University Systems Group

N. J. Nigro Associate Professor, Mechanical Engineering
A. F. Elkouh Associate Professor, Mechanical Engineering
P. Nimiyongskul Graduate Student
K. S. Shen Graduate Student
V. N. Jhaveri Graduate Student
A. Sethi Graduate Student

Report Prepared for:
National Aeronautics and Space Administration
Research Grant NSG 1025
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>I. INTRODUCTION ..............................................</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Motivation and Relevance of Report ....................</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Objective of Report ..................................</td>
<td>1</td>
</tr>
<tr>
<td>II. ATTITUDE DETERMINATION PROCESS .........................</td>
<td>5</td>
</tr>
<tr>
<td>2.1 Introduction ........................................</td>
<td>5</td>
</tr>
<tr>
<td>2.2 System Simulation Process ............................</td>
<td>5</td>
</tr>
<tr>
<td>- System Math Model ..................................</td>
<td>8</td>
</tr>
<tr>
<td>- Solution of System Model ............................</td>
<td>12</td>
</tr>
<tr>
<td>- Computation of Balloon Angular Velocities ...........</td>
<td>13</td>
</tr>
<tr>
<td>2.3 Parameter Estimation Method ..........................</td>
<td>15</td>
</tr>
<tr>
<td>2.4 Hooke and Jeeves - Direct Search Method ............</td>
<td>16</td>
</tr>
<tr>
<td>2.5 Test Problem ........................................</td>
<td>23</td>
</tr>
<tr>
<td>III. RESULTS AND CONCLUSION ................................</td>
<td>26</td>
</tr>
<tr>
<td>3.1 Data for LACATE Mission ..............................</td>
<td>26</td>
</tr>
<tr>
<td>- Balloon and System Data .............................</td>
<td>26</td>
</tr>
<tr>
<td>- Standard Atmosphere Data ............................</td>
<td>26</td>
</tr>
<tr>
<td>- Math Model Data ....................................</td>
<td>26</td>
</tr>
<tr>
<td>- Gyro Data ..........................................</td>
<td>26</td>
</tr>
<tr>
<td>- Eigenvalue Problem ..................................</td>
<td>27</td>
</tr>
<tr>
<td>3.2 Evaluation of Parameter Determination Process ......</td>
<td>43</td>
</tr>
</tbody>
</table>

Appendix

A. Fortran Coding for Computation of Eigenvalues and
Eigenvectors(Eq. 2.2-8).................................... 45

B. Fortran Coding for Hooke and Jeeves - Direct Search
Method.................................................. 59

BIBLIOGRAPHY.............................................. 62
CHAPTER I
INTRODUCTION

1.1 Motivation and Relevance of Report In April of 1974 the National Aeronautics and Space Administration conducted a high altitude balloon experiment called LACATE (Lower Atmosphere Composition and Temperature Experiment) which employed an infrared radiometer to sense remotely vertical profiles of the concentrations of selected atmospheric trace constituents and temperature. The constituents were measured by inverting infrared radiance profile of the earth's horizon. The radiometer line of sight was scanned vertically across the horizon at approximately 0.25° per second, requiring 30 seconds to acquire a complete radiance profile. The specifications required that the relative vertical position of the data points making up a profile be known to approximately 30 arc seconds. The general description of the balloon system for accomplishing the mission is given in reference (1), refer figures 1.1-1 and 1.1-2.

In order to fix the orientation of the line of sight of the radiometer, it is necessary to be able to determine the configuration of the platform in space, i.e. the attitude of the system. This can be accomplished by simulating the balloon system and using the gyro output in conjunction with a parameter estimation process. Simulation of the balloon system requires a mathematical model plus analysis of the model. The required mathematical model has already been developed for use with the system simulation process and the details are described in reference (1) pages 9-38. The attitude of the balloon system can be determined once the initial conditions; i.e. the initial state, is known.

1.2 Objective of the Report The main objective of this report will be to develop a process to determine the unknown initial state parameters by employing the output of the system mathematical model in conjunction with
Figure 1.1-1 LOCATU Balloon System

- Balloon
- Radar Reflector
- Connector
- APOL System Parachute
- Flight Control Electronics
- Connector
- Radiomotor
- Instrumentation
- Crush Pad
- Ballast
Figure 1.1-2  Idealized Balloon System
the output obtained from the instrumentation system. This system consisted of three orthogonally oriented rate gyros and magnetometer which were fixed to the balloon platform. The development of this process involves 2 major steps. First, a parameter determination process must be developed which solves for the unknown initial state parameters of the system which, in turn, give the best fit to the data obtained from the instrumentation system. Second, a simulation process must be developed for computing the state of the balloon system which can then be compared with the actual state of the system.

The body of this report consists of 2 major parts. The first part discusses the **attitude determination process** in general. The details of this part include

(i) **System Simulation** Discussion of the method of solution and computation of the output of interest (i.e. angular velocity components along the platform axis).

(ii) **Computation of System Natural Frequencies.** Discussion of a method for solving system eigenvalues.

(iii) **Optimization Technique.** Discussion of the Hooke and Jeeves direct search method.

(iv) **Verification Process.** Discussion of a method for verification of the attitude determination process.

In the second part of the report all the numerical data and results are presented and discussed.
2.1 Introduction

The attitude (state) of the balloon system can be determined as a function of time if (a) a method for simulating the motion of the system is available and (b) the initial state is known. The system motion can be simulated once the system model is determined. The initial state can then be obtained by fitting the system motion (as measured by sensors) to the corresponding output predicted by the mathematical model. In the case of the LACATE experiment the sensors consisted of three orthogonally oriented rate gyros and a magnetometer all mounted on the research platform. The initial state was obtained by fitting the angular velocity components measured with the gyros to the corresponding values obtained from the solution of the math model.

A block diagram illustrating the attitude determination process employed for the LACATE experiment is shown in figure 2.1-1. The process consists of three essential parts; i.e., a process for simulating the balloon system (block 1), an instrumentation system (block 2) for measuring the output, and a parameter estimation process (block 3) for systematically and efficiently solving the initial state. A more detailed discussion of each of these parts is presented below.

2.2 System Simulation Process

The main steps in the system simulation process are shown in the block diagram of figure 2.2-1. They consist of (a) development of a system model to predict state (block 1), (b) solution of the model (block 2) and (c) developing a math model for computing the angular velocity components of the research platform. Each of these steps will be discussed in greater detail below.
Figure 2.1-1  Attitude Determination Process
System Initial State

Determination of System Mathematical Model

Solution of System Mathematical Model

Determination of Platform Angular Velocity Components

Platform Angular Velocity

Figure 2.2-1 System Simulation Process
System math model  A system math model must be obtained which enables one to predict the motion of the system at float altitude since all forms of output (e.g. angular velocity, tension in the cables, etc.) can be determined once this is known. This motion is very complex and involves various types of oscillation including bounce, pendulation and spin. Moreover, the complexity of the motion is increased with increasing number of subsystems.

The math model of the balloon system is complicated primarily by the two important factors. They are as follows:

(a) the balloon itself is a distributed parameter system which has motion in an infinite fluid media. Hence, it is necessary to first idealize it as an equivalent rigid body in order to develop a lumped parameter model for the entire system.

