20000831138

# NAVAL POSTGRADUATE SCHOOL
Monterey, California

# THESIS

## ACCURACY ASSESSMENT FOR THE AUXILIARY TRACKING SYSTEM

by

Michael P. [illegible]aylor

September, 1991

Thesis Advisor: Robert R. Read

Approved for public release; distribution is unlimited.

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE

## REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

| | |
|---|---|
| 1a. REPORT SECURITY CLASSIFICATION<br>UNCLASSIFIED | 1b. RESTRICTIVE MARKINGS |
| 2a. SECURITY CLASSIFICATION AUTHORITY | 3. DISTRIBUTION/AVAILABILITY OF REPORT<br>Approved for public release; distribution is unlimited. |
| 2b. DECLASSIFICATION/DOWNGRADING SCHEDULE | |
| 4. PERFORMING ORGANIZATION REPORT NUMBER(S) | 5. MONITORING ORGANIZATION REPORT NUMBER(S) |
| 6a. NAME OF PERFORMING ORGANIZATION<br>Naval Postgraduate School<br>6b. OFFICE SYMBOL (If applicable)<br>3A | 7a. NAME OF MONITORING ORGANIZATION<br>Naval Postgraduate School |
| 6c. ADDRESS (City, State, and ZIP Code)<br>Monterey, CA 93943-5000 | 7b. ADDRESS (City, State, and ZIP Code)<br>Monterey, CA 93943-5000 |
| 8a. NAME OF FUNDING/SPONSORING ORGANIZATION<br>8b. OFFICE SYMBOL (If applicable) | 9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER |
| 8c. ADDRESS (City, State, and ZIP Code) | 10. SOURCE OF FUNDING NUMBERS<br>PROGRAM ELEMENT NO.<br>PROJECT NO.<br>TASK NO.<br>WORK UNIT ACCESSION NO. |

11. TITLE (Include Security Classification)
ACCURACY ASSESSMENT FOR THE AUXILIARY TRACKING SYSTEM

12. PERSONAL AUTHOR(S)
Taylor, Michael P.

| 13a. TYPE OF REPORT | 13b. TIME COVERED | 14. DATE OF REPORT (Year, Month, Day) | 15. PAGE COUNT |
|---|---|---|---|
| Master's Thesis | FROM ______ TO ______ | September 1991 | 46 |

16. SUPPLEMENTARY NOTATION
The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government

17. COSATI CODES

| FIELD | GROUP | SUB-GROUP |
|---|---|---|
| | | |
| | | |

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
Auxiliary Tracking System, Portable Tracking System, Torpedo Tracking, Range Tracking Accuracy.

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

The Auxiliary Tracking System (ATS) is being developed to allow for an accurate underwater tracking capability that can be deployed anywhere in the world. A simulation was created to determine the impact upon the relative accuracy of the system due to a normally distributed noise. The simulation was used to look at the impact of range between targets and the depth spacing of the suspended transducers on the relative tracking accuracy.

| | |
|---|---|
| 20. DISTRIBUTION/AVAILABILITY OF ABSTRACT<br>☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS | 21. ABSTRACT SECURITY CLASSIFICATION<br>Unclassified |
| 22a. NAME OF RESPONSIBLE INDIVIDUAL<br>Robert R. Read | 22b. TELEPHONE (Include Area Code)<br>(408) 646-2382<br>22c. OFFICE SYMBOL<br>OR/Re |

DD Form 1473, JUN 86

Previous editions are obsolete.

S/N 0102-LF-014-6603

SECURITY CLASSIFICATION OF THIS PAGE
Unclassified

# Accuracy Assessment for the
# Auxillary Tracking System

by

Michael P. Taylor
Lieutenant, United States Navy
B.S., United States Air Force Academy

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN APPLIED SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
September 1991

**Author:**
Michael P. Taylor

**Approved by:** R R Read
Robert R. Read, Thesis Advisor

Alan B. Coppens, Second Reader

James N. Eagle, Chairman
Antisubmarine Warfare Academic Group

# ABSTRACT

The Auxiliary Tracking System (ATS) is being developed to allow for an accurate underwater tracking capability that can be deployed anywhere in the world. A simulation was created to determine the impact upon the relative accuracy of the system due to a normally distributed noise. The simulation was used to look at the impact of range between targets and the depth spacing of the suspended transducers on the relative tracking accuracy.

