

AD-A107 070

NAVAL UNDERWATER SYSTEMS CENTER NEWPORT RI
UNDERWATER TARGET TRACKING USING MEASUREMENTS FROM A LINEAR ARR--ETC(U)
OCT 81
NUSC-TR-6010

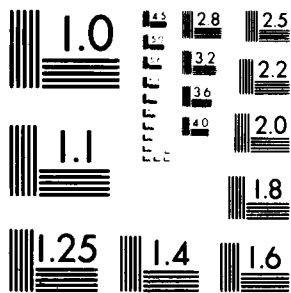
F/0 17/1

UNCLASSIFIED

NL

1 1
A-1
1

END
DATE
INDEXED
12-81
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

NUSC Technical Report 6010
22 October 1981

LEVEL

12

Underwater Target Tracking Using Measurements from a Linear Array of Omnidirectional Acoustic Elements

Combat Control Systems Department

AD A107070

DTIC
EXECTE
NOV 9 1981
H



Naval Underwater Systems Center
Newport, Rhode Island / New London, Connecticut

Approved for public release; distribution unlimited.

81 11 06 028

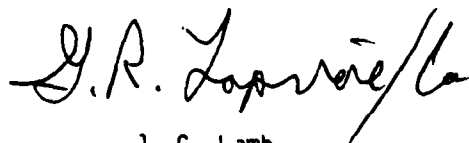
DTIC FILE COPY

PREFACE

This investigation was conducted under NUSC Project No. B45127, "WAA and Multisensor TMA Improvements for Advanced Combat Control Systems," principal investigator, B. W. Guimond (Code 3521), and under Navy Subproject and Task No. S0222-AS. The sponsoring activity is the Naval Sea Systems Command, program manager, R. Cockerill (SEA-63D6).

The major contributor to this report was D. J. Murphy, Professor of Electrical Engineering, Southeastern Massachusetts University, who is a consultant to the Systems Analysis and Synthesis Branch (Code 3521) of the NUSC Combat Control Systems Department. Technical reviewer for this report was B. W. Guimond (Code 3521).

REVIEWED AND APPROVED: 22 October 1981



J. C. Lamb
Head, Combat Control Systems Department

14

1776 to 1 repl.

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER TR-6010	2. GOVT ACCESSION NO. AD A107070	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) UNDERWATER TARGET TRACKING USING MEASUREMENTS FROM A LINEAR ARRAY OF OMNIDIRECTIONAL ACOUSTIC ELEMENTS.		5. TYPE OF REPORT & PERIOD COVERED
7. AUTHOR(s) Combat Control Systems Department		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Underwater Systems Center Newport, Rhode Island 02840		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Subproject/Task No. 16 S0222-AS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Sea Systems Command (SEA-63D6) Washington, DC 20632		12. REPORT DATE 22 Oct 1981
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) 12/63		13. NUMBER OF PAGES 66
15. SECURITY CLASS. (of this report) UNCLASSIFIED		
15a. DECLASSIFICATION / DOWNGRADING SCHEDULE		
16. DISTRIBUTION STATEMENT (of this Report) Approved for Public Release; Distribution Unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Underwater Target Tracking Arrayed Sonar Systems Acoustic Signal Processing		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) Tracking of underwater targets using measurements from a geometrically linear array of omnidirectional acoustic elements is a nonlinear estimation problem. The technique constructed and used in this report is an iterative, least-squares algorithm which uses the modified Gauss-Newton method. In this algorithm the search direction is found by applying a Householder transformation to the Gauss-Newton equations; the size of the step in this direction is found by minimizing a polynomial approximation to the		

406068

20. Abstract (Cont'd)

performance index. The results show: (1) the sign on the estimate of the target's depth cannot be uniquely determined, (2) the Gauss-Newton method works as well as its modified version for these experiments, (3) the problem is singular on the first leg, and (4) the initialization of the algorithm influences convergence.

TABLE OF CONTENTS

	Page
LIST OF ILLUSTRATIONS	ii
LIST OF TABLES	iii
LIST OF SYMBOLS	v
1. INTRODUCTION	1
2. SIGNAL MODEL	5
3. ITERATIVE LEAST-SQUARES TECHNIQUE	13
4. EXPERIMENTAL RESULTS	21
5. CONCLUSIONS AND FURTHER WORK	33
6. REFERENCES	35
APPENDIX A. THE JACOBIAN OF $f(x)$	A-1
APPENDIX B. STEP SIZE FOR THE MODIFIED GAUSS-NEWTON METHOD	B-1
APPENDIX C. NUMBER OF OPERATIONS FOR THE MODIFIED GAUSS-NEWTON METHOD	C-1
APPENDIX D. COMPUTER PROGRAM LISTING	D-1

Accession For	
NTIS GRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Avail and/or	
Special	
A	

LIST OF ILLUSTRATIONS

Figure		Page
1	A Plane Wave Arriving at a Two-Element Omni-directional Linear Array	7
2	The Surfaces of Four Cones, Each for a Different Time Delay	7
3	The (R_{11} , R_T , R_z) Coordinates System Fixed at the Array	10
4	Two Coordinate Systems Fixed at the Array	10
5	A Single Bottom-Bounce Propagation Path	23
6	Target and Observer Tracks in the (R_y , R_x) Plane	23
B-1	A Diagram for Selecting a_i , the Step Size for the i th Iteration of the Modified Gauss-Newton Method	B-3

LIST OF TABLES

Table		Page
1	True Target Parameters with Speed in Meters/ Second and Range in Meters	21
2a	Results from the Modified Gauss-Newton Method with $\sigma_t = 0.0$ Seconds (First Member of Set No. 1)	26
2b	Results from the Modified Gauss-Newton Method with $\sigma_t = 0.05 \times 10^{-5}$ Seconds (Second Member of Set No. 1)	26
2c	Results from the Modified Gauss-Newton Method with $\sigma_t = 0.25 \times 10^{-5}$ Seconds (Third Member of Set No. 1)	27
3a	Results from the Modified Gauss-Newton Method with $\sigma_t = 0.0$ Seconds (First Member of Set No. 2)	27
3b	Results from the Modified Gauss-Newton Method with $\sigma_t = 0.05 \times 10^{-5}$ Seconds (Second Member of Set No. 2)	28
3c	Results from the Modified Gauss-Newton Method with $\sigma_t = 0.25 \times 10^{-5}$ Seconds (Third Member of Set No. 2)	28
4a	Results from the Modified Gauss-Newton Method with $\sigma_t = 0.0$ Seconds (First Member of Set No. 3)	29
4b	Results from the Modified Gauss-Newton Method with $\sigma_t = 0.05 \times 10^{-5}$ Seconds (Second Member of Set No. 3)	29
4c	Results from the Modified Gauss-Newton Method with $\sigma_t = 0.25 \times 10^{-5}$ Seconds (Third Member of Set No. 3)	30
5	Estimates from the First Two Legs of Experiment Set No. 3.	31

LIST OF TABLES (Cont'd)

Table		Page
C-1	Number of Operations Required for the $(J f)$ Array and P11	C-2
C-2	Number of Operations Required to Transform the First Column of the A Matrix	C-3
C-3	Number of Operations Required for the Householder Transformation of the A Matrix	C-4
C-4	Final Count of Operations for One Iteration of the Modified Gauss-Newton Method.	C-5

LIST OF SYMBOLS

b = distance between adjacent omnidirectional elements in the linear array

d = perpendicular distance traveled by the plane wave as it moves from one element to the next

α = a conical angle

t_1 = time delay

t_m = measured time delay

c_{11} = angle of rotation

v = noise term

σ_t = standard deviation of the time delay noise

σ_α = standard deviation of the noise on α

k = time count

h = sampling period in seconds

ψ = observer's heading angle

∇ = gradient operator

$\|x\|$ = the norm of x

x^T = the transpose of x

PI1 = performance index for the nonlinear problem

PI2 = performance index for the linearized problem

g_i = "direction" of i th step (not unit magnitude)

a_i = length of the i th step

$x = (V_{xt}, V_{yt}, R_{xt}(0), R_{yt}(0), R_{zt}(0))^T$

$f_k(x)$ = the k th measurement residual

LIST OF SYMBOLS (Cont'd)

R_s = slant range

R_{11}, R_T = linear array coordinates

$V_{xt}, V_{yt}, R_{xt}, R_{yt}, R_{zt}$ = target parameters

$V_{xo}, V_{yo}, R_{xo}, R_{yo}, R_{zo}$ = observer parameters

V_x, V_y, R_x, R_y, R_z = target-observer parameters

K = total number of measurements

n = number of unknowns

a_m = the value of a_i which minimizes the quadratic polynomial

$A = (J_i^T f)$

UNDERWATER TARGET TRACKING USING MEASUREMENTS FROM A LINEAR ARRAY OF OMNIDIRECTIONAL ACOUSTIC ELEMENTS

1. INTRODUCTION

The tracking problem discussed herein is defined by a moving observer in an underwater environment which uses measurements from a trailing linear acoustic array to estimate the initial position and velocity of a target. The assumptions made in discussing this problem are as follows. The array is a multispot linear array of uniformly-spaced omnidirectional elements operating in a passive mode; it is considered to be an inflexible body which is rigidly connected to the observer's stern. The target-observer kinematics are described in a three-dimensional Cartesian coordinate system. Both the velocity and depth of the target are constant throughout the problem. The observer maneuvers to improve the quality of the estimate and these maneuvers are generated by impulses in acceleration and are restricted to a horizontal plane at a depth that may differ from the target's. The underwater environment is modeled as an infinite, homogeneous medium with the target being the only noise source in this medium. Finally, it is assumed that the linear array is operating in the far-field region, so that only plane-wave propagation is considered.

Various versions of this target tracking problem are produced by changing the type of sensor, altering the restrictions on the environment, and relaxing the restrictions on target motion and other assumptions. One well-studied version is the bearings-only case, where both the target and observer are operating at the same depth and a bow-mounted sonar measures the azimuthal angle between north and the line-of-sight to the target. In other words, by steering in a horizontal plane, this sonar isolates a line segment on which the target lies. In contrast with this, the linear array measures a conical angle between the array's longitudinal center line and the surface of a cone on which the target lies.

The measurement equation describing this conical angle is a nonlinear function of the target parameters: velocity and initial position. Consequently, an iterative least-squares estimator is used. Although the extended Kalman filter is also applicable, it was not tried because of its erratic behavior with the Cartesian model of bearings-only target motion analysis,¹ which has many of the same properties as the omnidirectional linear array problem. Also, the additional experience with iterative techniques may prove useful in treating other problems where past measurements from secondary sensors must be incorporated into the current estimate. The measurement noise is assumed to be Gaussian even though the iterative least-squares estimator does not require any information about the noise structure.

The modified Gauss-Newton method² is the particular iterative technique used in this study. However, its unmodified version, the Gauss-Newton method, is also discussed and some results obtained with it are cited. The performance index, PII, for this nonlinear least-squares problem is the sum of the squared measurement residuals, which is minimized by iteratively estimating the target parameters. The previous estimate defines a nominal target track which is used to formulate a linearized least-squares problem that is solved by a Householder transformation.³ This solution updates the previous estimate, and the procedure repeats until the terminating criteria are satisfied. The modified Gauss-Newton method is used because of its elementary structure, and the Householder transformation is used because of its numerical accuracy.

The next section of this report develops the Cartesian signal model required by the estimation algorithm. This is followed by a section that describes the Gauss-Newton method, the modified Gauss-Newton method, and the Householder transformation. The Levenberg-Marquardt method⁴ is also mentioned because of its success with singular or ill-conditioned

problems. The experimental results are given in the fourth section, and the conclusions and recommendations for further work are given in the last section.