(b) The balloon system is subjected to nondeterministic wind gusts which result in forces acting externally on the system. At the present time very little is known about the nature of these gusts.

The exact dynamic model for the balloon itself consists of the equations of motion for the solid (i.e. balloon fabric) and the fluid dynamic equations. These equations are coupled through the boundary conditions which must be satisfied at the interface of the solid and fluid media. The resulting model is extremely complex and consists of a system of coupled partial differential equations. The system model is simplified by treating the balloon as a lumped parameter (rigid body) element. This is accomplished by developing approximate expressions for the aerodynamic forces and torques which result due to the interaction between the balloon and fluid media. These forces and torques are then treated as external reactions on the solid system.

A linear systems model which includes the effect of the aerodynamic reactions has been developed (3) by neglecting the effect of second order
terms. An equivalent form of this model which involves only the pendulation angles in two orthogonally oriented planes is given as follows

\[ \ddot{\bar{\theta}} + \lambda \dot{\bar{\theta}} = \ddot{\bar{\theta}}, \]  
\[ \ddot{\bar{\psi}} + \lambda \dot{\bar{\psi}} = \ddot{\bar{\psi}}, \]  
\[ \dot{\phi}_3 = 0, \]  
where

\[ \bar{\theta} = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix}, \]  
\[ \bar{\psi} = \begin{bmatrix} \psi_1 \\ \psi_2 \\ \psi_3 \end{bmatrix}. \]

\[ \bar{\theta}, \bar{\psi} = \text{pendulation angles in two orthogonally oriented planes}, \]

and

\[ \Lambda = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}. \]  

The elements \( a_{ij} \) are defined in terms of the system parameters in reference (3). The pendulation angles \( \theta_i \) and \( \psi_i \) and the spin angle \( \phi \) are illustrated in figure 2.2-2 and figure 2.2-3.
Figure 2.2-2 Eulerian Angles
Figure 2.2-3 System Pendulation Angles in $X_2X_3$ Plane
Solution of System Model. The solution to equation 2.2-1 can be obtained by assuming the following form for $\bar{\theta}$; i.e.

$$\bar{\theta} = \tilde{X} \sin \omega t.$$  \hspace{1cm} 2.2-7

Substitution in to equation 2.2-1 yields the following eigenvalue problem, i.e.

$$A\tilde{X} = \omega^2 \tilde{X}, \text{ where}$$  \hspace{1cm} 2.2-8

$$\tilde{X}_i(i = 1, 2, 3) = X_{1i}, X_{2i}, X_{3i}$$

are the eigenvectors and

$$\Omega_i(i = 1, 2, 3)$$

are the eigenvalues.

Equation 2.2-8 can be solved numerically on the computer. A listing of the fortran program employed to solve this equation for the particular problem under study is given in appendix A.

The solution to equation 2.2-2 is obtained in the same way and yields the identical eigenvalue problem. The solution for equation 2.2-3 is easily obtained by integrating the equation twice with respect to time ($t$). The final closed form solution for $\bar{\theta}$, $\bar{\psi}$ and $\phi_3$ is given as

$$\bar{\theta}(t) = \tilde{X}_1(c_1 \sin \Omega_1 t + c_2 \cos \Omega_1 t)$$

$$+ \tilde{X}_2(c_3 \sin \Omega_2 t + c_4 \cos \Omega_2 t)$$

$$+ \tilde{X}_3(c_5 \sin \Omega_3 t + c_6 \cos \Omega_3 t),$$  \hspace{1cm} 2.2-9
\[
\dot{\psi}(t) = \bar{X}_1 (t_1 \sin \Omega_1 t + c_8 \cos \Omega_1 t) \\
+ \bar{X}_2 (c_9 \sin \Omega_2 t + c_{10} \cos \Omega_2 t) \\
+ \bar{X}_3 (c_{11} \sin \Omega_3 t + c_{12} \cos \Omega_3 t), \quad \text{and} \\
\phi_3(t) = at + \phi_3((t_0), \text{ where}
\]
2.2-10

\[c_j (j = 1, 2, \ldots, 12) = \text{unknown constants which are determined from the initial states; i.e. } \bar{\theta}(t_0), \bar{\psi}(t_0), \dot{\bar{\theta}}(t_0) \text{ and } \dot{\bar{\psi}}(t_0),
\]

\[a = \text{constant rate of spin, and}
\]

\[\phi_3((t_0) = \text{initial spin displacement.}
\]

**Computation of Balloon Angular Velocities.** The relationship between the platform motion (\(\theta_3', \dot{\theta}_3', \psi_3', \dot{\psi}_3', \phi_3' \text{ and } \dot{\phi}_3\)) and the platform angular velocity components is obtained through the application of the Euler angle transformations to the system platform shown in figure 2.2-4. The transformation equations are given as follows:

\[
\omega_1 = \dot{\theta}_3 \cos \psi_3 \cos \phi_3 + \psi_3 \sin \phi_3,
\]

\[
\omega_2 = -\dot{\theta}_3 \cos \psi_3 \sin \phi_3 + \dot{\psi}_3 \cos \phi_3,
\]

\[
\omega_3 = \dot{\theta}_3 \sin \psi_3 + \dot{\phi}_3, \quad \text{where}
\]

2.2-12

\[\omega_i (i = 1, 2, 3) = \text{angular velocity components of the platform along } \bar{e}_i \text{ direction,}
\]

\[\theta_3 = \text{pendulation angle in the } \bar{e}_2 \bar{e}_3 \text{ plane,}
\]

\[\psi_3 = \text{pendulation angle in the } \bar{e}_1 \bar{e}_3 \text{ plane, and}
\]

\[\phi_3 = \text{spin angle about the } \bar{e}_3 \text{ axis.}
Figure 2.2-4  Platform Angular Velocities
2.3 **Parameter Estimation Method**

The main object of the parameter estimation method is to determine the initial system state \((\vec{s}_0, \vec{\psi}_0, \phi_3^0)\) such that the rates \((\vec{\omega})\) obtained from the rate gyros fit (over some time interval \(0 \leq t \leq T\)), in an optimal sense, those rates \((\vec{\omega})\) predicted from the system model. With this initial state determined, the instantaneous system state is obtained simply as the output \((\vec{s}(t), \vec{\psi}(t), \phi_3(t))\) of the system model. Hence, the problem is basically one of the parameter determination in which the initial state parameters \((\vec{s}_0, \vec{\psi}_0, \phi_3^0)\) play the role of the unknown parameters. For the purpose of this work, the platform rates will be fit in a least square sense; i.e., a performance function \((\phi)\) will be formed and the initial state determined such that this function is minimized.

The process will be repeated (i.e., \(\vec{s}_0, \vec{\psi}_0\) and \(\phi_3^0\) will be updated) every \(T\) seconds. Initially \(T\) will be set equal to 30 seconds (time required to acquire a complete radiance profile) although a study (to be conducted in the future) will be made to determine the minimum \(T\) required such that the necessary precision and accuracy are satisfied.

In this research, the function \(\phi\) is formed as follows:

\[
\phi = \sum_{j=1}^{N} \sum_{i=1}^{3} (\omega_i - \vec{\omega}_i)^2, 
\]

where

- \(N\) = number of data points taken in \(0 \leq t \leq T\),
- \(\omega_i\) = angular velocity computed from system model, and
- \(\vec{\omega}_i\) = angular velocity given by the rate gyros and to be compared with \(\omega_i\).