## TABLE OF CONTENTS

# I. INTRODUCTION

Within the last several years, the Navy has seen a tightening of the purse strings from congress, and as a result, must find more cost effective ways to perform the various missions that are still required. One area that has long been a concern is the limited number of underwater tracking ranges. Alternatives to the development of another fixed range must be considered because of the financial impact.

On the west coast, there are currently three areas where underwater tracking ranges are available. Washington, San Clemente, California, and Hawaii. With the limited number of sites and the other demands on range time for research and development, fleet ASW training opportunities are limited. Added to this is the fact that for many exercises, the units must transit to Hawaii simply because that is the only range available. The portable Auxiliary Tracking System (ATS) was conceived for these reasons.

The ATS has been designed as a truly portable underwater tracking range. This thesis will look at the theoretical tracking accuracy of the system currently under engineering development and testing at the Naval Undersea Warfare Engineering Station (NUWES).

# II. ATS CONCEPT

## A. GENERAL DESCRIPTION

With the recent improvements made in electronics and the Global Positioning Satellite System (GPSS), the dream of a portable tracking range now seems possible with current technology. The engineering development model consists of four underwater tracking buoys and a portable computer site. The buoys are deployed either from a ship or helicopter and are recoverable. For the four buoy system, deployment will be similar to that shown in Figure 1. The final system will consist of nine buoys to be deployed in various patterns based upon the needs of the system being tested.

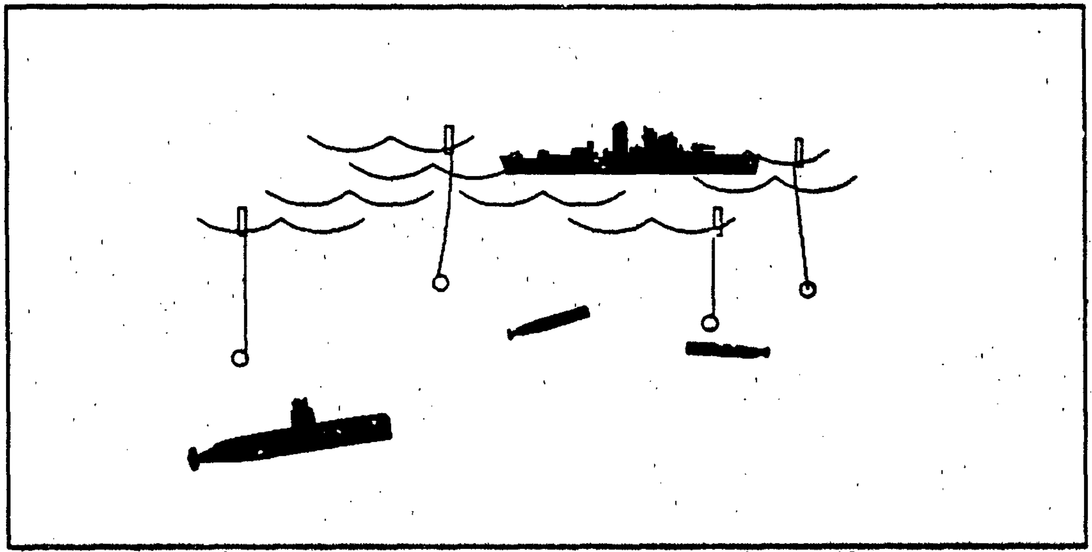

Figure 1. ATS Deployment Configuration

### 1. Buoy System

The buoys used in the ATS will be free floating, self contained buoys. They will have a transducer suspended below at a predetermined depth (between 50 and 300 meters) used both for tracking the targets and locating the other units in the tracking range. The acoustic signal processor contained in each buoy will provide for the detection of the acoustic pings emitted by the other sources. The GPSS receiver in the buoy will provide the Latitude/Longitude and a very accurate time standard for the system. The on-board computer will control the GPSS receiver and acoustic signal processor, monitor the status of the buoy, perform a system up/down self test, time stamp and format the position data collected by the GPSS receiver and acoustic signal processor and send the data to the computer site via an RF data link.

### 2. Computer Site

The control station for the ATS range will be the computer site, which will be positioned on a ship located off the range, or on land if the range is set up in coastal waters. The computer site will be contained in a portable van allowing transport anywhere in the world either by ship or by cargo aircraft. The only support needed will be AC power, as the computer site will be self contained. The computer site will contain the RF data link receivers, RF voice communications equipment, a GPSS receiver, the range tracking computers and peripherals, and other equipment as needed.