Appendix A shows the Jacobian matrix required in the Gauss-Newton method, and appendix B shows a scheme for selecting the step size in the modified Gauss-Newton method. The number of computer operations (additions, multiplications, divisions, square roots, and trigonometric functions) required for one iteration of the modified Gauss-Newton method is given in appendix C. A listing of the computer code used to define the experiments is contained in appendix D.

2. SIGNAL MODEL

The signal model is defined by the measurement process and the target-observer kinematics, and both of these are described in a three-dimensional Cartesian coordinate system in this section. The complexity of this description is reduced because neither the target nor the observer changes depth. A further simplification results from the assumption on the observer's maneuvers; i.e., instantaneous changes in speed and heading are permitted. Consequently, the center lines of the observer and the linear array are always coincident. This gives a shorter convergence time when compared with the results from a signal model that uses a realistic description of the maneuvers.

Figure 1 shows the two-element model of the linear array used in this report. The distance between elements is b , and the distance traveled by the plane wave as it moves from the first to the second element is d . The direction to the target is given by the angle α , which should be defined at the midpoint between the two elements. This error is neglected, as is the offset between the positions of the array and the observer.

Inspection of figure 1 shows that

$$\cos \alpha = d/b = c t_1/b, \quad (2-1)$$

where c is the nominal speed of sound in sea water and t_1 is the wave front time delay between elements. The observer reconstructs the plane wave by combining the electrical signals from these two elements after first inserting the time delay, t_1 , into the signal path of the second element. Further inspection of figure 1 shows that α and, consequently, t_1 are unchanged when the direction of propagation line is rotated

about the array's longitudinal center line. Hence, a time delay measurement defines the surface of a cone on which the target lies. This is shown in figure 2.

In this report t_1 rather than α is defined as the measured variable because $\cos \alpha$ is the required input signal for the estimation algorithm presented in the next section. Assuming that α is the measured variable, the noise level on t_1 is established from equation (2-1). The noisy version of this equation is

$$\cos(\alpha + v_\alpha) = c(t_1 + v_t)/b \quad (2-2)$$

where v_α and v_t are the noise terms. Expanding the cosine term and simplifying the result gives

$$v_t = v_\alpha (b/c) \sin \alpha \quad (2-3)$$

Assuming that $v_\alpha \sim N(0, \sigma_\alpha^2)$ and that at any point in time α is unknown but constant over an ensemble of experiments, then $v_t \sim N(0, \sigma_t^2)$ where

$$\sigma_t = b |\sin \alpha| \sigma_\alpha / c. \quad (2-4)$$

Hereafter,

$$\sigma_t = b \sigma_\alpha / c \quad (2-5)$$

is used. This approximation leads to a pessimistic result.

The noisy time delay is defined as t_m ; i.e.,

$$t_m = t_1 + v \quad (2-6)$$

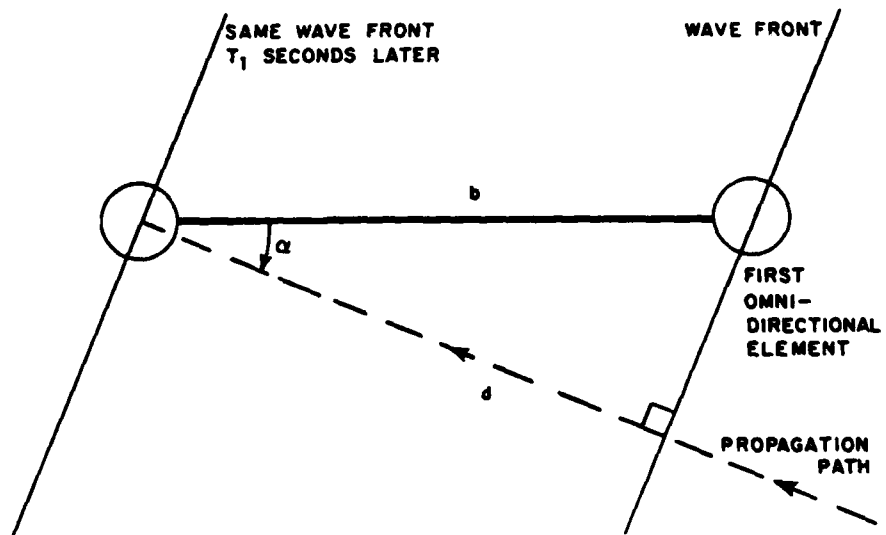


Figure 1. A Plane Wave Arriving at a Two-Element Omnidirectional Linear Array

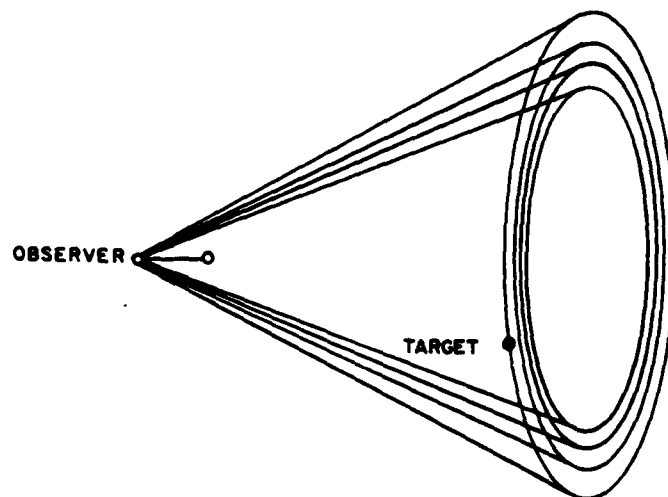


Figure 2. The Surfaces of Four Cones, Each for a Different Time Delay

where the t subscript on v has been dropped for brevity. Solving equation (2-1) for t_1 and substituting this result into equation (2-6) yields

$$t_m(k) = (b \cos \alpha(k))/c + v(k) \quad (2-7)$$

where k denotes the time $t = kh$ at which the measurement is taken with a fixed sampling period h seconds. Equation (2-7) defines the measured time delay as a function of the conical angle α . To define it as a function of the target's position variables, a Cartesian coordinate system is introduced.

A (R_{11}, R_T, R_Z) coordinate system fixed at the array with R_{11} along the center line, R_Z in the depth direction, and R_T orthogonal to both R_{11} and R_Z is shown in figure 3. Inspection of this figure shows that

$$\cos \alpha = R_{11}/R_S \quad (2-8)$$

where

$$R_S = (R_{11}^2 + R_T^2 + R_Z^2)^{1/2} \quad (2-9)$$

is the slant range to the target. Substituting equation (2-8) into equation (2-7) gives

$$t_m(k) = b R_{11}(k)/c R_S(k) + v(k), \quad (2-10)$$

which shows the measurement as a function of the target's relative position variables. Also, from figure 3, it is easily shown that

$$R_{11}^2 - R_s^2 \cos^2 \alpha = 0 \quad (2-11)$$

defines the surfaces of the cone on which the target lies.

Because the (R_{11}, R_T) plane rotates when the observer changes course, these variables must be transformed to a north-referenced coordinate system where the constant velocity target is defined. The new variables are R_y (which points north) and R_x (which is orthogonal to both R_y and R_z). The (R_x, R_y) and (R_{11}, R_T) planes are coincident but differ by a rotation, and R_z is the same in both coordinate systems. Both of these coordinate systems are fixed at the array.

This rotation is defined by the transformation shown in the next equation.

$$\begin{bmatrix} R_x \\ R_y \end{bmatrix} = \begin{bmatrix} \cos c_{11} & \sin c_{11} \\ -\sin c_{11} & \cos c_{11} \end{bmatrix} \begin{bmatrix} R_T \\ R_{11} \end{bmatrix} \quad (2-12)$$

where c_{11} is the angle between the R_y and the R_{11} axes, measured positive clockwise from the north reference. Because this is an orthogonal transformation, R_s is also given by

$$R_s = (R_x^2 + R_y^2 + R_z^2)^{1/2} \quad (2-13)$$

Figure 4 shows these two Cartesian coordinate systems, the array, and the observer's heading angle.

This transformation is expressed in terms of the observer's heading angle, ψ , by substituting

$$c_{11} = \psi - 180^\circ \quad (2-14)$$

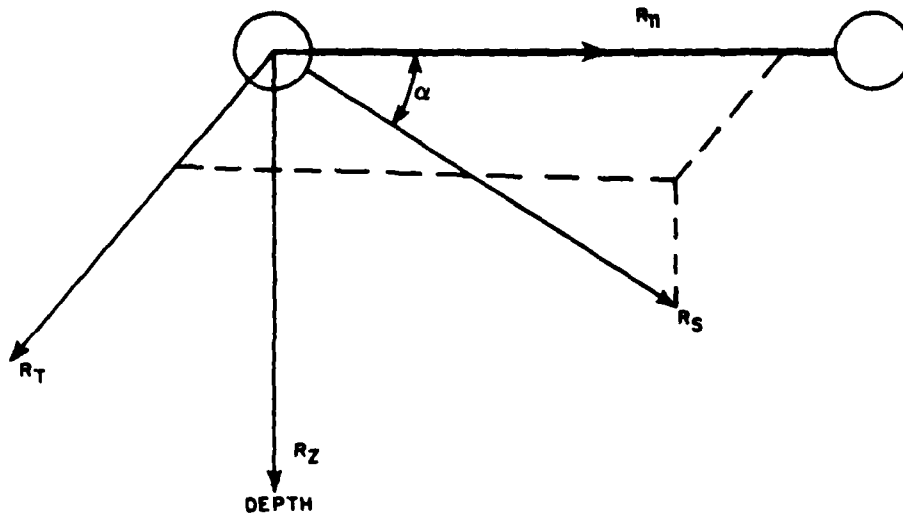


Figure 3. The (R_{11}, R_T, R_Z) Coordinates System Fixed at the Array

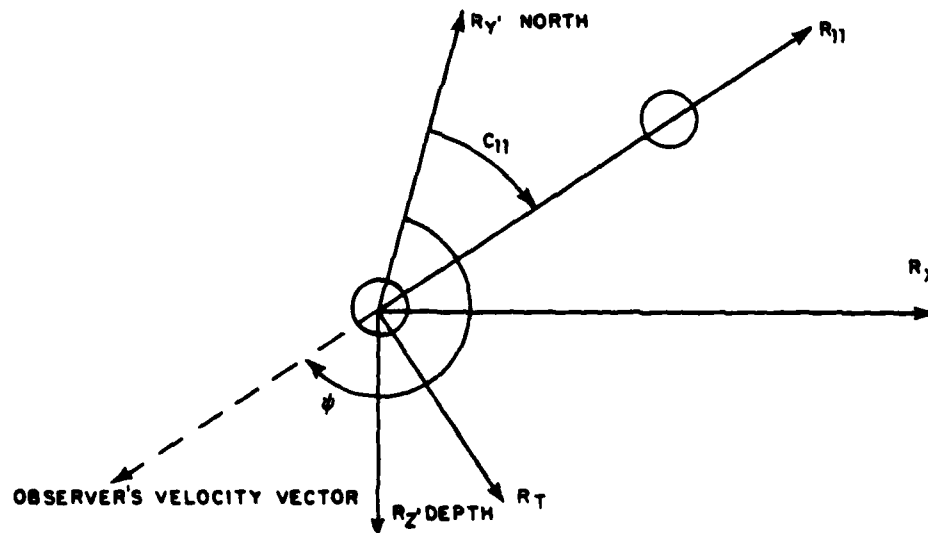


Figure 4. Two Coordinate Systems Fixed at the Array

into equation (2-12). After first multiplying equation (2-12) by the inverse transformation, this gives

$$\begin{bmatrix} R_T \\ R_{11} \end{bmatrix} = \begin{bmatrix} -\cos \psi & \sin \psi \\ -\sin \psi & -\cos \psi \end{bmatrix} \begin{bmatrix} R_x \\ R_y \end{bmatrix}. \quad (2-15)$$

Using the last equation in the matrix equation (2-15) to eliminate R_{11} from equation (2-10) gives

$$t_m(k) = -b(R_x(k)\sin \psi(k) + R_y(k)\cos \psi(k))/c R_s(k) + v(k) \quad (2-16)$$

where R_s is given by equation (2-13). Equation (2-16) shows the measured variable as a function of relative position variables which are defined in a north-referenced system.