The function \(\phi\) is clearly dependent on the initial state. This initial state is obtained from the condition that \(\phi\) take on a minimum; i.e., by solving the following optimization problem

\[
\min \phi = \phi(\vec{\omega}).
\]

The angular velocity components \((\omega_i)\) are obtained from the transformation
equations 2.2-12 and since \( \varrho, \psi, \phi_3 \) are functions of the initial state \((\varrho_o, \psi_o, \phi_{3o})\), equation 2.3-2 can be written as

\[
\min \psi = \psi(\varrho_o, \psi_o, \phi_{3o}) \tag{2.3-3}
\]

Since the values of \( \psi \) are obtained numerically (\( \tilde{\omega} \) is given as a discrete data point) it will be necessary to employ some direct search technique to solve the above optimization problem. In general, the algorithm for any direct search techniques is given as follows

\[
\tilde{x}_{o}^{k+1} = \tilde{x}_{o}^{k} + \delta \tilde{x}_{o}^{k}, \quad (k = 1, 2, 3, \ldots), \tag{2.3-4}
\]

\( \tilde{x}_{o}^{k} \) is the vector of old values,
\( \delta \tilde{x}_{o}^{k} \) is a vector of increments, and
\( \tilde{x}_{o}^{k+1} \) is the vector of improved values.

The vector \( \delta \tilde{x}_{o}^{k} \) is found such that \( \psi(\tilde{x}_{o}^{k+1}) < \psi(\tilde{x}_{o}^{k}) \). The value of \( k \) is incremented until \( \tilde{x}_{o} \) converges; i.e., until the norm of \( \delta \tilde{x}_{o} \) satisfies some error criteria. Figure 2.3-1 illustrates the application of equation 2.3-4 to minimize \( \psi \).

There are many direct search techniques for systematically determining \( \delta \tilde{x}_{o} \). After comparing several of these, the direct search method of Hooke and Jeeves was chosen for the following reasons, i.e., (a) the special feature in accelerating of distance, so called pattern search, and (b) this method is already available in subroutine form (5) and was employed with the process outlined in figure 2.3-1. The discussion of Hooke and Jeeves direct search algorithm is presented in section 2.4.

2.4 Hooke and Jeeves - Direct Search Method.

The Hooke and Jeeves [2] method is a simple and powerful univariant method for finding the minimum of a function. A modification of the basic univariant numerical search method, it involves trial explorations and then...
Input: Initial guess \((\mathbf{x}_0), \mathbf{f}, \) and required accuracy (ERR).

Solve System Math Model

Evaluate \(\phi(\mathbf{x}_0)\)

\[
\mathbf{x}_0 = \mathbf{x}_0 + \delta\mathbf{x}_0
\]

Obtain \(\delta\mathbf{x}\) from direct search technique

\(|\delta\mathbf{x}_0| : ERR\)

\(|\delta\mathbf{x}_0| \leq ERR\)

Output \(\mathbf{x}(t)\)

Figure 2.3-1 Block Diagram Illustrating Process for Attitude Determination of LACATE Mission
ever-expanding steps, called pattern moves, in the direction indicated by
the explorations. The method is designed to follow a ridge, so that it does
not suffer from the main disadvantages of the basic univariant approach
which can not effectively cope with ridges or sharp valleys.

The Hooke and Jeeves algorithm consists of two major phases, an
"exploratory search" around the current base point and a "pattern search"
in the direction selected for minimization. Figure 2.4-1 is a simplified
informational flow diagram for the algorithm as implemented by Wook [4].
The steps (blocks) in this figure are as follow:

Block 1 The initial estimates for all decision variables (X_i) as well
as initial incremental changes or step sizes (ΔX_i) in the decision variables
are provided.

Block 2 The objective function,  \( \phi(X) \) is evaluated at the base point
\( (X) \) which is the vector of initial guesses (X_i) of the decision variables.

Block 3 An exploratory search (type 1) is performed next, i.e.
each decision variable (X_i) is changed in rotation, one at a time, by the
incremental amount (ΔX_i) and the objective function is evaluated at the
new point. If this incremental fails to improve the objective function
then X_i is changed by (-ΔX_i) and the value of \( \phi(X) \) again evaluated as
before. If the objective function is still not improved then X_i is left
unchanged, and the same procedure is employed again with X_i+1. This
process is repeated until all the decision variables (X_1, X_2,...,X_n) have
been so changed. Figure 2.4-2 illustrates the steps in an exploratory search
for a two dimensional problem. For each change in the decision variable,
the value of the objective function \( \phi(X) \) is compared with its value at the
previous point. If, upon completion of the exploratory search, none of the
changes yields an improved \( \phi(X) \) (i.e. X_1,X_2,...,X_n) remain unchanged , then
the stages in block 7 are performed next; otherwise block 4 is implemented.

Block 4 After completing the type I exploratory search, the new base
1. **Input:** Initial base point \((X)\), and required accuracy \((ERR)\)

2. Evaluate \(\phi(\bar{X})\) at the base point

3. Carry out type I exploratory search from base point. Compare \(\phi_{i+1}(X)\) and \(\phi_i(X)\)

4. If \(\phi_{i+1} < \phi_i\), Set new base point \(\phi_i(\bar{X}) = \phi_{i+1}(\bar{X})\)

5. Carry out pattern search

6. Carry out type II exploratory search. Compare \(\phi_{i+1}(\bar{X})\) with \(\phi_i(\bar{X})\)

7. Compare increment size \((\Delta X)\) with required accuracy \((ERR)\)

8. Reduce increment size.

\(\Delta X > ERR\) \(\Rightarrow\) STOP

\(\Delta X < ERR\)

Figure 2.4-1 Block Diagram for Pattern Search Method.
Final base point after one exploratory search

Base point

Failure (retain initial base point)

Figure 2.4-2: Exploratory Search Strategy
point is set equal to the final base point obtained from block 3.

**Block 5** After completing the type I exploratory search and obtaining a new base point, a "pattern search" is made. The new value of the decision variables define a vector, $S$ (see figure 2.4-2), that represent a successful direction for minimization. A series of pattern searches is now made along this vector, usually in increments of $2|S|$ until $\phi(X)$ no longer decreases. The magnitude of the step sizes for the pattern search (i.e., $S'$ in figure 2.4-3) is roughly proportional to the number of successes previously encountered in each coordinate direction during the exploratory searches for the previous cycle. The success or failure of a pattern move is not established until after a type II exploratory search (block 6) has been completed. If $\phi(X)$ does not decrease after the type II exploratory search, then the pattern search has failed and a new type I exploratory search (block 3) is made in order to define a new successful direction.

The base point for a pattern search in the case of a two dimensional problem is illustrated in figure 2.4-3.

**Block 6** The type II exploratory search (figure 2.4-3) is made after a temporary exploration point is obtained from a pattern search. The chief difference between the type I and type II exploratory searches is the magnitude of the step sizes ($\Delta X_i$). In the case of the type II exploratory search, $\Delta X_i$ is taken as some multiple of the $\Delta X_i$ (i.e. $C_i \Delta X_i$), see figure 2.4-3) used in the type I exploratory search. This is done in order to accelerate the search.