## B. SYSTEM OPERATION

The concept embodied by the ATS is a "sing-around" self calibrating tracking range. This means that the buoys will locate each other. Each buoy has an acoustic pinger on the floating buoy and a transducer suspended below. In a two minute cycle, each buoy pinger will transmit an acoustic signal followed by each suspended transducer also transmitting an acoustic signal. The acoustic processor within each buoy will control the pinging cycle; a coded pulse will be used to identify the signal. The transit time of the signal will be known as each buoy will transmit a pulse at a specific time, and the reception time of the signal will be sent to the computer site via the RF link from the buoy that received the signal. From this, the transit time of the signal is known, and the range between the source and receiver is calculated by using the Sound Velocity Profile of the water mass. The signal from the buoy pinger to the transducer suspended below is used to calculate the depth of each transducer. The transit times from transducer to transducer are then used to calculate the slant ranges when the depth of each transducer has been determined. From this data, a range grid is set up as shown in Figure 2. The location of each transducer and the range grid will be updated every two minutes by this method.

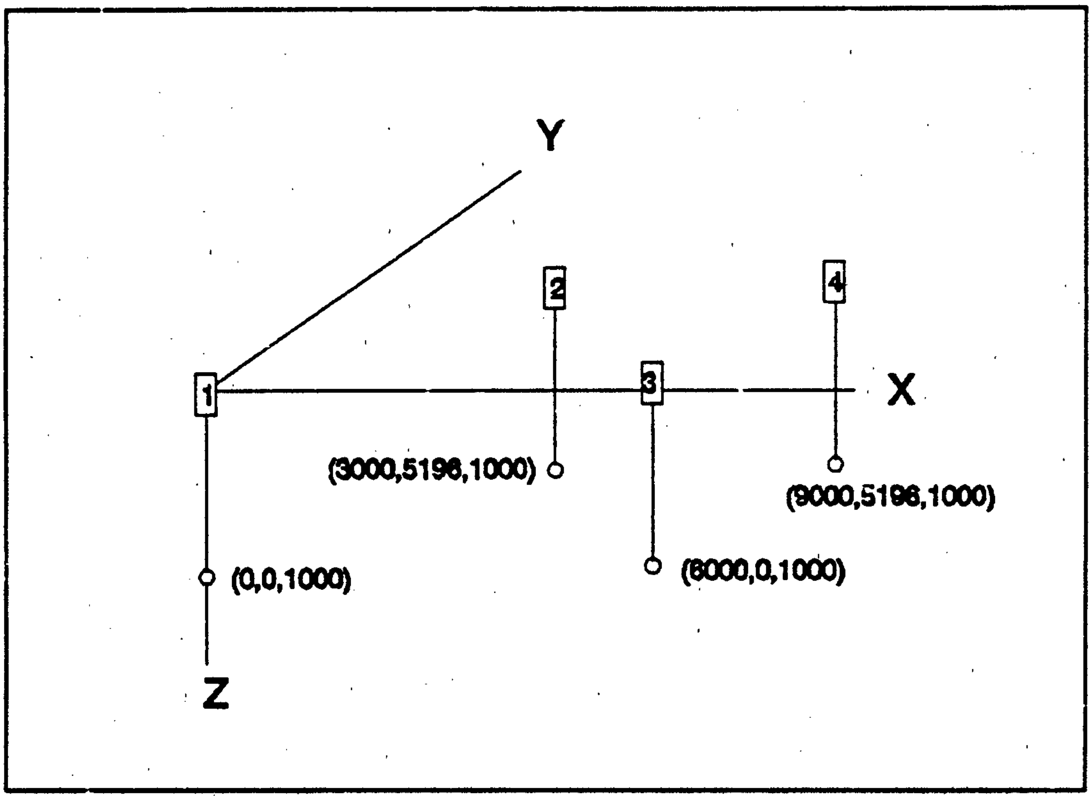

Figure 2. ATS Range Grid

With a relative grid established, the system now determines the positions of the targets on the range. The ATS has three distinct frequency bands to utilize, by incorporating a Shift Frequency, Shift Keying (SFSK) processor and transmitter. The high frequency will be used for the sing-around calibration already described, and the mid and low frequencies will be used to track the torpedo and the target. The SFSK processor and transmitter is similar to the system used on the Quinalt Shallow Water range at NUWES.