Inspection of equation (2-16) shows that the sign on R_z cannot be resolved by this measurement. This was demonstrated by simulation, although this elementary experiment is not included in this report.

For a constant velocity target, these relative position variables are defined by

$$R_x(k) = V_{xt}kh + R_{xt}(0) - R_{x0}(k) \quad (2-17a)$$

$$R_y(k) = V_{yt}kh + R_{yt}(0) - R_{y0}(k) \quad (2-17b)$$

$$R_z(k) = R_{zt}(0) \quad (2-17c)$$

where V_{xt} , V_{yt} , $R_{xt}(0)$, $R_{yt}(0)$, and $R_{zt}(0)$ define the target parameters in another north-referenced coordinate system which is fixed in

space and which at $t = 0$ is coincident with the north-referenced system fixed to the observer. Also, $R_{x_0}(k)$ and $R_{y_0}(k)$ define the observer's position in this stationary coordinate system.

Equation (2-16) defines the measurement process and equation (2-17) defines the target-observer kinematics. Together, these equations define the signal model required by the estimation algorithm presented in the next section.

3. ITERATIVE LEAST-SQUARES TECHNIQUE

Because the measurement equation (2-16) is nonlinear, an iterative least-squares algorithm which batch processes a set of measurements is selected to estimate the target parameters. The extended Kalman filter could be used for this problem, but it was not tried because its performance with a Cartesian signal model of the bearings-only problem is suspect.

In this least-squares problem the modified Gauss-Newton method is used to iteratively minimize the sum of the squared measurement residuals. At each iteration the Gauss-Newton equations are solved by a Householder transformation, and this yields an increment or step which is added to the previous estimate. The modified Gauss-Newton method first multiplies this step by a positive scalar, which is selected at each iteration. The Levenburg-Marquardt method is also mentioned in this section; however, it did not demonstrate any advantage in the experiments, so its use is not recommended at this time. These experiments are not contained in this report.

The sum of the squared measurement residuals which defines the performance index PII is given by

$$PII(x) = ||f(x)||^2 = f^T(x)f(x) = \sum_{k=0}^{K-1} f_k^2(x) \quad (3-1)$$

where $K > 5$, $x^T = (V_{xt}, V_{yt}, R_{xt}(0), R_{yt}(0), R_{zt}(0))$,

$$f^T(x) = (f_0(x), f_1(x), \dots, f_{K-1}(x)), \text{ and}$$

$$f_k(x) = (R_x(k)\sin \psi(k) + R_y(k)\cos \psi(k)) + c R_s(k)t_m(k)/b \quad (3-2)$$

In this section x represents the estimate of the target parameters. Equation (3-2) defines the k th measurement residual. With K finite, it is expected that the true target parameters can minimize (3-1) only when the measurements are noiseless.

A general iterative procedure for minimizing PII is given by

$$x_{i+1} = x_i + g_i \quad (3-3)$$

where x_{i+1} is the $(i+1)$ th estimate, x_i is the i th estimate, and g_i is the i th step. In theory, the initial estimate x_0 is usually selected as zero. Unfortunately, this did not work well, so the first measurement is used to aid in the selection of x_0 . The procedure terminates in a finite number of iterations when no further significant improvement in PII occurs and/or when $\|x_{i+1} - x_i\|$ is less than \max some threshold. A requirement for convergence of this procedure using the modified Gauss-Newton method is given in reference 5. Although convergence proofs are beneficial, they do not guarantee the performance of the algorithm. Even though convergence is assured, double precision arithmetic and/or an excessive number of iterations may be needed.

The linearized least-squares problem which is solved at each iteration is derived from the following Taylor series expansion of PII:

$$PII(x_{i+1}) = PII(x_i + g_i) = PII(x_i) + \nabla PII^T(x_i)g_i + \text{higher-order terms} \quad (3-4)$$

where ∇ is the gradient operator. If x_{i+1} yields a local minimum, then $\nabla PII(x_{i+1}) = 0$, and the gradient of equation (3-4) gives

$$0 = \nabla PII(x_i) + \nabla^2 PII(x_i)g_i \quad (3-5)$$

where the higher-order terms have been neglected. The step g_i to the next estimate is given by the solution to the matrix equation

$$\nabla^2 P_{II}(x_i) g_i = -\nabla P_{II}(x_i) \quad (3-6)$$

where

$$\nabla P_{II}(x) = a f^T(x) f(x)/ax = 2 J^T(x) f(x) \quad (3-7)$$

and

$$\nabla^2 P_{II}(x) = 2 J^T(x) J(x) + 2 \sum_{k=0}^{K-1} (\nabla^2 f_k(x)) f_k(x). \quad (3-8)$$

$J(x)$, the K by 5 Jacobian matrix of $f(x)$, is given by

$$J(x) = a f(x)/ax = [a f_k/a x_i], \quad (3-9)$$

which is given in appendix A. This appendix shows that the last column of $J(x)$ is zero if the estimate of the target's depth is zero. This singularity did not present any difficulties for the experiments reported here; it will be discussed in detail in a future report.

The computer burden is reduced if the last term in equation (3-8) is replaced by an approximation which does not require any second derivatives. Because this term depends upon the residuals, the approximation usually introduces negligible error near the solution. Various iterative techniques result from different approximations. Using equation (3-8) as stated gives Newton's method, while dropping the last term gives the Gauss-Newton method. In addition, setting $\nabla^2 P_{II}(x) = I$, an identity matrix, gives a steepest decent step $g_i = -\nabla P_{II}(x_i)$. The

Levenberg-Marquardt method provides a mix of these latter two techniques. Matrix-updating methods, such as the Davidon-Fletcher-Powell method, approximate the Hessian matrix, $\nabla^2 \text{PI}$, by a successive addition of low-rank matrices.

The convergence rate for Newton's method is second order, while reference 5 states that at most the convergence rate for the Gauss-Newton method is linear unless $\text{PI}(x_{\min}) = 0$, in which case it is ultimately second order.

Dropping the last term in equation (3-8) gives

$$J^T(x_i) J(x_i) g_i = -J^T(x_i) f(x_i), \quad (3-10)$$

which defines the normal equations for the Gauss-Newton linearization. Solving these equations for g_i minimizes a linearized performance index given by

$$\text{PI2}(g_i) = \|J(x_i) g_i - (-f(x_i))\|^2. \quad (3-11)$$

These normal equations (3-10) are solved by applying a Householder transformation to the squared error shown in equation (3-11). This numerically more accurate approach⁶ is developed by introducing $T^T T$ into equation (3-11) as follows:

$$\begin{aligned} \text{PI2}(g_i) &= (J(x_i) g_i + f(x_i))^T T^T T (J(x_i) g_i + f(x_i)) \\ \text{PI2}(g_i) &= (T J(x_i) g_i + T f(x_i))^T (T J(x_i) g_i + T f(x_i)) \\ \text{PI2}(g_i) &= \|T J(x_i) g_i + T f(x_i)\|^2 \end{aligned} \quad (3-12)$$

where T is the Householder transformation. Because this is an orthogonal transformation, $T^T T = I$ and inserting this into equation (3-11) leaves $PI2$ unchanged. In addition, T is selected such that

$$T J = \begin{pmatrix} R \\ 0 \end{pmatrix} \quad (3-13)$$

where R is a 5 by 5 upper triangular matrix and 0 is a $(k-5)$ by 5 zero matrix. This gives

$$PI2 = \left\| \begin{pmatrix} R \\ 0 \end{pmatrix} g_i + \begin{pmatrix} \bar{f} \\ e \end{pmatrix} \right\|^2 \quad (3-14)$$

where

$$T f = \begin{pmatrix} \bar{f} \\ e \end{pmatrix}. \quad (3-15)$$

The g_i which minimizes $PI2$ is given by

$$g_i = -R^{-1} \bar{f} \quad (3-16)$$

and

$$\min PI2 = \|e\|^2. \quad (3-17)$$

The details associated with defining the elements of this transformation T are given in references 3, 6, and 7.

In the vicinity of a local minimum at x_{i+1} ,

$$f(x_{i+1}) = f(x_i + g_i) = f(x_i) + J(x_i) g_i \quad (3-18)$$

and, consequently, (3-11) becomes

$$PI2(g_i) = ||f(x_{i+1})||^2 = PII(x_{i+1}). \quad (3-19)$$

Equation (3-19) is not a sufficient condition for a local minimum, but it is convenient to use it as the termination criterion for the experiments.

The modified Gauss-Newton method is given by

$$x_{i+1} = x_i + a_i g_i \quad (3-20)$$

where the positive scalar a_i is selected such that

$$PII(x_{i+1}) < PII(x_i). \quad (3-21)$$

In this method it is said that g_i gives the direction of the step while a_i gives its length.

Once g_i is found, by the Gauss-Newton method, $PII(x_{i+1}) = PII(x_i + a_i g_i)$ is only a function of the scalar a_i ; therefore, this function can be minimized by the selection of a_i . There are several schemes for selecting a_i . The one used in this report finds the minimum of a quadratic polynomial which fits the $(a_i, PI2)$ data in the g_i direction. This is generated by computing $PI2$ for three different values of a_i . An analytical expression for the minimum in terms of these three data points is derived in appendix B.

Inspection of equation (3-10) shows that a non-singular $J^T J$ matrix is required by the Gauss-Newton method. Ill-conditioning of this matrix is also a concern. The Levenberg-Marquardt method is useful in these situations, and it also provides a way to incorporate a priori information into the problem.

The Levenberg-Marquardt method is defined by equation (3-5) but with

$$\nabla^2 P_{II}(x_i) = 2 J^T(x_i) J(x_i) + 2 \lambda_i D \quad (3-22)$$

where λ_i is a positive scalar and D is a positive definite diagonal matrix. By comparing equation (3-22) with equation (3-8) it is seen that $2 \lambda_i D$ may be viewed as an approximation to the term that is omitted by the Gauss-Newton method. The details of the selection procedure for both λ_i and D are given in reference 4, but inspection of equation (3-22) shows that $\lambda_i = 0$ yields the Gauss-Newton method while $\lambda_i \rightarrow \infty$ with $D = I$ yields steepest descent.

In the experiments reported in the next section, no difficulties with the $J^T J$ matrix occurred, so the Levenberg-Marquardt method did not prove useful. These experimental results are not shown in this report. The Levenberg-Marquardt method is discussed because the omnidirectional linear array problem is singular prior to an observer maneuver, and estimates during this time interval may be required.

4. EXPERIMENTAL RESULTS

The results from a digital computer simulation of three different sets of experiments are presented in this section. The different sets are produced by changing the target's initial position. Each of these sets is composed of three members, which are produced by changing the standard deviation of the measurement noise. The target parameters for these three sets of experiments are given in table 1.

Table 1. True Target Parameters with Speed in Meters/Second and Range in Meters

Experiment Set	V_{xt}	V_{yt}	$R_{xt}(0)$	$R_{yt}(0)$	$R_{zt}(0)$
No. 1	-5	10	10,000	10,000	1000
No. 2	-5	10	40,000	40,000	0
No. 3	-5	10	10,000	10,000	10,000

Experiment set No. 1 is a short-to-mid range situation while No. 2 is a mid-to-long range one. Experiment set No. 3 may be viewed as a single-bottom-bounce problem where the actual target depth is located by knowing the bottom depth. This is shown in figure 5.