**Block 7** If the type I exploratory search fails to give a new successful direction, then the current $|\Delta X_i|$ is compared to some preset allowable tolerance (error input in block 1). If $|\Delta X_i|$ is larger than the allowable error, then block 8 is implemented. Failure to improve $\phi(X)$ for $|\Delta X_i|$ smaller than the allowable error indicates that a local optimum has been
3, 5: Temporary base points
2, 4: Current base points
1 + 3, 2 + 4: Pattern search

Figure 2.4-3 General Search Strategy
reached and the search is terminated.

**Block 8** If $|\Delta x_1|$ is still larger than the prespecified error, then $|\Delta x_1|$ is reduced gradually until the type I exploratory search can be employed to define a new successful direction.

In order to terminate the search, two additional basic tests must be satisfied. These are described as follows:

(i) After each exploratory and pattern search, the increment in the objective function $|\Delta f|$ is compared with a prescribed test value. If this increment is less than the test value, then the exploratory or pattern search is said to have failed. In this case block 3 or 7 is implemented.

(ii) If $|\Delta f|$ is greater than the prescribed test value, then a test is made to determine if the objective has increased (a failure) or decreased (a successful search). This second test ensures that the values of the objective function is always being improved.

The fortran coding for the Hooke and Jeeves method has been provided (with some minor revision) by M.I.T. Joint Computer Facility. This is given in appendix B. The program is available in subroutine form with various parameters in the calling sequence. These parameters include

(i) the number of decision variables (n),
(ii) the initial guesses of the variables ($x_1, x_2, \ldots, x_n$),
(iii) the required accuracy (ERR), and
(iv) the allowable number of allowable iterations.

2.5 Test Problem

In order to verify the accuracy and precision of proposed parameter estimation process (refer section 2.3) it is necessary to employ the process to a test problem in which the initial state is already known. Preferably, the test problem model should be identical to the one employed in this study; i.e. equations 2.2-1 to 2.2-3. With the initial state known, the
resulting values of \( \theta_3, \psi_3, \phi_3, \dot{\theta}_3, \dot{\psi}_3 \) and \( \dot{\phi}_3 \) which determines the state of the platform (as a function of time) can be obtained from equations 2.2-9 to 2.2-11. With these, the values of \( \omega_1, \omega_2 \) and \( \omega_3 \) can be computed from equation 2.2-12.

The testing process will consist of the following steps.

(a) Assign fixed values to the initial state parameters \( \bar{\theta}(0), \bar{\psi}(0), \bar{\phi}(0) \) and \( \bar{\psi}(0) \), or equivalently to the unknown constants \( c_1, c_2, \ldots, c_{12} \) in equations 2.2-9 and 2.2-10.

(b) Compute the orientation (state) of the platform \( \theta_3, \psi_3, \phi_3, \dot{\theta}_3, \dot{\psi}_3, \dot{\phi}_3 \) as a function of time by employing equations 2.2-9 to 2.2-11.

(c) Determine the values of \( \ddot{\theta}_1, \ddot{\psi}_2 \) and \( \ddot{\phi}_3 \) as a function of time by substituting the results from part (b) into equation 2.2-12.

(d) Employ the results (sampled at various times) from part (c) as input to the parameter estimation process and utilize this process to recover the initial state values (or unknown constants \( c_1, c_2, \ldots, c_{12} \)).

The accuracy and precision of the process can be determined by comparing the results of part (d) to the corresponding assumed values of part (a). The testing procedure can be repeated for various sets of input values. A flow chart illustrating the test process is given in figure 2.5-1.
Input: Initial guess \( (c) \) and required accuracy

Solve math model, (i.e. employ equations 2.2-9 to 2.2-11)

Compute platform angular velocity (i.e. employ transformation equation 2.2-12)

Compute objective function
\[
\phi = \sum_{i=1}^{n} \sum_{j=1}^{m} (\omega_i - \bar{\omega}_i)^2
\]

Employ Hooke and Jeeves direct search technique and check for required accuracy

Assumed \( \bar{\omega} \) computed from equation 2.2-12 with known \( C \)

> Required accuracy

\( \leq \) Required accuracy

Output \( \bar{C} \)

Compare output with the known \( \bar{C} \)

Figure 2.5-1 Flow Chart for Test Problem
CHAPTER III
RESULTS AND CONCLUSIONS

3.1 Data for LACATE Mission

Balloon and System Data. Figure 1.1-1 illustrates the actual LACATE balloon system and figure 1.1-2 illustrates the corresponding idealized system used in this study. The values for the various lengths and masses of the idealized system are given in table 3.1-1.

The actual design profile of the balloon at float altitude is illustrated in figure 3.1-1 and the corresponding dimensions are given in table 3.1-2. Values for (a) mass center, (b) center of volume, and (c) moment of inertia (at the mass center) of the balloon were computed and these values are presented in table 3.1-3 along with other balloon properties.

Standard Atmosphere Data. Figure 3.1-2 illustrates the U.S. Standard Atmosphere and the corresponding properties are presented in table 3.1-4. Table 3.1-5 presents the actual properties of the atmosphere at the float altitude (47 kilometers).

Math Model Data. The expressions for the element of A matrix of equations 2.2-1 and 2.2-2 for the system model are given in reference (3). The numerical values for these elements were computed based on the data given above and these values are presented in table 3.1-6.

Gyro Data. The platform coordinate axis which were employed for referencing the angular velocity components (as measured by the instrumentation package) do not coincide with the coordinate axis used in the Euler angle transformation equation (2.2-12). The relationship between these two coordinate systems is shown in figure 3.1-3. The resulting transformation equations which relate the angular velocity components are given as
\( \omega_1 = -\tilde{\omega}_1 \), \quad 3.1-1

\( \omega_2 = -\tilde{\omega}_2 \), and \quad 3.1-2

\( \omega_3 = -\tilde{\omega}_3 \), where \quad 3.1-3

\( \omega_i \) is the angular velocity component along the \( X_i \) axis, and

\( \tilde{\omega}_i \) is the corresponding angular velocity component obtained from the gyro.

The numerical values of the angular velocity components obtained from the gyro system are presented in appendix C. Typical plots of the angular velocity components obtained from the gyros are illustrated in figures 3.1-4 to 3.1-6.

The azimuth angle \( \phi_m \) obtained from the magnetometer was measured clockwise from magnetic north to the negative \( X_1 \) platform axis. However, the angle \( \phi \) in equation 2.2-11 is measured counterclockwise from north to the same negative \( X_1 \) platform axis. This is illustrated in figure 3.1-7. The transformation equation relating these two angles is given as

\[ \phi = 360^\circ - \phi_m, \quad \text{where} \]

\[ \phi \] is the spin displacement in equation 2.2-11, and

\[ \phi_m \] is the spin displacement measured by the magnetometer.

A typical plot of the azimuth angle \( \phi_m \) is shown in figure 3.1-8.