# III. SIMULATION DESCRIPTION

## A. PROGRAM DESCRIPTION

A simulation program was written to determine the effects of a normal error on the transit times received at the computer site to investigate the accuracy of the ATS. The effects of shadow zones were not considered in the simulation due to the relatively short distances between any source and receiver pair. The simulation program has three main sections; solving for transit times, solving for positions, and collecting statistics.

### 1. Simulation Inputs

The ATS simulation program requires three sets of inputs. The first set of inputs is the sound speed profile for the water mass. The program can accept up to 60 data points for the profile. The second set of inputs is the actual positions for the four buoys, the four suspended transducers and the two targets. The inputs are cartesian coordinate with buoy one being at the origin and buoy three establishing the X-Z plane. The third sat of inputs is for the number of replications desired and the standard deviation of the noise for the systems. The standard deviation is a function of the distance between the source and receiver, and within the program is modeled by changing the travel times in each of three groups; buoy to transducer, transducer to transducer, and target to transducer.

### 2. Solving for Times

Once the inputs have been entered, the program solves for the actual transit time between each source and receiver pair. This is done by taking the slant range between each source and receiver (Equation 3.1) and dividing by the

$$Range=\sqrt{(X_s-X_r)^2+(Y_s-Y_r)^2+(Z_s-Z_r)^2} \qquad (3.1)$$

effective sound speed between source and receiver. The effective sound speed is calculated by solving for the travel time for a vertically propagating wave in each layer. The travel times are summed between the source and receiver depths and the difference in depths is divided by the travel time to yield the effective sound speed between the source and receiver. The derivation of the equation is shown below and starts with the definition of the travel time (Equation 3.2).

$$t=\int dt=\int_{s_0}^{s}\frac{ds}{c(z)} \qquad (3.2)$$

The ray path within a constant gradient is defined as the arc of a circle of radius equal to $\sigma$ (Equation 3.3 and 3.4) [Ref 1:p. 402].

$$ds=\sigma d\theta \qquad (3.3)$$

$$\sigma=\frac{c(z)}{g\ \cos\theta} \qquad (3.4)$$

Substitution of Equation 3.4 into Equation 3.2 yields Equation 3.5 and 3.6. The solution of the integral is shown in Equation 3.7

$$t=\int_{\theta_o}^{\theta}\frac{d\theta}{g\ \cos\theta} \qquad (3.5)$$

$$t=\frac{1}{g}\int_{\theta_o}^{\theta}\sec(\theta)\,d\theta \qquad (3.6)$$

$$t=\frac{1}{g}\left[\ln|\sec(\theta)+\tan(\theta)|\right] \qquad (3.7)$$

Separation of the terms yield Equation 3.8. Using snell's law [Ref 1: p. 401], as shown in Equation 3.9, and making the assumption that the sine terms cancel for a vertically propagating wave, Equation 3.8 can be reduced to Equation 3.10.

$$t=\frac{1}{g}\ln\left|\frac{1+\sin\theta}{\cos\theta}\,\frac{\cos\theta_o}{1+\sin\theta_o}\right| \qquad (3.8)$$

$$\frac{\cos\theta_o}{\cos\theta}=\frac{c(z_o)}{c(z)} \qquad (3.9)$$

$$t=\frac{1}{g}\ln\frac{c(z_2)}{(c(z_1)} \qquad (3.10)$$

By summing the travel times within each layer, the propagation time is found. If within a layer, the gradient is equal to zero, the program takes the vertical distance within the layer and divides by the sound speed within the layer to determine the time t. The effective sound speed is now found (Equation 3.11).

$$effspd=\frac{Z_2-Z_1}{t} \tag{3.11}$$

Knowing the slant range and the effective sound speed between each source and receiver pair, the transit times for all pairs are determined (Equation 3.12).

$$Time=\frac{Range}{effspd} \tag{3.12}$$

### 3. Adding Noise

When all of the transit times have been calculated and stored, the program begins the iteration cycle. The first step is the addition of the noise value to the transit times. This is done by taking a standard normal random variable and multiplying it by the transit time and the percentage value for the standard deviation. This results in a normal random variable with mean equal to zero and standard deviation equal to the given percentage of the transit time. This method was chosen because the transducer to transducer ranges and times are typically 5 times greater than the pinger to suspended transducer times, and the error should be a function of range.