The standard deviation for the white Gaussian measurement noise is established from equation (2-5) with $c = 1500$ meters/second and $b = 3/8$ meter (which is a quarter wave-length at a 1-kHz frequency). The three

standard deviations used in each of the sets shown in table 1 are $\sigma_t = 0.0$ seconds, $\sigma_t = 0.05 \times 10^{-5}$ seconds, and $\sigma_t = 0.25 \times 10^{-5}$ seconds. The first of these is the noiseless case while the last corresponds to a $\sigma_\alpha = 0.01$ radians (approximately 0.5 degrees). The remaining σ_t represents a low-noise case.

In all experiments the observer starts from the origin at $t = 0$ seconds with a speed of 10 meters/second and a heading of 0 degrees. Just prior to the measurement at 240 seconds the observer instantaneously changes heading to 90 degrees and just prior to 480 seconds changes back to 0 degrees. Each of these headings defines a leg of the problem, and each leg consists of 240 measurements. The first measurement occurs at 0 seconds while the last occurs at 719 seconds. With the exception of the target's initial position, the target and observer tracks are the same for all experiments. Figure 6 shows these tracks projected onto the (R_y, R_x) plane.

The time-delay measurements are taken at a 1-second data rate; they are then compressed by a measurement preprocessor, which also reduces the noise. For example, this preprocessor produces a "measurement" at 9.5 seconds by first adding together the measurements from 0 to 19 seconds and then dividing this sum by 20. These preprocessed measurements are then fed to the estimation algorithm. Hereafter, the indices of all the variables denote the preprocessed measurement count; e.g., $t_m(0)$ is the first of these measurements and it is associated with a real time of 9.5 seconds while $t_m(1)$ is the second, and real time is 29.5 seconds. The error introduced by measurement preprocessing is not modeled in these experiments. This error is zero only if the true time delay is constant for each of the 20-second intervals. Finally, the time-delay preprocessor averages $\cos \alpha$, not α itself.

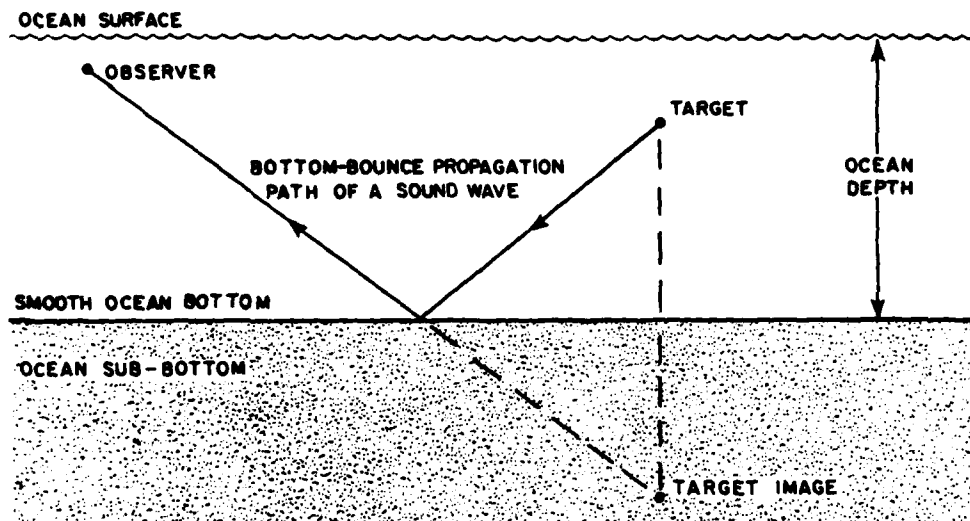


Figure 5. A Single Bottom-Bounce Propagation Path

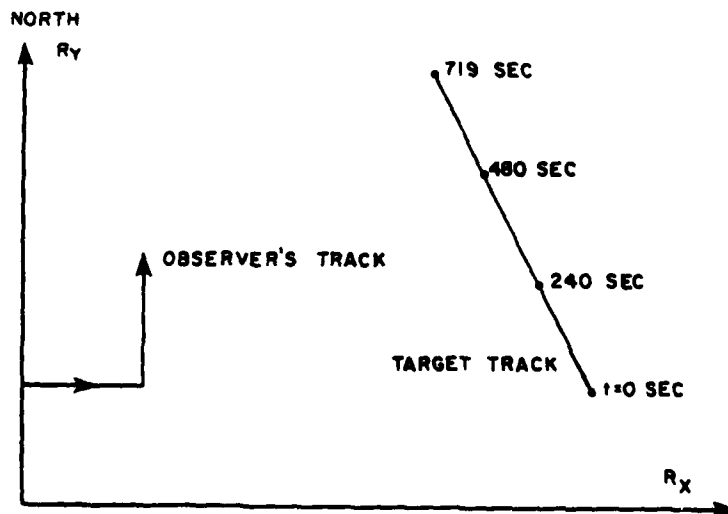


Figure 6. Target and Observer Tracks in the (R_y , R_x) Plane

To start the iterative least-squares algorithm an initial estimate of the target parameters is required. This is specified by using the first measurement together with $V_{xt} = 0$, $V_{yt} = 0$, $R_{zt}(0) = 3,000$ meters, and $R_s(0) = 20,000$ meters; this gives,

$$R_{11}(0) = c R_s(0) t_m(0)/b \quad (4-1)$$

and

$$R_T(0) = \pm (R_s^2(0) - R_{zt}^2(0) - R_{11}^2(0))^{1/2} \quad (4-2)$$

where the sign on $R_T(0)$ determines the initial side of the cone's surface on which the target is assumed to be lying. Both signs are tried and the one which yields the smaller P11 is selected as the initial point. Although this does require that the estimate of the target-observer track be generated twice, it does not require a Householder transformation.

Another initialization scheme is to select $R_T(0) = 0$ and

$$R_{zt}(0) = \pm (R_s^2(0) - R_{11}(0))^{1/2}, \quad (4-3)$$

where either sign may be used without penalty. This scheme may also be used whenever the quantity under the square-root operator in equation (4-2) is negative. A future report will investigate the use of equation (4-3) and $R_T(0) = 0$ as the primary initialization procedure.

In these experiments the iterative procedure is terminated when $PI1(x_{i+1})$ is within ± 10 percent of $PI2(g_1)$. This criterion is selected as a convenience for these initial experiments. It is recommended that a similar bound be placed upon the component of x_{i+1} , which undergoes the largest percent change. Monitoring the percent change in P11 is also recommended.

The first experiments use the modified Gauss-Newton method. The initialization uses the sign on $R_T(0)$ which produces the smaller PII, so the estimates associated with generating the additional target-observer track are not shown in the results. However, some additional experiments do show the effects of using the "wrong" sign on $R_T(0)$. This is followed by some results for both a one-legged and a two-legged problem. Finally, results obtained with the Gauss-Newton method are mentioned.

The results from the first experiment set are shown in tables 2a, 2b, and 2c; from the second in tables 3a, 3b, and 3c; and from the third in tables 4a, 4b, and 4c. In all of these tables the iteration number defines the number of calls to the Householder transformation subroutine, and the last row shows the PII which occurs when the true target parameters are used. As expected, this PII is larger than the one reached by the algorithm in all cases.

Inspection of all these tables shows that the algorithm converges by the fourth iteration, except for the noiseless cases. In these cases it is seen that $R_{zt}(0)$ is the only target parameter which experiences more than a 10 percent decrease in moving from the third to the fourth iteration, and this only occurs in tables 2a and 3a. Perhaps, after the fourth iteration a 1-dimensional search or a less stringent performance bound should be used for $R_{zt}(0)$. Regardless, further investigation or discussion of these noiseless cases is not warranted.

Inspection of tables 2a, 2b, and 2c shows that the magnitude of the error in the final estimate increases, as expected, when the variance of the noise is increased. This is also shown in the other sets of tables. A Monte Carlo simulation or an analytical study could quantify this observation.

Table 2a. Results from the Modified Gauss-Newton Method
with $\sigma_t = 0.0$ Seconds (First Member of Set No. 1)

Iteration Number	V_{xt}	V_{yt}	$R_{xt}(0)$	$R_{yt}(0)$	$R_{zt}(0)$	PI1	PI2
0	0	0	13,800	14,161	3000	3.79×10^8	4.70×10^5
1	2.75	10.25	6179	6993	3087	1.89×10^7	8463
2	-2.80	12.99	8742	8563	1701	5.87×10^5	46.47
3	-5.39	9.64	10,230	10,233	1127	3561	0.024
4	-4.99	10.01	9997	9997	1006	4.58	0.057
5	-5.00	10.00	10,002	10,002	1002	0.059	----
True Parameters	-5.00	10.00	10,000	10,000	1000	0.57	----

Table 2b. Results from the Modified Gauss-Newton Method with
 $\sigma_t = 0.05 \times 10^{-5}$ Seconds (Second Member of Set No. 1)

Iteration Number	V_{xt}	V_{yt}	$R_{xt}(0)$	$R_{yt}(0)$	$R_{zt}(0)$	PI1	PI2
0	0	0	13,827	14,136	3000	3.83×10^8	4.75×10^5
1	2.70	10.25	6178	6993	3109	1.86×10^7	8096
2	-3.05	12.89	8899	8699	1734	5.88×10^5	763
3	-5.56	9.53	10,328	10,326	1165	4014	734
4	-5.18	9.89	10,106	10,100	1054	737	----
True Parameters	-5.00	10.00	10,000	10,000	1000	1029	----

Table 2c. Results from the Modified Gauss-Newton Method with $\sigma_t = 0.25 \times 10^{-5}$ Seconds (Third Member of Set No. 1)

Iteration Number	V _{xt}	V _{yt}	R _{xt} (0)	R _{yt} (0)	R _{zt} (0)	PI1	PI2
0	0	0	13,801	14,161	3000	3.79x10 ⁸	5.02x10 ⁵
1	2.98	10.00	6285	7058	2939	2.38x10 ⁷	42,580
2	-1.22	13.52	7637	7572	1526	6.52x10 ⁵	29,736
3	-4.24	10.47	9494	9451	858	37,858	29,648
4	-3.75	10.84	9191	9166	613	29,739	----
True Parameters	-5.00	10.00	10,000	10,000	1000	37,793	----

Table 3a. Results from the Modified Gauss-Newton Method with $\sigma_t = 0.0$ Seconds (First Member of Set No. 2)

Iteration Number	V _{xt}	V _{yt}	R _{xt} (0)	R _{yt} (0)	R _{zt} (0)	PI1	PI2
0	0	0	13,812	14,150	3000	5.51x10 ⁷	22,182
1	17.56	30.92	26,420	26,916	4659	3.64x10 ⁶	2041
2	-5.74	10.06	41,391	41,203	2628	1.37x10 ⁵	0.148
3	-4.85	9.99	39,729	39,766	1013	3813	0.148
4	-5.02	9.99	40,037	40,032	454	98	0.140
5	-5.00	10.00	39,995	39,996	178	3	0.142
6	-5.00	10.00	40,003	40,003	101	0.173	0.137
7	-5.00	10.00	40,004	40,004	89	0.163	0.134
8	-5.00	10.00	40,004	40,004	87	0.134	----
True Parameters	-5.00	10.00	40,000	40,000	0.0	missing	

Table 3b. Results from the Modified Gauss-Newton Method with $\sigma_t = 0.05 \times 10^{-5}$ Seconds (Second Member of Set No. 2)

Iteration Number	V_{xt}	V_{yt}	$R_{xt}(0)$	$R_{yt}(0)$	$R_{zt}(0)$	PI1	PI2
0	0	0	13,806	14,156	3000	5.46×10^7	30,848
1	17.60	30.94	26,416	26,921	4668	3.62×10^6	25,604
2	-5.70	9.92	41,177	41,016	2663	1.54×10^5	23,343
3	-4.83	9.92	39,558	39,616	1131	26,524	23,343
4	-4.98	9.91	39,820	39,844	754	23,378	----
True Parameters	-5.00	10.00	40,000	40,000	0.0	24,205	----