**Eigenvalue Problem.** The solution to the eigenvalue problem (equation 2.2-8) was obtained by employing the computer program given in appendix A in conjunction with the coefficients presented in table 3.1-6. The solution for the eigenvalues \( \Omega_i \) and corresponding eigenvectors is presented in table 3.1-7. The magnitude of \( \Omega_i \) is equal to the natural frequency of the system.
### TABLE 3.1-1

Idealized LACAT System Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r_0$ (maximum radius of the balloon)</td>
<td>249.989 ft.</td>
</tr>
<tr>
<td>$r_1$ (distance from center of mass of balloon shell to mass $m_1$)</td>
<td>222.091 ft.</td>
</tr>
<tr>
<td>$r_2$ (distance from mass $m_1$ to $m_2$)</td>
<td>75 ft.</td>
</tr>
<tr>
<td>$r_3$ (distance from mass $m_2$ to $m_3$)</td>
<td>15 ft.</td>
</tr>
<tr>
<td>$d$ (distance from center of mass (C.G.) to center of volume (C.V.))</td>
<td>23.712 ft.</td>
</tr>
<tr>
<td>$m_0$ (mass of balloon shell)</td>
<td>2850 lb$_m$</td>
</tr>
<tr>
<td>$m_1$ (lumped mass)</td>
<td>135 lb$_m$</td>
</tr>
<tr>
<td>$m_2$ (lumped mass)</td>
<td>135 lb$_m$</td>
</tr>
<tr>
<td>$m_3$ (lumped mass)</td>
<td>375 lb$_m$</td>
</tr>
</tbody>
</table>
Figure 3.1-1 Actual Balloon Profile
Table 3.1-2

Balloon Profile Data

<table>
<thead>
<tr>
<th>HEIGHT-Z (ft)</th>
<th>RADIUS-r (ft)</th>
<th>FABRIC WT.-WZ (lbs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>.0</td>
<td>.2</td>
<td>.0</td>
</tr>
<tr>
<td>10.6</td>
<td>32.2</td>
<td>42.2</td>
</tr>
<tr>
<td>21.9</td>
<td>63.8</td>
<td>102.8</td>
</tr>
<tr>
<td>34.3</td>
<td>95.1</td>
<td>181.8</td>
</tr>
<tr>
<td>46.8</td>
<td>122.2</td>
<td>267.2</td>
</tr>
<tr>
<td>62.9</td>
<td>151.7</td>
<td>380.7</td>
</tr>
<tr>
<td>81.9</td>
<td>179.5</td>
<td>511.6</td>
</tr>
<tr>
<td>104.4</td>
<td>204.4</td>
<td>658.6</td>
</tr>
<tr>
<td>130.7</td>
<td>225.4</td>
<td>3'3.7</td>
</tr>
<tr>
<td>160.5</td>
<td>240.7</td>
<td>991.9</td>
</tr>
<tr>
<td>193.1</td>
<td>248.9</td>
<td>1171.3</td>
</tr>
<tr>
<td>226.6</td>
<td>248.8</td>
<td>1353.3</td>
</tr>
<tr>
<td>258.9</td>
<td>240.1</td>
<td>1532.5</td>
</tr>
<tr>
<td>287.9</td>
<td>223.1</td>
<td>1703.9</td>
</tr>
<tr>
<td>311.7</td>
<td>199.4</td>
<td>1862.8</td>
</tr>
<tr>
<td>329.4</td>
<td>170.9</td>
<td>2005.7</td>
</tr>
<tr>
<td>340.9</td>
<td>139.3</td>
<td>2130.2</td>
</tr>
<tr>
<td>347.4</td>
<td>106.4</td>
<td>2315.5</td>
</tr>
<tr>
<td>350.4</td>
<td>72.9</td>
<td>2491.0</td>
</tr>
<tr>
<td>351.0</td>
<td>.0</td>
<td>2850.0</td>
</tr>
</tbody>
</table>
TABLE 3.1-3

BALLOON PROPERTIES

\( r_o \) (maximum radius) = 249.989 ft.
\( Z_o \) (height, corresponded to \( r_o \)) = 209.526 ft.
\( V_H \) (inflated volume) = 45,378,282 cu. ft.
\( H \) (inflated height) = 350.1 ft.
\( m_o \) (total weight of balloon shell including top cap weight) = 2880 lbs.
Gore length = 674.83 ft.
Surface area = 635,711 sq. ft.
\( Z_1 \) (distance from bottom apex to the mass center of balloon shell) = 222.091 ft.
\( Z_2 \) (distance from nadir to center of volume) = 198.379 ft.
\( t \) (thickness of balloon shell (strato film\(^\text{\textregistered}\)) = 0.0006 inch.
\( I_{ol} \) (moment of inertia at the mass center (C.G.) = 2.622374 \( \times \) 10\(^6\) slug-ft\(^2\)
Figure 3.1-2 U.S. Standard Atmosphere (6)
Table 3.1-4
GeneraJ Properties of the U.S. Standard Atmosphere (6)

| ALTI- | TEMPER- | LAPSE | PRESSURE | DENSITY |
| TUDÉ, | ATURE | RATE | (N/m²) | (kg/m³) |
| (m) | (C) | (C/km) | (m/sec²) | | |
| 0 | 15.0 | Polytropic | -6.5 | 9.790 | 1.225 | 1.013 x 10⁴ |
| 11,000 | -56.5 | Polytropic | +1.0 | 9.685 | 0.924 | 8.680 x 10² |
| 20,000 | -56.5 | Isothermal | 0.0 | 9.654 | 1.019 x 10² | 1.427 x 10⁻³ |
| 32,000 | -44.5 | Polytropic | +2.8 | 9.633 | 1.031 | 8.804 x 10⁻² |
| 47,000 | -2.5 | Polytropic | -2.0 | 9.633 | 1.031 | 8.804 x 10⁻² |
| 52,000 | -2.5 | Polytropic | -3.0 | 9.592 | 1.030 | 8.680 x 10⁻² |
| 61,000 | -20.5 | Polytropic | -4.0 | 9.592 | 1.030 | 8.680 x 10⁻² |
| 79,000 | -91.5 | Polytropic | -5.5 | 9.549 | 1.030 | 8.680 x 10⁻² |
| 83,743 | -92.5 | Polytropic | -6.5 | 9.549 | 1.030 | 8.680 x 10⁻² |

C = temperature in degrees Centigrade

km = Kilometers

n = polytropic exponent

\( g \) = local acceleration of gravity
TABLE 3.1-5

Properties of Atmosphere at Float Altitude (47 km)

\[ \rho_{\text{ca}} \text{ (density of air)} = 2.78 \times 10^{-6} \text{ slug/ft}^3 \]

\[ \rho_{\text{OH}} \text{ (density of helium)} = 3.834 \times 10^{-7} \text{ slug/ft}^3 \]

\[ P_o \text{ (pressure)} = 2.254 \text{ lb/ft}^2 \]

\[ C_D \text{ (viscous drag coefficient)} = .5 \]

\[ g \text{ (gravity)} = 31.7 \text{ ft/sec}^2 \]

\[ \mu_a \text{ (viscosity of air)} = 3.57 \times 10^{-7} \text{ lbm/ft sec} \]

\[ \mu_H \text{ (viscosity of helium)} = 3.74 \times 10^{-7} \text{ lbm/ft sec} \]
TABLE 3.1-6

Coefficients of A Matrix [Equation 2.2-1 & 2.2-2]

<table>
<thead>
<tr>
<th>i</th>
<th>j</th>
<th>( a_{ij} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>( 2.10463 \times 10^{-2} )</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>( -6.08078 \times 10^{-2} )</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>( 0.0 )</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>( -3.8521 \times 10^{-2} )</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>( 1.74685 )</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>( -7.98370 )</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>( -1.17407 )</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>( 7.98370 )</td>
</tr>
</tbody>
</table>
$X_i$ = coordinate system employed for purposes of math model.