### 4. Solving for Transducer Locations

After the noise has been added to each of the transit times, the location of each transducer is calculated. The depth of each transducer is found first using the transit time from the buoy pinger to the transducer suspended below (i.e. buoy pinger one to transducer one). From this transit time and using the length of the cable as a starting point, the depth is determined through an iterative process by calculating the effective sound velocity between the pinger and the estimated depth of the transducer. The solution is then found by multiplying the transit time by the effective sound velocity. The resulting depth is then compared with the estimated depth, and when the difference between the two values is less than one foot, the program continues, otherwise the resulting depth is used for the estimated depth and the calculation is repeated. Once the four transducer depths are established, the range coordinate system is established using transducer one as the origin of the X-Y grid and transducer three as the intersection with the horizontal axis (X-axis). There are two transit times for each range calculation made (transit time from transducer one to three and from transducer three to one), and the times are averaged. The slant range (SR) is then found by calculating the effective sound velocity between the two depths and multiplying by the average transit time between the two transducers. The horizontal distance (H) is calculated using Equation 3.13. The position of transducer

$$H=\sqrt{(SR)^2-(Z_1-Z_2)^2} \tag{3.13}$$

two is found by using the horizontal ranges from transducers one and three and using the GPSS position of buoy two to determine which of two unique solutions is correct (the intersection of two circles). Transducer four is then located the same way using the positions and ranges from transducers two and three.

### 5. Solving for Target Locations

With the range coordinate system established and the location of all transducers found, we are positioned to calculate the location of the two targets. It is assumed that all four transducers receive the signal from each target. The location of the target is then calculated through a least squares method similar to one used by Larry Anderson at NUWES. The detailed procedures for the algorithm are presented below.

Given the location of each transducer as ($X_i$, $Y_i$, $Z_i$) and the solution range between transducer i and the target ($R_{mi}$) from the transit time, the location of the Target ($X_T$, $Y_T$, $Z_T$) is calculated. $R_{mi}$ is the measured slant range between target and transducer i, and $R_i$ is the calculated slant range between target and transducer i as given by Equation 3.14. The

$$R_i=\sqrt{(X_T-X_i)^2+(Y_T-Y_i)^2+(Z_T-Z_i)^2} \tag{3.14}$$

location of the target is then calculated with respect to minimizing the Sum of Squares of errors (SSE), where the SSE is defined by Equation 3.15. The calculated slant ranges ($R_i$)

$$SSE=\sum_{i=1}^{4}(R_i-R_{mi})^2 \tag{3.15}$$

are linearized around the trial value ($X_{To}$, $Y_{To}$, $Z_{To}$), shown in Equation 3.16, where $|_o$ means evaluated at point "o". The

$$R_i=R_{i_o}+\left.\frac{\partial R_i}{\partial X_T}\right|_o(X_T-X_{T_o})+\left.\frac{\partial R_i}{\partial Y_T}\right|_o(Y_T-Y_{T_o})+\left.\frac{\partial R_i}{\partial Z_T}\right|_o(Z_T-Z_{T_o}) \tag{3.16}$$

partial derivatives are listed in Equations 3.17 through 3.19.

$$\left.\frac{\partial R_i}{\partial X_T}\right|_o=\frac{X_{T_o}-X_i}{R_{i_o}} \tag{3.17}$$

$$\left.\frac{\partial R_i}{\partial Y_T}\right|_o=\frac{Y_{T_o}-Y_i}{R_{i_o}} \tag{3.18}$$

$$\left.\frac{\partial R_i}{\partial Z_T}\right|_o=\frac{Z_{T_o}-Z_i}{R_{i_o}} \tag{3.19}$$

The linearized model is substituted into Equation 3.15 and the SSE is minimized with respect to ($X_T$, $Y_T$, $Z_T$). The resulting normal equations are put in matrix form as shown in Equation

3.20. The individual matrices are shown in Equations 3.21 through 3.23. Equation 3.20 is solved for the correction matrix (Equation 3.22) which when added to ($X_{T_o}$, $Y_{T_o}$, $Z_{T_o}$)