Table 3c. Results from the Modified Gauss-Newton Method with $\sigma_t = 0.25 \times 10^{-5}$ Seconds (Third Member of Set No. 2)

Iteration Number	V_{xt}	V_{yt}	$R_{xt}(0)$	$R_{yt}(0)$	$R_{zt}(0)$	PI1	PI2
0	0	0	13,757	14,204	3000	5.04×10^7	2.03×10^5
1	19.54	32.84	25,919	26,460	4549	3.83×10^6	3.31×10^5
2	-7.51	7.22	41,133	41,199	2844	4.00×10^5	3.04×10^5
3	-6.94	7.14	40,120	40,346	1880	3.06×10^5	----
True Parameters	-5.00	10.00	40,000	40,000	0.0	3.77×10^5	----

Table 4a. Results from the Modified Gauss-Newton Method with $\sigma_t = 0.0$ Seconds (First Member of Set No. 3)

Iteration Number	V_{xt}	V_{yt}	$R_{xt}(0)$	$R_{yt}(0)$	$R_{zt}(0)$	PI1	PI2
0	0	0	16,039	11,565	3000	8.69×10^8	3.91×10^5
1	-1.66	1.58	13,726	11,045	10,376	4.63×10^8	1.49×10^5
2	-6.58	6.73	9975	10,586	10,905	1.34×10^7	1163
3	-5.16	10.11	10,196	10,112	10,059	47,132	0.0066
4	-5.00	10.00	10,000	10,000	10,001	0.522	0.0073
5	-5.00	10.00	9999	10,000	10,000	0.0061	0.0061
6	-5.00	10.00	9999	10,000	10,000	0.0060	----
True Parameters	-5.00	10.00	10,000	10,000	10,000	0.261	----

Table 4b. Results from the Modified Gauss-Newton Method with $\sigma_t = 0.05 \times 10^{-5}$ Seconds (Second Member of Set No. 3)

Iteration Number	V_{xt}	V_{yt}	$R_{xt}(0)$	$R_{yt}(0)$	$R_{zt}(0)$	PI1	PI2
0	0	0	16,036	11,569	3000	8.68×10^8	3.80×10^5
1	-1.67	1.58	13,730	11,049	10,375	4.63×10^8	1.42×10^5
2	-6.55	6.75	9950	10,558	10,886	1.39×10^7	2579
3	-5.00	10.12	10,083	10,017	9980	49,698	1815
4	-4.86	10.01	9900	9915	9931	1816	----
True Parameters	-5.00	10.00	10,000	10,000	10,000	2003	----

Table 4c. Results from the Modified Gauss-Newton Method with $\sigma_t = 0.25 \times 10^{-5}$ Seconds (Third Member of Set No. 3)

Iteration Number	V_{xt}	V_{yt}	$R_{xt}(0)$	$R_{yt}(0)$	$R_{zt}(0)$	PI1	PI2
0	0	0	16,065	11,529	3000	8.77×10^8	4.92×10^5
1	-1.70	1.54	13,718	10,995	10,364	4.65×10^8	2.28×10^5
2	-6.87	6.54	10,059	10,609	10,890	1.44×10^7	55,891
3	-5.67	9.97	10,440	10,247	10,152	1.01×10^5	53,939
4	-5.46	9.86	10,213	10,112	10,072	53,937	----
True Parameters	-5.00	10.00	10,000	10,000	10,000	60,327	----

Comparing tables 2c, 3c, and 4c shows that the final value of PI1 increases with increasing slant range. The results shown in these tables demonstrate that the modified Gauss-Newton method converges to a reasonable estimate of the target parameters in approximately four iterations.

The high noise cases for these three experiment sets were repeated, but they were initialized on the "wrong" side of the cone; i.e., the side which yields the larger PI1. The one from experiment set no. 3 displayed results similar to those reported in table 4c while the remaining two high noise cases performed poorly. After five iterations the one from experiment set no. 2 was stopped because convergence to a reasonable estimate was judged unlikely. After seven iterations the one from experiment set no. 1 was also stopped; however, convergence in one more iteration was expected but, unfortunately, with the wrong sign on the $R_{zt}(0)$ estimate.

To investigate the effects of fewer legs, the third experiment set shown in table 1 was repeated, first with two legs (0 to 479 seconds) and then with one (0 to 239 seconds). Table 5 shows the final results of the two-legged problem.

Table 5. Estimates from the First Two Legs of Experiment Set No. 3

Number of Iterations	σ_t	V_{xt}	V_{yt}	$R_{xt}(0)$	$R_{yt}(0)$	$R_{zt}(0)$	PI1
5	0.0	-5.0	10.0	10,000	10,000	10,000	0.0033
4	0.05×10^{-5}	-6.2	9.50	10,820	10,726	10,629	965.0
7	0.25×10^{-5}	10.0	2.55	-2495	1773	-90	852.0

This table shows that the noiseless and the low noise cases converged to reasonable estimates while the high noise case converged to a worthless result. This worthless result may be caused by fewer legs or by fewer measurements; additional experiments could resolve this point. As expected, the one-legged problem generated diverging results because the $J^T J$ matrix is singular. The Levenberg-Marquardt method could be used if an estimate on this leg is required.

Finally, the Gauss-Newton method was used to treat the three high noise cases. The results obtained were essentially the same as those shown in tables 2c, 3c, and 4c; consequently, the additional computations required to select a step size are not justified by these experiments. However, these experiments do not constitute an exhaustive set of all possible target-observer situations, and the modified Gauss-Newton method may yet prove useful.

5. CONCLUSIONS AND FURTHER WORK

The main conclusions are:

1. There is a sign ambiguity on the estimate of the target's depth.
2. The problem is singular on the first leg.
3. Both the Gauss-Newton method and the modified Gauss-Newton method converge in four iterations for these experiments.
4. The Jacobian matrix is singular if the estimate of the target's depth is zero.
5. Initialization of the iterative estimator affects convergence.

The sign on the target's depth, $R_{zt}(0)$, cannot be resolved by an observer operating at a constant depth. This is seen from equation (2-16), and it was also demonstrated experimentally although these results are not shown.

The target tracking problem with a linear array of omnidirectional elements is singular on the first leg. This was shown in the experiments. An analytic treatment of this point is not a high priority item, but is recommended since it would lend additional insight into this tracking problem.

Both the Gauss-Newton and the modified Gauss-Newton methods demonstrate convergence in approximately four iterations. Additional

experiments are needed to investigate convergence and, in those situations where convergence occurs, to determine if convergence in four iterations is typical.

Inspection of appendix A shows that the last column of the Jacobian matrix is zero if the estimate of the target's depth is zero; i.e., the target is estimated to be at the same depth as the observer. This singularity will be investigated in a subsequent report.

Initialization of the estimation algorithm can reduce the number of iterations needed for convergence. Further experiments on this item are needed.

To evaluate the Gauss-Newton method, more experiments with different target-observer tracks and with different noise variances are needed. A Monte Carlo simulation is also needed.

Different PII functions should be tried, with a view towards finding one that reduces the bias in the final estimate. Also, a probabilistic description of this problem might be more useful than the least-squares approach used here.

Experiments which expand the iterative least-squares estimator to included measurements from additional sensors would be useful.

Finally, results from an extended Kalman filter which explores the effects of different signal models should be documented.

6. REFERENCES

1. V. J. Aidala, "Kalman Filter Behavior in Bearings-Only Tracking Applications," IEEE Trans. Aerosp. Electron. Syst., Vol. AES-15, pp. 29-39, January 1979.
2. G. Dahlquist and A. Bjorck, Numerical Methods, Prentice-Hall, 1974.
3. G. J. Bierman, Factorization Methods for Discrete Sequential Estimation, Academic Press, 1977.
4. Y. Bard, Nonlinear Parameter Estimation, Academic Press, 1974.
5. S. A. Lill, (L.C.W. Dixon, ed.), "A Survey of Methods for Minimizing Sums of Squares of Nonlinear Functions," Optimization in Action, Academic Press, 1976.
6. C. L. Lawson and R. I. Hanson, Solving Least Squares Problems, Prentice-Hall, 1974.
7. M. F. Gouveia and D. J. Murphy, "Bearings-Only Target Tracking With a Square-Root Filter", Southeastern Massachusetts University Technical Report, June 1, 1980.

APPENDIX A. THE JACOBIAN OF $f(x)$

The Jacobian of $f(x)$, which is used in the performance index of equation (3-1), is shown below. This $f(x)$ is defined by equation (3-2).

The Jacobian is defined by

$$J = (\partial f_k / \partial x_j)$$

for $j = 1, 2, 3, 4$, and 5 ; and $k = 0, 1, 2, \dots, 35$.

The entries in the k th row of J are given by:

$$\partial f_k / \partial x_1 = \partial f_k / \partial v_{xt} = kh \partial f_k / \partial R_{xt}(0)$$

$$\partial f_k / \partial x_2 = \partial f_k / \partial v_{yt} = kh \partial f_k / \partial R_{yt}(0)$$

$$\partial f_k / \partial x_3 = \partial f_k / \partial R_{xt}(0) = -\sin \psi(k) - ct_m(k) R_x(k) / b R_s(k)$$

$$\partial f_k / \partial x_4 = \partial f_k / \partial R_{yt}(0) = -\cos \psi(k) - ct_m(k) R_y(k) / b R_s(k)$$

$$\partial f_k / \partial x_5 = \partial f_k / \partial R_{zt}(0) = -ct_m(k) R_z(0) / b R_s(k).$$

APPENDIX B. STEP SIZE FOR THE MODIFIED GAUSS-NEWTON METHOD

In this appendix a scheme is given for selecting the step size a_i in the modified Gauss-Newton iterative formula.

$$x_{i+1} = x_i + a_i g_i \quad (B-1)$$

Actually, because it is not normalized, the direction g_i also contributes to the size of the step. It is convenient to redefine equation (B-1) as

$$x_{i+1} = x_i + a_j g_i \quad (B-2)$$

where a_j denotes the j th value of the step size at the i th iteration.

Once g_i is found from the Gauss-Newton equations, P_{i1} is a function only of a_j ; i.e.,

$$P_{i1}(a_j) = P_{i1}(x_i + a_j g_i) \quad (B-3)$$

and this is minimized by a judicious selection of a_j . In this report, a_j is defined by the minimum of a quadratic polynomial which passes through three $(a_j, P_{i1}(a_j))$ data points. For equally spaced values of a_j , the minimum of this quadratic is given by

$$a_m = \frac{(a_2 + a_3) P_{i1}(a_1) - 2(a_1 + a_3) P_{i1}(a_2) + (a_2 + a_1) P_{i1}(a_3)}{2 P_{i1}(a_1) - 4 P_{i1}(a_2) + 2 P_{i1}(a_3)} \quad (B-4)$$

where $a_3 > a_2 > a_1 \geq 0$.

The first of these data points is readily available, namely, $(a_1 = 0, P_{i1}(0) = P_{i1}(x_i))$; and if

$$PII(1) < PII(0), \quad (B-5)$$

$a_2 = 1$ gives the second data point and $a_3 = 2a_2$ gives the third. However, if equation (B-5) is not satisfied the length of the interval is reduced by selecting $a_2 = 1/2$ and $a_3 = 2a_2 = 1$, provided

$$PII(1/2) < PII(0). \quad (B-6)$$

If this is unsuccessful, the next selection is $a_2 = 1/4$ and $a_3 = 2a_2 = 1/2$, and subsequent selections are given by repeatedly dividing a_2 by 2. This continues until $PII(a_2) < PII(a_1)$ or a threshold which causes termination of the estimation algorithm is crossed. After a_m is found, then $PII(a_m)$, $PII(a_2)$, and $PII(a_3)$ are compared to determine which of these is the smallest. This is necessary because the quadratic polynomial may not always provide a good fit to the PII function and $PII(a_2)$ or $PII(a_3)$ may be smaller than $PII(a_m)$. A diagram of this scheme is shown in figure B-1.