$X_{iG}$ = gyro coordinate system.

Figure 3.1-3  Gyro vs Math Model Coordinate System
direction along magnetometer boom.

Figure 3.1-7 Measurement of Spin Displacement
TABLE 3.1-7
Solution for the Eigenvalues $\sigma_i$ and Corresponding Eigenvectors.

<table>
<thead>
<tr>
<th>$i$</th>
<th>$\sigma_i$</th>
<th>$\mathbf{x}_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$1.296692 \times 10^{-1}$</td>
<td>9.08376</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.997919</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.000</td>
</tr>
<tr>
<td>2</td>
<td>$7.066434 \times 10^{-1}$</td>
<td>$-7.550639 \times 10^{-2}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.937457</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.000</td>
</tr>
<tr>
<td>3</td>
<td>3.038977</td>
<td>$6.555123 \times 10^{-4}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$-0.156788$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.000</td>
</tr>
</tbody>
</table>
3.2 Evaluation of Parameter Determination Process

The process for evaluating the attitude determination process by employing a test problem was discussed in section 2.5. The form of the math model for the test problem was identical to the system model given in equation 2.2-1 through 2.2-3. Several different sets of unknown constants \( C_i \) were assumed and the resulting angular velocity components \( \mathbf{\Omega}_i \) were computed by employing the solution equations 2.2-9 to 2.2-11 and the transformation equation 2.2-12. These values of \( \mathbf{\Omega}_i \) were input to parameter determination process which was then employed to recover the unknown constants.

Table 3.2-1 gives the results for one test case. These results indicate that the process is capable of recovering the unknown constants \( c_i \) with good precision. The results presented in table 3.2-1 are typical of those obtained for other test cases. Hence one can conclude that the process is suitable for evaluating the attitude of the LACATE system.

During the next stage of the research, the process will be employed (in conjunction with the math model given in eqs. 2.2-1 – 2.2-3) to determine the angular velocity components of the platform of the LACATE balloon system. These values will then be compared to the actual data in order to evaluate the system model. Results will be presented in the next report.
APPENDIX A

Fortran Coding for Computation of Eigenvalues and Eigenvectors (Eq. 2.2-8).
1.000 DIMENSION BMAT(7,7),AC(7,7),C(50,10),X(50,1)
2.000 DIMENSION A(50),RR(50),RI(50),IANA(50),W(50),P(50),F(50)
3.000 COMMON /BLK1/ A101,D,R(3)
4.000 OUTPUT 'ENTER HERE ---> A101, D, R(1)
5.000 INPUT A101, D, R(1)
6.000 DO 1000 I=1,7
7.000 DO 1000 J=1,7
8.000 BMAT(I,J)=0.0
9.000 1000 AC(I,J)=0.0
10.000 CALL COEFF(BMAT, AC)
11.000 DO 4000 J=1,3
12.000 A(J)=BMAT(J,1)
13.000 A(J+3)=BMAT(J,2)
14.000 A(J+6)=BMAT(J,3)
15.000 4000 WRITE(108,5000) BMAT(J, I=1,3)
16.000 5000 FORMAT('A4(1PE15.5)')
17.000 OUTPUT 'ENTER M ORDER OF THE MATRIX'
18.000 INPUT M
19.000 CALL HSAG(M,A,M)
20.000 WRITE(108,70)
21.000 70 FORMAT('/', 'X' TRANGULAR (ALMOST!) MATRIX A FROM HSAG /
22.000 DO 40 L=1,M
23.000 40 WRITE(108,10) A(I), I=L, M*(M-1)+L, M
24.000 10 FORMAT('A4(1PE15.5)')
25.000 CALL ATEIG(M,A,RR,RI, IANA, M)
26.000 WRITE(108,80)
27.000 80 FORMAT('/', 'EIGENVALUE /
28.000 OUTPUT IANA(I), I=1, M
29.000 OUTPUT RR(I), I=1, M
30.000 OUTPUT RI(I), I=1, M
31.000 3000 FORMAT('/', 'RESULTS ON RR(I), OMEGA(I), REDDII(I), FREQUENCY(I) /
32.000 DO 2000 I=1, M
33.000 W(I)=ABS(5GR), RR(I))
34.000 P(I)=2.3.1415926/W(I)
35.000 F(I)=W(I)/(2.3.1415926)
36.000 2000 WRITE(108,6000) I, RR(I), W(I), P(I), F(I)
37.000 6000 FORMAT('5X', 'I2', '4(1PE16.6)')
38.000 C
39.000 C
40.000 90  OUTPUT 'ENTER EIGENVALUE'
41.000  INPUT E
42.000  X<3,1>=1.00
43.000  X<1,1>=(-B<1,3>**B<2,2>-E+B<1,1>*B<2,3>)/(B<1,1>)
44.000  1-E*(B<2,2>-E-B<1,2>*B<2,1>)
45.000  X<2,1>=(-B<2,3>**B<1,1>-E+B<1,3>*B<2,1>)/(B<1,1>)
46.000  1-E*(B<2,2>-E-B<1,2>*B<2,1>)
47.000  WRITE(108,7000) E
48.000 7000  FORMAT('HERE IS THE EIGENVALUE --> ',F10.5,')'
49.000  1 'AND THE CORRESPONDING EIGENVECTOR : ' ')
50.000  OUTPUT X<1,1>,X<2,1>,X<3,1>
51.000  OUTPUT 'ENTER 0 IF YOU HAVE MORE EIGENVECTOR TO COMPUTE'
52.000  INPUT IKJ
53.000  IF(IKJ.EQ.0)  GO TO 90
54.000  STOP 'EIGVEC'
55.000  END
SUBROUTINE HSIG

PURPOSE
TO REDUCE A REAL MATRIX INTO UPPER ALMOST TRIANGULAR FORM

USAGE
CALL HSIG(N,A,IA)

DESCRIPTION OF THE PARAMETERS
N    ORDER OF THE MATRIX
A    THE INPUT MATRIX, N BY N
IA   SIZE OF THE FIRST DIMENSION ASSIGNED TO THE ARRAY
A IN THE CALLING PROGRAM WHEN THE MATRIX IS IN
DOUBLE SUBSCRIPTED DATA STORAGE MODE. IA=N WHEN
THE MATRIX IS IN SSP VECTOR STORAGE MODE.

REMARKS
THE HESSIAN FORM REPLACES THE ORIGINAL MATRIX IN THE
ARRAY A.

SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
NONE

METHOD
SIMILARITY TRANSFORMATIONS USING ELEMENTARY ELIMINATION
MATRICES, WITH PARTIAL PIVOTING.

REFERENCES
J.H. WILKINSON - THE ALGEBRAIC EIGENVALUE PROBLEM -
CLARENDON PRESS, OXFORD, 1965.