$$AX=B \tag{3.20}$$

$$A = \begin{vmatrix} \sum \left.\frac{\partial R_i}{\partial X_T}\right|_o^2 & \sum \left.\frac{\partial R_i}{\partial Y_T}\right|_o \left.\frac{\partial R_i}{\partial X_T}\right|_o & \sum \left.\frac{\partial R_i}{\partial Z_T}\right|_o \left.\frac{\partial R_i}{\partial X_T}\right|_o \\ \sum \left.\frac{\partial R_i}{\partial X_T}\right|_o \left.\frac{\partial R_i}{\partial Y_T}\right|_o & \sum \left.\frac{\partial R_i}{\partial Y_T}\right|_o^2 & \sum \left.\frac{\partial R_i}{\partial Z_T}\right|_o \left.\frac{\partial R_i}{\partial Y_T}\right|_o \\ \sum \left.\frac{\partial R_i}{\partial X_T}\right|_o \left.\frac{\partial R_i}{\partial Z_T}\right|_o & \sum \left.\frac{\partial R_i}{\partial Y_T}\right|_o \left.\frac{\partial R_i}{\partial Z_T}\right|_o & \sum \left.\frac{\partial R_i}{\partial Z_T}\right|_o^2 \end{vmatrix} \tag{3.21}$$

$$X = \begin{vmatrix} X_T-X_{T_o} \\ Y_T-Y_{T_o} \\ Z_T-Z_{T_o} \end{vmatrix} \tag{3.22}$$

$$B = \begin{vmatrix} \sum (R_{mi}-R_{io}) \left.\frac{\partial R_i}{\partial X_T}\right|_o \\ \sum (R_{mi}-R_{io}) \left.\frac{\partial R_i}{\partial Y_T}\right|_o \\ \sum (R_{mi}-R_{io}) \left.\frac{\partial R_i}{\partial Z_T}\right|_o \end{vmatrix} \tag{3.23}$$

yields the improved target position ($X_T$, $Y_T$, $Z_T$). If the sum of the squares of the corrections is less than some value epsilon, Equation 3.24, the iteration is stopped. Otherwise,

$$(X_T-X_{T_o})^2+(Y_T-Y_{T_o})^2+(Z_T-Z_{T_o})^2<e \tag{3.24}$$

the target position ($X_T$, $Y_T$, $Z_T$) is used for the next iteration in place of ($X_{To}$, $Y_{To}$, $Z_{To}$) and the procedure is repeated until 50 iterations have been completed.

### 6. Collecting Statistics

When the positions of the targets have been solved for, the errors are calculated. The distance between the actual and solution location of each target (Equation 3.25), and the distance in the relative position of the targets with respect to each other (Equation 3.26) are calculated.

$$TE_i=\sqrt{(X_{oi}-X_{si})^2+(Y_{oi}-Y_{si})^2+(Z_{oi}-Z_{si})^2} \qquad (3.25)$$

$$RE=\sqrt{((X_{o1}-X_{o2})-(X_{s1}-X_{s2}))^2+\ldots} \qquad (3.26)$$

$X_{oi}$ is the X-value for the true position of the ith target, $X_{si}$ is the X-value for the solved position of the ith target , $TE_i$ is the True error for the ith target, and RE is the relative error between the two targets. The values for $TE_i$ and RE are summed over the iteration loop and the programs final output is the mean and standard deviation of the true position error for each target and the relative position error between the two targets.

## B. SIMULATIONS CONDUCTED

Three factors are considered as to the influence of the relative accuracy: Magnitude of the noise in the measurements, the proximity of target one to target two, and whether the transducers are at equal or staggered depths.

### 1. Noise Values

Noise values are entered into the simulation by the standard deviation being a percentage of the true value. Values examined range between 0.1 and 1.0 percent.

### 2. Target Locations

The simulations were conducted for targets at close range (100 ft separation) and long range (3000 ft separation) to investigate the effects of target separation on the relative accuracy.

### 3. Transducer Depths

Some discussion has taken place as to the effects of staggering the transducer depths or having them all at an equal depth with the thought being would accuracy be gained by having the suspended transducers staggered at varying depths. Would this allow for better target resolution given the four slant ranges from the target to the transducers? Simulations were also conducted to draw a conclusion on the selection of transducers depths; equal depths or staggered depths with intervals of 100 feet.

## IV. DATA ANALYSIS

Each of the four following cases were run using the simulation program with the noise values ranging from 0 to 1 percent. The performance goal for the ATS is 20 yards relative accuracy. The specifics of the four cases are as follows:

1. Case One: Transducers at equal depth (1000 feet) and targets at close range (100 feet).
2. Case Two: Transducers at equal depth and targets at long range (3000 feet).
3. Case Three: Transducers at staggered depths (100 foot intervals) and targets at close range.
4. Case Four: Transducers at staggered depths and targets at long range.

The results for the four cases are shown in Figure 3. The limit for the noise varies from 0.3 percent for cases two and four (targets at long range) to 0.4 percent for cases one and three (targets at close range).