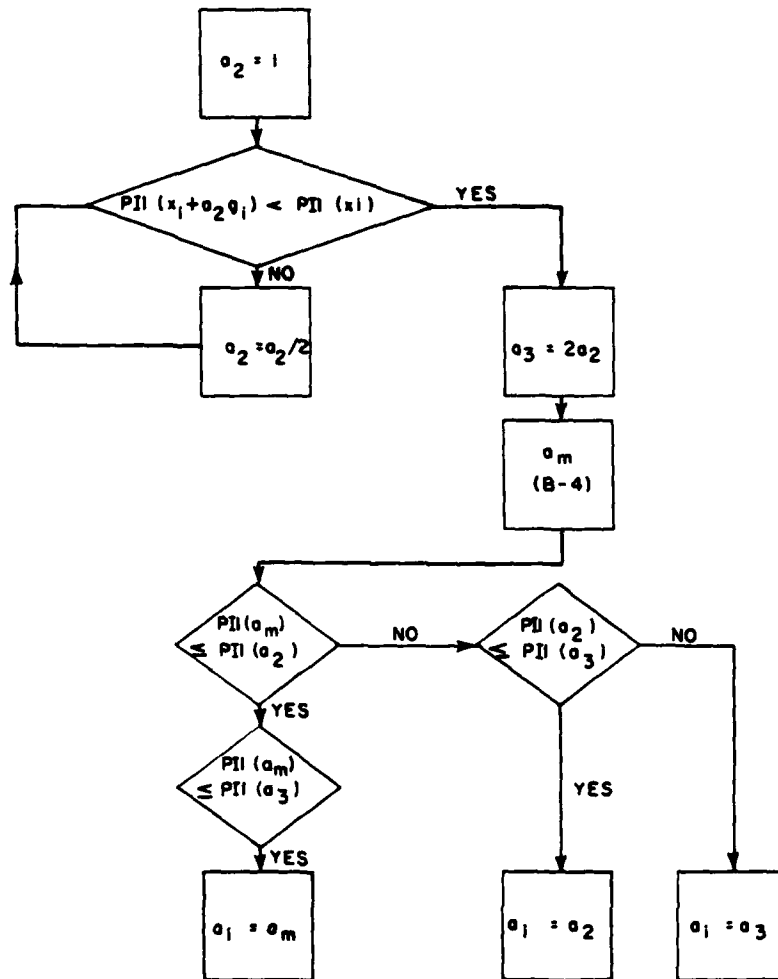


Figure B-1. A Diagram for Selecting a_i , the Step Size for the i th Iteration of the Modified Gauss-Newton Method

APPENDIX C. NUMBER OF OPERATIONS FOR THE MODIFIED GAUSS-NEWTON METHOD

The number of operations (additions, multiplications, divisions, square roots, and trigonometric functions) required for one iteration of the modified Gauss-Newton method is determined in this appendix.

This method is defined by

$$x_{i+1} = x_i + a_i g_i \quad (C-1)$$

where a_i is the step size and g_i is determined from

$$J(x_i) g_i = f(x_i) \quad (C-2)$$

by applying a Householder transformation to the $(J(x_i) \mid f(x_i))$ array. This transformation gives

$$R g_i = \bar{f} \quad (C-3)$$

where R is an upper triangular matrix and g_i is found by back substitution.

The number of operations needed to determine a_i is given, primarily, by the number of additional times that P11 must be evaluated. Appendix B showed that a minimum of two additional evaluations per iteration are required, and the experiments demonstrated that frequently this was all that was needed. Furthermore, equation (B-4) shows that in addition to these operations, 5 additions, 6 multiplications, and 1 division are required. The divisions needed to reduce the step size (see figure B-1), assuming that a reduction is called for, are not counted. Therefore, the minimum number of operations required to find the step size are $2(K-1)+5$ additions, $2K+6$ multiplications, and 1 division.

The equations which define the elements in the kth row of the $(J(x_i) | f(x_i))$ array are given in appendix A. The operations needed to evaluate these elements are shown in table C-1. The operations needed for PII are included.

Table C-1. Number of Operations Required for the $(J|f)$ Array and PII.

Equation	+	x	÷	√	trig
$f(k) = R_{11}(k) - R_s(k)t_1(k)c/b$	1	2	-	-	-
$R_{11}(k) = -R_x(k)\sin\psi(k) - R_y(k)\cos\psi(k)$	1	2	-	-	2*
$R_s(k) = (R_x^2(k) + R_y^2(k) + R_z^2(0))^{1/2}$	2	3	-	1	-
$\partial f/\partial R_{xt}(0) = -\sin\psi(k) - R_x(k)t_1(k)c/R_s(k)b$	1	2	1	-	**
$\partial f/\partial R_{yt}(0) = -\cos\psi(k) - R_y(k)t_1(k)c/R_s(k)b$	1	1	**	-	**
$\partial f/\partial R_{zt}(0) = -R_z(0)t_1(k)c/R_s(k)b$	-	1	**	-	-
$\partial f/\partial V_{xt} = kh \partial f/\partial R_{xt}(0)$	-	2	-	-	-
$\partial f/\partial V_{yt} = kh \partial f/\partial R_{yt}(0)$	-	1	-	-	-
Total for one row	6	14	1	1	-
Total for K rows	6K	14K	K	K	*
PII requires	K-1	K	-	-	-

*Only 2 per leg, so they are not included in the totals.

**c/b is computed off-line, so $c/r_s(k)b$ is 1 division. Also $t_1(k)c/R_s(k)b$ is computed only once.

The number of operations needed to find $R_x(k)$ and $R_y(k)$, given an estimate of the target's velocity and initial position and the observer track, are not counted because they are the same for all iterative estimation algorithms which use a Cartesian signal model.

The Householder transformation used in the iterative least-squares algorithm is, in fact, a product of Householder transformations. The first member of this product operates on the matrix

$$A = \left(J \begin{array}{c} | \\ f \end{array} \right), \quad (C-4)$$

which has K rows and n columns, where $n = 6$. The second member in this product leaves the first row and column of the transformed A matrix unchanged, so it operates on a $(K-1)$ by $(n-1)$ matrix. The next member operates on a $(K-2)$ by $(n-2)$ matrix and so on until the first $(n-1)$ columns of A are transformed. The number of operations needed to transform A so that its new first column has the required structure is shown in the following table. The symbols used in this table are consistent with the Householder transformation algorithm shown on page 63 of reference 3, except that K is used here instead of m .

Table C-2. Number of Operations Required to Transform the First Column of the A Matrix

Symbol	+	x	÷	√
s	$K-1$	$K-1$	-	1
$u(1)$	1	-	-	-
B	-	1	1	-
γ	$(K-1)(n-1)$	$(K+1)(n-1)$	-	-
\bar{A}	$K(n-1)$	$K(n-1)$	-	-
Total	$Kn+(K-1)(n-1)$	$2Kn+n-K+1$	1	1

In the omnidirectional least-squares problem, $n = 6$, hence, the first five columns of A must be transformed to upper triangular form. The next table shows the number of operations needed for each of these transformations.

Table C-3. Number of Operations Required for the Householder Transformation of A

Transformation Defined by Column No.	+	x	÷	√
1	$Kn+(K-1)(n-1)$	$2Kn+n-K+1$	1	1
2	$(K-1)(n-1)+(K-2)(n-2)$	$2(K-1)(n-1)+n-K+1$	1	1
3	$(K-2)(n-2)+(K-3)(n-3)$	$2(K-2)(n-2)+n-K+1$	1	1
4	$(K-3)(n-3)+(K-4)(n-4)$	$2(K-3)(n-3)+n-K+1$	1	1
5	$(K-4)(n-4)+(K-5)(n-5)$	$2(K-4)(n-4)+n-K+1$	1	1
Total	$10Kn-25K-25n+85$	$10Kn-25K-15n+65$	5	5
Total with $n = 6$	$35K-65$	$35K-25$	5	5

To compute PI2 requires (K-6) additions and (K-5) multiplications.

The back substitution algorithm requires 10 additions, 10 multiplications, and 5 divisions.

The total number of operations required for one iteration of the modified Gauss-Newton method is shown in table C-4.

Table C-4. Final Count of Operations for One Iterations of the Modified Gauss-Newton Method

Function	+	x	÷	√
Step size	$2K+3*$	$2K+6*$	$1*$	-
$(J \mid f)$	$6K$	$14K$	K	K
PI1	$K-1$	$35K-25$	-	-
*minimum				

Table C-4. Final Count of Operations for One Iterations of the Modified Gauss-Newton Method (Cont'd)

Function	+	x	÷	√
Householder Transformation	35K-65	35K-25	5	5
PI2	K-6	K-5	-	-
Back substitution	10	10	5	-
Gauss-Newton Total	43K-62	51K-20	K+10	K+5
Minimum Number of Extra Operations for Modified G-N	2K+3	2K+6	1	0

When faced with an ill-conditioned problem, it is advisable to interchange the columns (and maybe the rows as well) of J before applying the Householder transformation. This is explained in reference 6. The number of operations required for this interchange are now shown in table C-4.

APPENDIX D. COMPUTER PROGRAM LISTING

This appendix contains the computer programs used to generate the results shown in section 4. There are two parts to this program: the first generates the averaged measurements; the second estimates the target parameters. Six significant digits are carried in all computations.

```

0 LPRINT 'THIS PROGRAM GENERATES THE AVERAGED MEASUREMENTS'
1 DATA 10,10,10,0,90,0,-10,0,10,-5,10,10000,10000,1000,0,0
5
6 -----
6 TRANSLATION OF THE NAMES OF THE VARIABLES USED IN THIS PROGRAM
7 -----
10 'SO=SPEED OF THE OBSERVER. INPUTTED AS S0,S1,S2.
11 'GAMAC=HEADING OF THE OBSERVER. INPUTTED AS G0,G1,G2
12 'VZ0=Z VELOCITY OF THE OBSERVER. INPUTTED AS Z0,Z10,Z20
13 'VXT=X VELOCITY OF THE TARGET
14 'VYT=Y
15 'RXT=X POSITION OF THE TARGET
16 'RYT=Y
17 'RZT=Z
18 'VIX0=X VELOCITY OF THE OBSERVER
19 'VIZ0=Y
20 'RIX0=X POSITION OF THE OBSERVER
21 'RIY0=Y
22 'RIZ0=Z
23 'RAX=RELATIVE X POSITION(TARGET-OBSERVER)
24 'RAY=Y
25 'RAZ=Z
26 'MEAN=MEAN OF MEASUREMENT NOISE
27 'SDEV=STANDARD DEVIATION OF THE MEASUREMENT NOISE
28 'C11=GAMAC-180 IN RADIANS. ANGLE OF ROTATION
29 'H=SAMPLING PERIOD
30 'RA=R11, PARALLEL TO CENTER LINE OF ARRAY
31 'RP=R11, PERPENDICULAR
32 'ALPHA=COS(1/2) * RLE ANGLE.
33 'TDELAY=TRUE TIME DELAY
34 'SI=USED IN RANDOM NUMBER GENERATOR
35 'NOISE=
36 'TMEAS=NOISY TIME DELAY
37 'AVTM=CONE DIMENSIONAL VARIABLE, 36 ELEMENTS, AVERAGED TMEAS
38 'STALAS=USED TO GENERATOR AVTM
100 *** INPUT OBSERVER INITIAL PARAMETERS AND MANEUVERS. INPUT
    TARGET PARAMETERS. INPUT MEAN AND SDEV OF MEASUREMENT NOISE.***
103
104
105 PRINT 'DATA CARD 1,2,ETC.--OBSERVER SPEED,0,1,2--OBS. HEADING,0,1,2----Z VELL
    CITY 0,1,2-----TARGET Y VELL,0,1,2, X POS.,1 POS.,2 POS.,-----NOISE MEAN,SDEV-
    ----- TYPE CONT. AFTER DATA IS ENTERED'