SUBROUTINE HSIG(N,A,IA)
DIMENSION A(I)
DOUBLE PRECISION S
L=N
NIA=L+IA
LIA=NIA-IA

L IS THE ROW INDEX OF THE ELIMINATION

20 IF(L-3) 360,40,40
40 LIA=LIA-IA
L1=L-1
L2=L1-1

SEARCH FOR THE PIVOTAL ELEMENT IN THE LTH ROW

ISUB=LIA+L
IPIV=ISUB-IA
PIV=ABS(A(IPIV))
IF(L-3) 30,90,50
50 M=IPIV-IA
DO 80 I=L,M,IA
T=ABS(A(I))
IF(T-PIV) 80,80,60
60 IPIV=I
PIV=T
80 CONTINUE
90 IF(IPIV) 100,320,100
100 IF(IPIV-ABS(A(ISUB))) 120,130,120

INTERCHANGE THE COLUMNS

120 M=IPIV-L
DO 140 I=1,L
J=M+I
T=A(J)
K=LIA+I
A(J)=A(K)
140 A(K)=T

INTERCHANGE THE ROWS
C
M=L2-M/IA
DO 160 I=L1,N,IA
T=A(I)
J=I-M
A(I)=A(J)
160 A(J)=T
C
TERMS OF THE ELEMENTARY TRANSFORMATION
C
120 DO 200 I=L,IA,IA
200 A(I)=A(I)/A(ISUB)
C
RIGHT TRANSFORMATION
C
J=-IA
DO 240 I=1,L2
J=J+IA
LJ=J+J
DO 220 K=1,L1
KJ=K+J
KL=K+LIA
220 A(KJ)=A(KJ)-A(LJ)*A(KL)
240 CONTINUE
C
LEFT TRANSFORMATION
C
K=-IA
DO 300 I=1,N
K=K+IA
LK=K+L1
S=A(LK)
LJ=L-IA
DO 280 J=1,L2
JK=K+J
LJ=LJ+1A
280 S=S+A(LJ)*A(JK)*1.0DO
300 A(LK)=S
SET THE LOWER PART OF THE MATRIX TO ZERO

DO 310 I=L,LIA,IA
310 A(I)=0.0
320 L=L1
50 TO 20
360 RETURN
END

SUBROUTINE ATEIG

PURPOSE
COMPUTE THE EIGENVALUES OF A REAL ALMOST TRIANGULAR MATRIX

USAGE
CALL ATEIG(M,A,RR,RI,IANA,IA)

DESCRIPTION OF THE PARAMETERS
M   ORDER OF THE MATRIX
A   THE INPUT MATRIX, M BY M
RR  VECTOR CONTAINING THE REAL PARTS OF THE EIGENVALUES ON RETURN
RI  VECTOR CONTAINING THE IMAGINARY PARTS OF THE EIGENVALUES ON RETURN
IANA VECTOR WHOSE DIMENSION MUST BE GREATER THAN OR EQUAL TO M, CONTAINING ON RETURN INDICATIONS ABOUT THE WAY THE EIGENVALUES APPEARED (SEE MATH. DESCRIPTION)
IA  SIZE OF THE FIRST DIMENSION ASSIGNED TO THE ARRAY A IN THE CALLING PROGRAM WHEN THE MATRIX IS IN DOUBLE SUBSCRIPTED DATA STORAGE MODE. IA=M WHEN THE MATRIX IS IN SSP VECTOR STORAGE MODE.

REMARKS
THE ORIGINAL MATRIX IS DESTROYED
THE DIMENSION OF RR AND RI MUST BE GREATER OR EQUAL TO M

SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
NONE

METHOD
OR DOUBLE ITERATION

REFERENCES

SUBROUTINE ATEIG(M,A,RR,R1,IANA,IA)
DIMENSION A(1),RR(1),R1(1),PRR(2),PRI(2),IANA(1)
INTEGER P,P1,Q

E7=1.0E-8.
E6=1.0E-6
E10=1.0E-10
DELTA=0.5
MAXIT=30

INITIALIZATION

N=M
20 M1=N-1
IN=M1+IA
NN=IN+N
IF(M1) 30,1300,30
30 NP=N+1

ITERATION COUNTER

IT=0

ROOTS OF THE 2ND TRIER MAIN SUBMATRIX AT THE PREVIOUS ITERATION
DO 40 I=1,2
PRT(I)=0.0
40 PRT(I)=0.0

C LAST TWO SUBDIAGONAL ELEMENTS AT THE PREVIOUS ITERATION
C
PAN=0.0
PAN1=0.0

C ORIGIN SHIFT
C
R=0.0
S=0.0

C ROOTS OF THE LOWER MAIN 2 BY 2 SUBMATRIX
C
N2=N1-1
IN1=IN-IA
MN1=IN1+N
NIN=IN1+H1
NIN1=IN1+1
60 T=A(N11)-A(NN)
U=T*T
V=4.0*A(N11)*A(NN1)
IF ABS(V)<U*E7) 100,100,65
65 T=U+V
IF ABS(T)-AMAX1(U,ABS(V))*E6) 67,67,68
67 T=0.0
68 U=(A(N11)+A(NNN))/2.0
V=SQRT(ABS(T))/2.0
IF (T)>140,70,70
70 IF (U) 80,75,75
75 RR(N1)=U+V
RR(N1)=U-V
GO TO 130
80 RR(N1)=U-V
RR(N1)=U+V
GO TO 130
100 IF (T)>120,110,110
110 RR(N1)=A(N11)
   RR(N)=A(NN)
   GO TO 130
120 RR(N1)=A(NN)
   RR(N)=A(N11)
130 RI(N)=0.0
   RI(N1)=0.0
   GO TO 160
140 RR(N1)=U
   RR(N)=U
   RI(N1)=V
   RI(N)=-V
160 IF(N2)$1280,1280,180
C
C TESTS OF CONVERGENCE
C
180 N1N2=N1N1-1A
   RMOD=RR(N1)*RR(N1)+RI(N1)*RI(N1)
   EPS=E10*SORT(RMOD)
   IF(Abs(A(N1N2))=EPS)$1280,1280,240
240 IF(Abs(A(N1N1))-EPS)$1300,1300,250
250 IF(Abs(A(N1N1)+Abs(A(N1N1)))*E6)$1240,1240,260
260 IF(Abs(A(N1N1))=Abs(A(N1N1))*E6)$1240,1240,300
300 IF(IT-MAXIT)320,1240,1240
C
C COMPUTE THE SHIFT
C
320 J=1
   DO 360 I=1,2
      K=NP-I
      IF(Abs(RR(K))=PRR(I))+Abs(RI(K))=PRI(I)-DELTAS(A(RR(K))
1 +Abs(RI(K)))340,360,360
340 J=J+1
360 CONTINUE
   GO TO (440,460,460,480)*J
440 R=0.0
   S=0.0
   GO TO 500
460 J=N+2-J
R=RR(J)*RR(J)
S=RR(J)+RR(J)
GO TO 500
480 R=RR(N)*RR(N1)-RI(N)*RI(N1)
S=RR(N)+RR(N1)
C
SAVE THE LAST TWO SUBDIAGONAL TERMS AND THE ROOTS OF THE
SUBMATRIX BEFORE ITERATION
C
500 P=0=A(NN1)
N=N1=0(NIN2)
DO 520 I=1,N2
K=NP-1
PIR(I)=RI(K)
520 P=0=I=1,N2
C
SEARCH FOR A PARTITION OF THE MATRIX, DEFINED BY P AND Q
C
P=N2
IF (N-3)*600,600,525
525 IP1=NIN2
DO 580 J=2,N2
IP1=IP1+1A
IF (ABS(A(IP1)) ) EPS 600,600,530
530 IP1F=IP1+1A
IP1F=IP1+1A
D=A(IP1)*A(IP1)-S)+A(IP1F)*A(IP1F)+1)+R
540 IF (D) 540,560,540
540 IF (ABS(A(IP1)*A(IP1F)+1)*ABS(A(IP1F)+A(IP1F)+1)-S)+ABS(A(IP1F+2)
1)) EPS 620,620,560
560 P=N1-J
580 CONTINUE
600 Q=P
GO TO 680
620 P1=P-1
Q=P1
630 CONTINUE
650 DO 660 I=2,P1
IP1=IP1+1A-1
660 CONTINUE
IF (ABS(A(IPI)) - EPS) .GT. 680, 680, 660
660 Q = Q - 1