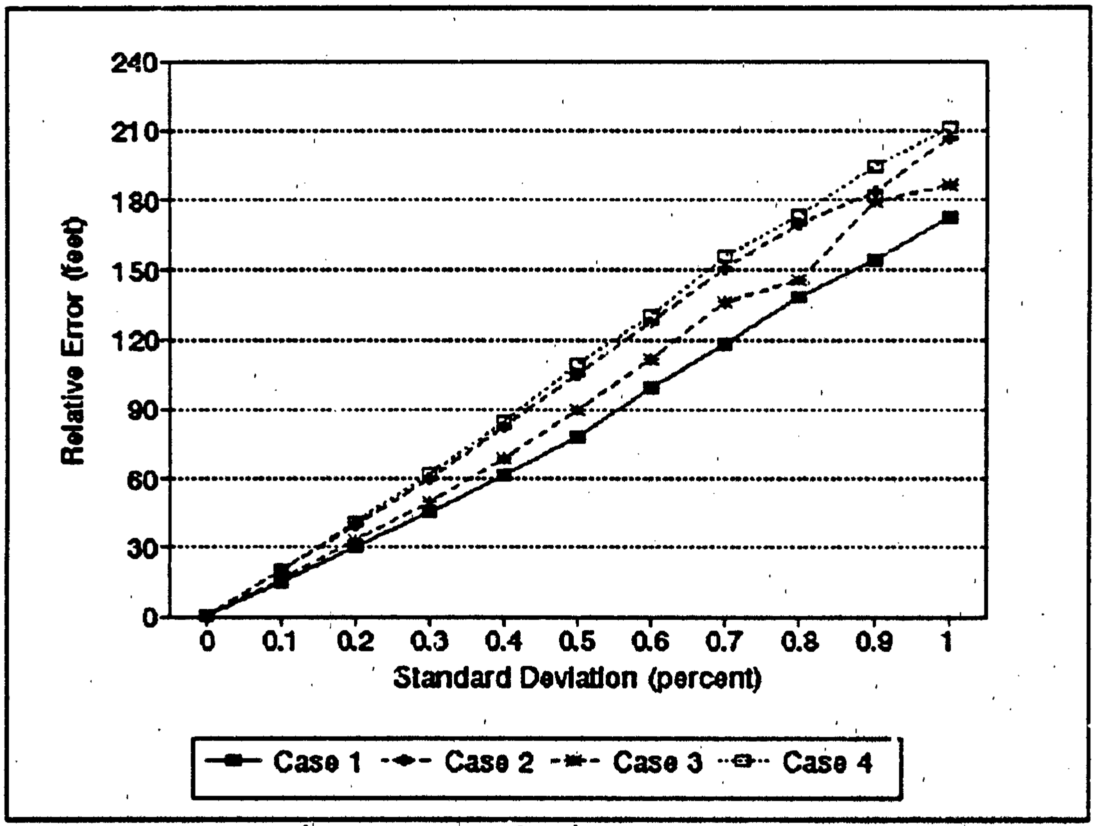

Figure 3. Relative Error Plot

# V. CONCLUSIONS AND RECOMMENDATIONS

## A. CONCLUSIONS

The results from the simulation show that the system can not handle a lot of noise as currently designed. The staggering of the transducer depths did not appear to have an effect on the accuracy of the system. All four cases were relatively close for noise value where the relative accuracy exceeded 20 yards.

## B. RECOMMENDATIONS

The accuracy assessment for the ATS is a very complex problem. This study has only focused on a few areas and a complete assessment of the system will require further study. From the initial results on the engineering development model, the outlook is very promising for the Navy to receive substantial benefits from its continued funding. Analysis of the data from the prototype buoys should reveal the magnitude of the noise, and the expected accuracy can then be determined. An additional area that should also be considered is to use telemetry data from the targets in the tracking problem. This could be utilized to provide the depth of the target and the velocity. By incorporating a Kalman Filter into the algorithm, it may be possible to reduce the impact from a bad ping cycle. This would also necessitate looking at the accuracy of the depth sensors currently available.

The Auxiliary Tracking System should continue to be developed. Potential benefits for the Navy warrant the investment now, as current ranges are taxed to keep up with the demands placed upon them by the many users. As new weapons systems are developed, the ATS will allow more realistic operational testing to be conducted by allowing the weapons systems to be evaluated in various ocean types.