```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM 2015 RELEASE UNDER E.O. 14176

```

108 '
109 '
110 '*** STOP STATEMENT ALLOWS USER TO ENTER DATA. TYPE CONT
      TO CONTINUE ONCE DATA IS ENTERED ***
111 STOP
112 '-----
117 '
118 '
119 '*** READS INPUT DATA DEFINED IN REF 100.***
      *** INITIAL OBSERVER SPEED AND HEADING (IN DEGREES
      AND RADIANS). ***
125 REAL S0,S1,S2,G0,G1,G2,Z0C,Z1C,Z2C,VXT,VYT,RXT,RYT,RZT,PEAN,SELE
130 S0=S0:(GA*AD=50:HA=50)*.0174533
131 '
132 '
133 '*** SAMPLING PERIOD IS H SECONDS.***
134 H=1
135 '
136 '          RANDOM IS USED BY THE NOISE GENERATOR
137 RANDOM
138 '          *** INITIAL X,Y,AND Z OBSERVER SPEEDS ARE SET. ***
140 VIX0=S0*SEN(HH):VZY0=S0*CGS(HH):VZ0=Z0C
142 '
143 '*** ELIMINATES INITIAL OFFSET IN LINE 520.***
145 RIX0=-VIX0*H:RZY0=-VZY0*H:RZ0=-VZ0*H
149 '-----
149 '          INPUT FOR AVERAGED TMEAS LOOP WHICH STARTS AT 700
150 DIM AVTM(36):ISTMEAS=0:IL=0:IN=0
151 '          THERE ARE 36 ELEMENTS IN AVTM. 0-ELEMENT CORRESPONDING WITH
      TIME=9.5 SECONDS AND 36-ELEMENT WITH 709.5 SEC.
152 '
153 '*** ELIMINATES INITIAL OFFSET IN LINE 340.***
155 RXT=-VXT*H+RXT:RYT=-VYT*H+RYT
158 '
159 LPRINT 'K:': '*RAX*': '*REY*': '*R6Z*': 'RA*': 'RE*':
      '*TDELAY*': 'TMEAS'
160 '
161 '
162 '
163 '-----
195 '*** START OF MAIN LOOP. RUNS THREE LEGS, 4 MINUTES EACH.***
196 '
197 '          STEP ON COUNTER K IS H, THE SAMPLING PERIOD
198 '-----
200 FOR K=0 TO 720 STEP 1
208 '*** CHECKS FOR FIRST OBSERVER MANUEVER AND SET
      NEW SPEED AND HEADING.***
209 '
240 IF 240=K AND K 480 THEN 245 ELSE 200
245 S0=S1:(GAMAD=G1:VZ0=Z1C
255 '
258 '*** CHECKS FOR SECOND OBSERVER MANUEVER AND SETS
      NEW SPEED AND HEADING.***
260 IF K=480 THEN S0=S2:(GAMAD=G2:VZ0=Z2C
270 '
278 '*** FINDS ROTATION ANGLE FOR THE OBSERVER IN ***
      *** FINDS OBSERVER HEADING IN RADIANS ***
280 C11=(GA*AD-180)*.0174533:(GAMAD=ONAL*.0174533

```

```

298 '
299 '*** X,Y,AND Z VEL. AND POSITION OF THE OBSERVER.***
      *****
300 VIXO=S0*SIN(GAMMA);ULYO=S0*COB(GAMMA)
301 R1XO=VIXO*H+R1XC;R2YO=V2YO*H+R2YC;R3ZO=V3ZO*H+R3ZO
302 '
303 '*** X,Y,AND Z TARGET POSITION.***
      *****
304 RXT=VXT*H+RXT;R2YT=V2YT*H+R2YT;R3ZT=R3ZT
305 '*** RELATIVE X,Y,AND Z POSITION (TARGET-OBSERVER).***
      *****
306 R4X=RAT-R1XO;R5Y=R2YT-R2YC;R6Z=R3ZT-R3ZO
307 '
308 '***
      RELATIVE X,Y,Z ARE FORMED.      ***
      TRANSFORMATION TO TOWED      ***
      ARRAY COORDINATES IS NEXT.      ***
309 '
310 '-----
311 '
312 '          TRANSFORMATION
313 RA=R4X*SIN(C11)-R5Y*COB(C11); RP=R4X*COB(C11)-R5Y*SIN(C11)
314 RS=(RA*RA+RP*RP+R6Z*R6Z)*.5
315 '
316 '*** TRUE TIME DELAY ***
317 ALPHA=RA/RS
318 TDELAY=ALPHA/4000.
319 '
320 '-----
321 '
322 '*** GAUSSIAN NOISE GENERATOR. ***
323 '
324 '          12 NUMBERS FROM UNIFORM(0 TO 1)
325 BI=-6
326 FOR I=1 TO 12
327   BI=BI+RND(0)
328 NEXT I
329 '
330 '          NOISE=N(MEAN,SDEV*2)
331 NCISE=SDEV*BI+MEAN
332 '
333 '-----
334 '          NOISY TIME DELAY MEASUREMENT
335 TMEAS=TDELAY+NOISE
336 '
337 '-----
338 '
339 '          AVERAGED TMEAS AT 9.5,29.5,49.5,-----,709.5 SEC.
340 STMEAS=STMEAS+TMEAS
341 L=L+1
342 '          COMPLETES AND STORES AVERAGED TMEAS AND RESETS
          L,STMEAS,AND INCREMENTS N
343 '
344 '
345 IF L=20 THEN L=0:AUTM(N)=STMEAS/20. :N=N+1 :STMEAS=0
346 NEXT K
347 '

```

301
300 PRINT(AVERAGED THEM) IS LISTED BELOW. THERE ARE 30 ELEMENTS 0-TO-29
INCREASING FROM LEFT TO RIGHT.

80
810 FOR NEXT TO 35
820 LEFT(AVTHMND)
870 NEXT N
900

1000
1900
1910
1920
1930 *** WRITE AVERAGED THEM TO DISK ***
2000 OPEN(0,1, 'TYRANOND')
2100 FOR I=0 TO 29
2200 PRINT(AVTHMND)
2300 NEXT I
2400 CLOSE
2500
3000 FOR I=1 TO 9
3100 LEFT(CORRIGISE THEM)
3200 END

THIS PAGE IS BEST QUALITY PRINT FROM
FROM COPY FURNISHED TO DDC

J LPRINT ESTIMATE TARGET PARAMETERS--MODIFIED GAUSS-NEWTON METHOD
 1 CATAIC,10,10,0,90,0,0,0,0,C

11
 20 GIVEN THE ESTIMATE OF THE TARGET PARAMETERS, THIS PROGRAM
 GENERATES THE JACOBIAN AND THE FORCING FUNCTION.

31
 32

35 TRANSLATION OF THE NAMES OF THE VARIABLES

36
 37 RS=SLANT RANGE
 38 M1=DUMMY VARIABLE USED IN LINE 1670
 39 AVTM=AVERAGED TIME DELAY MEASUREMENTS
 41 LL=ITERATIVE COUNTER
 42 M1,M2,& M3 STORE INITIAL X,Y,&Z TARGET POSITION
 43 A=JACOBIAN WITH -F(X) IN THE SIXTH COLUMN
 44 C11=TRANSFORMATION ANGLE
 45 P11=PERFORMANCE INDEX BEFORE HOUSEHOLDER
 46 P21= AFTER
 47 U=USED IN HOUSEHOLDER
 48 SIGN=
 49 MAG=
 50 DATA=
 51 ETA=
 52 X= CORRECTION TO TARGET PARAMETERS
 53 B= USED IN BACK SUBSTITUTION
 54 F1 AND F2 ARE FIRST AND LAST MEASUREMENT WEIGHTS, RESPECTIVELY
 55 TT=TEMPORARY STORAGE
 56 T(35,5)= TEMPORARY STORAGE FOR INTERCHANGE OF ROWS OF A
 57 KI=USED IN QUADRATIC FIT
 58 ZI=USED IN QUADRATIC FIT
 59 IP=(KI)=
 60 SA=
 61 A2=
 62 Q1=TERMINATION PARAMETER
 63 Q2=
 64 Q5= OUTPUT P1 TO DISK PARAMETER
 99

100 THE FIRST FIVE COLUMNS OF A(35,5) CONTAIN THE JACOBIAN
 WHILE THE SIXTH (& LAST) COLUMN CONTAINS -F(X).

1010 THE AVTM, AVERAGED TOELAY MEASUREMENTS, DATA IS CALLED
 FROM DISK.

1030 DIM AVTM(35),A(35,5),U(35),X(4),B(4),T(35,5)
 1035 INFL*FILESPEC*FILESPEC*
 1040 OPEN 'I',1,FILESPEC*
 1050 FOR L=0 TO 35
 1060 INFL*1-AVTM(L)
 1070 PRINT 'L=';L;'AVTM=';AVTM(L);
 1080 NEXT L
 1090 CLOSE
 1100 LPRINT*
 1110 LPRINT*FILESPEC = *FILESPEC*
 1107 LPRINT*
 1113

THIS PAGE IS BEST QUALITY PRINTING
 FROM DATA FURNISHED TO DDC

```

1110 * *** OPTION TO WEIGHT THE FIRST & LAST MEASUREMENTS ***
1120 INPUT 'MEASUREMENT WEIGHTING' W1,W2
1130 IF W1=0 THEN GOTO 1160
1140 INPUT 'SET FIRST AND LAST WEIGHTS' W1,W2
1150 PRINT 'FIRST AND LAST WEIGHTS ARE:' W1 'AND' W2
1160 *
1170 PRINT
1180 PRINT '*****'
1190 *
1200 *
1210 *-----*
1220 'READS INPUT DATA DEFINED IN 1000
1230 READ S1,S2,S3,G1,G2,Z0,Z10,Z20
1240 * *** SET INITIAL TARGET PARAMETERS ***
1250 *-----*
1260 LL=-1
1270 II=-1
1280 RII=1E+12
1290 VXT=0:VYT=0
1300 RS=20000
1310 RZT=3000
1320 RA=RS>400(KAVT/10)
1330 RT=TT*(RS*RS-RA*RA-RZT*RZT)/0.5
1340 C11=(G0-19.1)*.0174533
1350 RXT=RT*COB(C11)+RA*SB(C11)
1360 RYT=-RT*SB(C11)+RA*CB(C11)
1365 GS=0
1370 *
1380 * *** LL=ITERATIVE COUNT ***
1390 *-----*
1400 LL=LL+1
1401 *
1410 KI=0:ZI=0
1420 KI=KI+1
1425 IF KI=9 THEN PRINT 'WRONG DIRECTION': END
1428 *
1430 PRINT '*****'
1440 PRINT 'OBSERVER AND TARGET PARAMETERS', LL='LL', KI='KI', IFF(KI-1)='IFF'
1450 'ZI='ZI', SA='SA'
1460 PRINT '-----'
1465 'MENTS', S1,S2,S3,G1,G2,Z0,Z10,Z20,VXT,VYT,RXT,RYT,RZT
1470 ' SETS OBSERVER INITIAL SPEED AND HEADING IN RADIAN
1480 PRINT
1490 PRINT '-----'
1500 * *** STORES TARGET RANGE COMPONENTS ***
1510 W1=VXT:W2=RYT:W3=RT
1520 EQ=SQ(1-GAMAD)
1530 W1X0=EQ*SB(GAMAD) : W2Y0=EQ*CB(GAMAD) : W3Z0=Z0
1540 ' INITIAL POSITION OF THE OBSERVER, I.E., POSITION AT 9.5 SECONDS
1550 *
1560 R1Y0=V1A0*9.5 : R2Y0=V2Y0*9.5 : R3Z0=V3Z0*9.5
1570 ' INITIAL POSITION OF THE TARGET AT 9.5 SECONDS
1580 * *** STORE INITIAL TARGET POSITION ***
1590 RXT=VXT*9.5+RXT:RYT=VYT*9.5+RYT
1600 *-----*
1610 * RELATIVE POSITION AT 9.5 SECONDS
1620 R4X=RYT-R1Y:R5Y=RYT-R2Y:R6Z=RZT-R3Z
1630 R6=SQ(R4X**2+R5Y**2+R6Z**2)*0.5