C OR DOUBLE ITERATION
C
680 II = (P - 1) * IA + P
DO 1220 I = P, N1
   II1 = II - IA
   IIP = II + IA
   IF (I - P) .GT. 720, 700, 720
700 IFI = II + 1
   IFIP = IIP + 1
C
C INITIALIZATION OF THE TRANSFORMATION
C
61 = A(II) * (A(II) - S) + A(IIP) * A(IPI) + R
62 = A(IPI) * (A(IPIP) + A(II) - S)
63 = A(IPI) * A(IPIP + 1)
R(IPI + 1) = 0.0
60 TO 780
720 61 = A(II1)
62 = A(II1 + 1)
   IF (I - N2) .GT. 740, 740, 760
740 63 = A(II1 + 2)
60 TO 780
760 63 = 0.0
780 CAP = SORT (61 + 62 + 62 + 63 + 63)
   IF (CAP) .GT. 800, 860, 800
800 IF (61) .GT. 820, 840, 840
820 CAP = - CAP
840 T = 61 + CAP
   PSI1 = 62 / T
   PSI2 = 63 / T
   ALPHA = 2.0 / (1.0 + PSI1 + PSI1 + PSI2 + PSI2)
60 TO 880
860 ALPHA = 2.0
   PSI1 = 0.0
   PSI2 = 0.0
880 IF (1 - 0) .GT. 900, 960, 900
900 IF(I-P)920,940,920
920 A(I)=-CAP
GO TO 960
940 A(I)=-A(I)
C
      ROW OPERATION
C
960    IJ=II
      DO 1040 J=I,N
      T=PSI1*A(IJ+1)
      IF(I-N)980,1000,1000
980    IP2J=IJ+2
      T=T+PSI2*A(IP2J)
1000   ETA=ALPHA*(T+A(IJ))
      A(I,J)=A(I,J)-ETA
      A(IJ+1)=A(IJ+1)-PSI1*ETA
      IF(I-N)1020,1040,1040
1020   A(IP2J)=A(IP2J)-PSI2*ETA
1040   IJ=IJ+1
C
      COLUMN OPERATION
C
1060   IF(I-N)1080,1060,1060
1080   K=N
      GO TO 1100
1060   K=I+2
1080   IP=IIP-I
      DO 1180 J=Q,K
      JIP=IP+J
      JI=JIP-1A
      T=PSI1*A(JIP)
      IF(I-N)1120,1140,1140
1120   JIP2=JIP+1A
      T=T+PSI2*A(JIP2)
1140   ETA=ALPHA*(T+A(JI))
      A(JI)=A(JI)-ETA
      A(JIP)=A(JIP)-ETA*PSI1
      IF(I-N)1160,1180,1180
1160   A(JIP2)=A(JIP2)-ETA*PSI2
1180 CONTINUE
  IF(I-N2)1200,1220,1220
1200  JI=JI+1
  JIP=JI+1A
  JIP2=JIP+1A
  ETA=ALPHA*PSI2*A(JIP2)
  A(JI)=-ETA
  A(JIP)=-ETA*PSI1
  A(JIP2)=A(JIP2)-ETA*PSI2
1220  II=IIP+1
  IT=IT+1
  GO TO 60
C END OF ITERATION
C
1240  IF(ABS(A(NN1))>ABS(A(N))>ABS(A(N1))) 1300,1280,1280
C TWO EIGENVALUES HAVE BEEN FOUND
C
1280  IANA(N)=0
  IANA(N1)=2
  N=N2
  IF(N2>1400,1400,20
C ONE EIGENVALUE HAS BEEN FOUND
C
1300  RF(N)=A(NN)
  R1(N)=0.0
  IANA(N)=1
  IF(N1>1400,1400,1320
1320  N=N1
  GO TO 20
1400  RETURN
END
APPENDIX B

Fortran Coding for Hooke and Jeeves Direct Search Method.
SUBROUTINE OPTMIN(PSI,SSI,N,DEL,DLMIN,LIMIT,FUNCTION)
DIMENSION PSI(1),PHI(25),THT(25),EPS(25)
ITNUM=0
ALFA=1.02
C EVALUATE AT INITIAL BASEPOINT.
CALL FUNCTION(PSI,SSI)
C START AT BASEPOINT.
100 $=SSI
   DO 101 I=1,N
101 PHI(I)=PSI(I)
   ICALL=1
   IF(ITNUM.GT.LIMIT) WRITE(108,2000); RETURN
C MAKE EXPLORATORY MOVES.
   GO TO 150
C IS PRESENT VALUE LESS THAN BASEPOINT VALUE?
160 IF(S.LT..9999*SSI) GO TO 200
   GO TO 300
C SET NEW BASEPOINT.
200 SSSI=$
   DO 201 I=1,N
201 THT(I)=PSI(I)
   PSI(I)=PHI(I)
C MAKE PATTERN MOVES.
   PHI(I)=PHI(I)+ALFA*(PHI(I)-THT(I))
   CALL FUNCTION(PHI,PSI)
   S=SPI
   ICALL=2
C MAKE EXPLORATORY MOVES.
   IF(ITNUM.GT.LIMIT) WRITE(108,2000) LIMIT; RETURN
   GO TO 150
C IS PRESENT VALUE LESS THAN BASEPOINT VALUE?
250 IF(S.LT..9999*SSI) GO TO 200
   GO TO 100
300 IF(DEL.LT.DLMN) RETURN
DEL=DEL/2.
GO TO 100
C MAKE EXPLORATORY MOVES.
150 DO 100 K=1,N
EPS(K)=.05+PHI(K)
IF(EPS(K).EQ.0.) EPS(K)=.05
PHI(K)=PHI(K)+EPS(K)*DEL
CALL FUNCT(PHI,SP1)
IF(SP1.LT.S) GO TO 179
PHI(K)=PHI(K)-2.*EPS(K)*DEL
CALL FUNCT(PHI,SP1)
IF(SP1.LT.S) GO TO 179
PHI(K)=PHI(K)+EPS(K)*DEL
GO TO 180
179 S=S1
180 CONTINUE
ITNUM=ITNUM+1
GO TO (160,260), ICALL
2000 FORMAT(/// '****VALUES OF X(I) DO NOT CONVERGE IN ' ,16,' ITERATIO
INS****///)
END