```

```

1643
1650 -----
1660 *** -F(0) AND U(0,1,2,3,4) AT TIME=9.5 SECONDS STORED
    IN A(0,I). F AND U DON'T APPEAR IN THE PROGRAM. ***
1670 C11=GAMA0-160*.0174533;PA=R4X*SIN(C11)+R5Y*COS(C11)
1680
1690 A(0,5)=- (R4-R5*4000)*AVTM(0)
1700 A(0,4)=-4000*AVTM(0)*R6Z/RS;A(0,3)=COS(C11)-4000*AVTM(0)*R5Y/RS;A(0,2)=SIN(
C11)-4000*AVTM(0)*R4X/RS
1710 A(0,1)=9.5*A(0,3);A(0,0)=9.5*A(0,2)
1720 -----
1730
1740 P11=A(0,5)*A(0,5)
1750
1760      MAIN LOOP K=1(TIME=29.5) TO K=35 (TIME=709.5 SEC.)
1770      *****
1780
1790
1800 FOR K=1 TO 35
1810     CHECKS FOR DESERVER MANEUVERS
1820 IF K=12 THEN 1850 ELSE 1950
1830     COMPUTE POSITION AT 229 SECONDS. THEN UPDATE SO & GAMA0
1840
1850 R1X0=V1X0*9.5+R1X0 ;R2Y0=V2Y0*9.5+R2Y0 ;R3Z0=V3Z0*9.5+R3Z0
1860
1870 SO=S1 ;GAMA0=G1*.0174533 ;VZ0=Z10
1880 V1X0=SO*SIN(GAMA0) ;V2Y0=SO*COS(GAMA0)
1890
1900     ***COMPUTE DESERVER POSITION AT K=12(TIME=249.5 SEC.) ***
1910 R1X0=V1X0*10.5+R1X0;R2Y0=V2Y0*10.5+R2Y0;R3Z0=V3Z0*10.5+R3Z0;GOTO 2110
1920 -----
1930
1940     CHECKS FOR SECOND DESERVER MANEUVER
1950 IF K=24 THEN 1970 ELSE 2070
1960     COMPUTE OBS. POS. AT 479 SEC. THEN UPDATE SO & GAMA0
1970 R1X0=V1X0*9.5+R1X0 ;R2Y0=V2Y0*9.5+R2Y0 ;R3Z0=V3Z0*9.5+R3Z0
1980
1990 SO=S2 ;GAMA0=G2*.0174533 ;VZ0=Z20
2000 V1X0=SO*SIN(GAMA0) ;V2Y0=SO*COS(GAMA0)
2010     COMPUTE OBS. POS. AT K=24(TIME=489.5 SEC.)
2020
2030 R1X0=V1X0*10.5+R1X0;R2Y0=V2Y0*10.5+R2Y0;R3Z0=V3Z0*10.5+R3Z0
2040 GOTO 2110
2050     IF A MANEUVER OCCURS THE NEXT POSITION IS COMPUTED
    ABOVE , SO THE LOGIC SHOWN BELOW IS SKIPPED.
-----
2060     X,Y, AND Z POSITION OF THE OBSERVER
2070
2080 R1X0=V1X0*20+R1X0      ;R2Y0=V2Y0*20+R2Y0      ;R3Z0=V3Z0*20+R3Z0
2090 -----
2100
2110 ESTIMATE OF X,Y, AND Z POSITION OF THE TARGET
2120 RXT=VXT*20+RXT ;RYT=VYT*20+RYT;RZT=RZT
2130     *** ESTIMATE OF RELATIVE POSITION ***
2140
2150
2160 R4X=RXT-R1X0;R5Y=RYT-R2Y0;R6Z=RZT-R30
2170 R6A=AR4X*AR4X+R5Y*R5Y+R6Z*R6Z;.5
2180 -----
2190     *** ELEMENTS OF THE F-ARRAY STORED AS -F IN A(K,5) ***

```

ALL INFORMATION CONTAINED
HEREIN IS UNCLASSIFIED

```

2200
2210 C11=3AFAD-18)*.0174527
2220 RA=R-Y*(SIN(C11)+R3)*(COS(C11)
2230 M1=4000*W/TM(0)
2240 A(K,3)=-((RA-R3*M1)
2250
2260 *** ELEMENTS OF THE U-ARRAY ***
2270 -----
2280 A(K,4)=-M1*R62/R3
2290 A(K,3)=208(C11)-M1*REY/R3
2300 A(K,2)=SIN(C11)-M1*R4/R3
2310 A(K,1)=(K*11+9.5)*A(K,3)
2320 A(K,0)=(K*11+9.5)*A(K,2)
2330
2340 P11=P11+A(K,0)*A(K,3)
2350 -----
2360 NEXT K
2370 -----
2380 *** OUTPUT FI TO DISK ***
2390 IF Q2=1 THEN 3000
2400 -----
2410 IF LL=0 THEN 2575
2420 *** TERMINATE BY INSERTING TRUE TARGET PARAMETERS ***
2430 Q1=(P11-P21)/P21 ; Q2=ABS(Q1)
2440 IF Q2 =0.1 THEN 4374
2450 -----
2460
2470 *** START OF QUADRATIC FIT ***
2480 IPP(K1)=P11
2490 IF LL=0 THEN 2575
2500 IF ZI=2 THEN 2575
2510 IF ZI=1 THEN 2530
2520 IF IPP(K1)≠PI(LL-1) THEN 2490
2530 SA=-1/(2*ZI)
2540 GOSUB 4450
2550 GOTO 1420
2560 IF KI=1 THEN 2490
2570 SA=1/(2*ZI)
2580 GOSUB 4450
2590 GOTO 1420
2600 A2=(3*PI(LL-1)-4*IPP(KI)+IPP(KI-1))/(2*PI(LL-1)-4*IPP(KI)+2*IPP(KI-1));ZI=1
2610 SA=(A2-1)/(2*ZI)
2620 GOSUB 4450
2630 GOTO 1420
2640 IF IPP(KI)≠IPP(KI-1) THEN 2575
2650 SA=(-PI(LL-1)+4*IPP(KI-1)-2*IPP(KI))/(2*(PI(LL-1)-2*IPP(KI-1)+IPP(KI)));ZI=
2
2660 GOSUB 4450
2670 GOTO 1420
2680 PI(LL)=IPP(KI)
2690 *** END OF QUADRATIC FIT ***
2700 -----
2710 PRINT 'LL=PI(LL) ; PI(LL)=PI(LL)
2720 GOS=611045-30
2730 IF LL=1 THEN 3000
2740 IF LL=2 THEN 3000
2800 -----
2810 *** WEIGHTING C. FIRST & LAST MEMBERSHIP ***

```



```

3300 -----
3310          SET THE NUMBER OF ROWS/COLS
3320 N=COLS-1  IM=ROWS-1
3330 *** MAIN LOOP: TRANSFORMS A COLUMN AT A TIME.
          STOP AT COL=(N-1)--ASSUMES LAST COL IS
          FORCING FUNCTION. ***
3340
3350 FOR K=0 TO (N-1)
3360     MAG=0
3370     *** COMPUTES MAGNITUDE OF COL K
3380     FOR I=K TO M
3390         IF K=I THEN GOTO 3400
3400         *** THE ABOVE CHECKS FOR SQ. ARRAY ***
3410         MAG=MAG+A(I,K)*A(I,K)
3420     NEXT I
3430     *** SETS NEW DIAGONAL ELEMENT
3440     SIGN=A(K,K)/ABS(A(K,K))
3450     MAG=-SIGN*(MAG*.5)
3460
3470     COMPUTES U VECTOR
3480     U(K)=A(K,K)-MAG
3490     FOR J=K+1 TO M
3500         U(J)=A(J,K)  A(J,K)=0
3510     NEXT J
3520     *** K-TH COL IS UPDATED STARTING
          AT THE K-TH ROW. ROW ELEMENTS
          FROM K+1 ON ARE SET TO 0 (SEE 2180)
3530     A(K,K)=MAG
3540     *** THE REMAINING K-1 COLS ARE NOW TRANSFORMED
3550 -----
3560     BATA=1/(MAG*.L(K))
3570     FOR I=K+1 TO N
3580         ETA=0
3590         FOR J=K TO M
3600             ETA=ETA+U(I)*A(I,J)
3610         NEXT I
3620         ETA=ETA*BATA
3630
3640     *** ELEMENTS OF THE I-TH COL OF A ARE REDEFINED.
3650 -----
3660     FOR I=K TO M
3670         A(I,K)=A(I,K)+ETA*U(I)
3680     NEXT I
3690     *** MOVING TO THE NEXT COLUMN ***
3700     NEXT K
3710
3720 *** MOVING TO THE NEXT TRANSFORMATION ***
3730 NEXT K
3740 *** FOR INDEX THOR SQ. ARR. AFTER TRANSFORMATION ***
3750
3760 FOR I=0 TO
3770 FOR J=0 TO
3780 FOR I=0 TO
3790 FOR J=0 TO
3800 NEXT I
3810 PRINT '.....'

```

BEST QUALITY PRACTICES
 APPLIED TO BDO

```

3830 '
3840 LPRINT 'P2I=';P2I;'LL=';LL
3850 '
3860 LPRINT '*****'
3870 RETURN
3875 '
3876 '
3880 'SUBROUTINE FOR INTERCHANGING ALL ROWS OF A(35,5). RETURNS TO 2353
3890 FOR K=0 TO 35
3900   FOR J=0 TO 5
3910     T(K,J)=A(K,J)
3920   NEXT J
3930 NEXT K
3940 FOR K=0 TO 35
3950   FOR J=0 TO 5
3960     A(35-K,J)=T(K,J)
3970   NEXT J
3980 NEXT K
3990 RETURN
4000 '-----
4070 '
4073 ' *** FROM A GOTO IN LINE 2369 ***
4074 PI(LL)=PII
4075 LPRINT 'LL=';LL,'PI(LL)=';PII,'Q2=';Q2;'P2I=';P2I
4076 Q5=1
4080 INPUT 'QXT=';QXT
4090 INPUT 'QYT=';QYT
4100 INPUT 'RXT=';RXT
4110 INPUT 'RYT=';RYT
4120 INPUT 'RZT=';RZT
4130 GOTO 1400
4140 '-----
4150 ' *** SUBROUTINE TO UPDATE TARGET PARAMETERS ***
4160 '   CALLED ( LINES 2430,2470,2510,&2550 ) BY QUADRATIC FIT
4170 QXT=QXT+SA*X(0) : QYT=QYT+SA*X(1) : RXT=R1+SA*X(2) : RYT=R2+SA*X(3) : RZT=R3+SA
X(4)
4180 RETURN
4190 '-----
5000 '   OUTPUT TO DISK
5001 ' *** FROM A GOTO IN LINE 2364 ***
5005 PI(LL)=PII
5006 LPRINT 'LL=';LL,'PI(LL)=';PI(LL)
5010 OPEN 'D',2,'PI(LL)/D11:1'
5020 FOR L=0 TO 9
5030 PRINT #2,PI(LL)
5040 NEXT LL
5050 CLOSE
5060 END

```

INITIAL DISTRIBUTION LIST

Addressee	No. of Copies
NAVSEA (SEA-63, -63R, -63R-1, -63R-13, -63X, -631X -63D (2 cy), PMS-409 (2 cy))	10
DTIC, Alexandria	12

END

DATE
FILMED

12-81

DTIC