# APPENDIX A

```
CC  ATSSIM.FOR
CC         SIMULATION PROGRAM TO EVALUATE AUXILLARY TRACKING SYSTEM
CC         ABSOLUTE AND RELATIVE ACCURACY ASSESSMENTS BASED UPON A
CC         RANDOM NOISE WITH A GIVEN MEAN AND VARIANCE.
CC
        REAL*8 SEED
        REAL*4 XBUOY, YBUOY,ZBUOY,XPHONE,YPHONE,ZPHONE,XTGT,YTGT,ZTGT,
       + MEAN1, VAR1, MEAN2, VAR2, MEANR, VARR, SRANGE, RANGE, NOISE,
       + TTIME1, HTIME1, BTIME1, TTIME, HTIME, BTIME, A, B
        INTEGER*4 BUOYS, MAXSSV, NCOUNT
        DIMENSION XBUOY(4),YBUOY(4),ZBUOY(4),XPHONE(4),YPHONE(4),
       +          ZPHONE(4), SRANGE(6),BTIME1(4,4),HTIME1(4,4),
       +          PHONEX(4), PHONEY(4),PHONEZ(4), DEPTH(40),SSPVEL(40),
       +          XTGT(2),YTGT(2),ZTGT(2),TGTX(2),TGTY(2),TTIME1(2,4),
       +          TGTZ(2),BTIME(4,4),HTIME(4,4),TTIME(2,4),
       +          TERROR(2),TESQ(2),TESUM(2),TRANGE(2,4)
        MAXSSV = 40
        BUOYS = 4
        ZCABLE = 1000.0
        CALL INPUT (NCOUNT, BTN, STN, TTN, SEED)
        CALL PROFILE (MAXSSV,NUMSSV,DEPTH,SSPVEL)
        CALL POSITS (XBUOY,YBUOY,ZBUOY,XPHONE,YPHONE,ZPHONE,
       +          XTGT,YTGT,ZTGT)
CC      CALCULATE TRANSIT TIMES FOR BUOYS
        DO 20 I = 1, 4
          DO 10 J = 1, 4
            RANGE = SQRT((XBUOY(I)-XPHONE(J))**2+(YBUOY(I)-YPHONE(J))**2 +
       +    (ZBUOY(I) - ZPHONE(J))**2)
            CALL HMVELOC(ZBUOY(I),ZPHONE(J),NUMSSV,DEPTH,SSPVEL,HMVEL)
            BTIME(I,J) = RANGE/HMVEL
   10     CONTINUE
   20   CONTINUE
        DO 50 I = 1, 4
          DO 40 J = 1, 4
            IF (I .EQ. J) GOTO 30
         RANGE = SQRT((XPHONE(I)-XPHONE(J))**2+
       +      (YPHONE(I)-YPHONE(J))**2+(ZPHONE(I)-ZPHONE(J))**2)
            CALL HMVELOC(ZPHONE(I), ZPHONE(J), NUMSSV,DEPTH,SSPVEL,HMVEL)
            HTIME(I,J) = RANGE/HMVEL
   30       CONTINUE
   40     CONTINUE
   50   CONTINUE
        DO 70 I = 1, 2
          DO 60 J = 1, 4
         RANGE = SQRT((XTGT(I)-XPHONE(J))**2+(YTGT(I)-YPHONE(J))**2+
       +    (ZTGT(I) - ZPHONE(J))**2)
            CALL HMVELOC(ZTGT(I),ZPHONE(J),NUMSSV,DEPTH,SSPVEL,HMVEL)
            TTIME(I,J) = RANGE/HMVEL
   60     CONTINUE
   70   CONTINUE
```

```
      DO 900 KOUNT=1, NCOUNT
   79 CONTINUE
CC    ADD NOISE
      DO 71 I = 1, 2
      J = I + 2
      CALL URAND(A,B,SEED)
      BTIME1(I,I) = BTIME(I,I) + (A*BTN*BTIME(I,I))
      BTIME1(J,J) = BTIME(J,J) + (B*BTN*BTIME(J,J))
   71 CONTINUE
      DO 73 I = 1, 4
      DO 72 J = 1, 2
        K = J + 2
        CALL URAND (A,B,SEED)
        HTIME1(I,J) = HTIME(I,J) + (A*STN*HTIME(I,J))
        HTIME1(I,K) = HTIME(I,K) + (B*STN*HTIME(I,K))
   72   CONTINUE
   73 CONTINUE
      DO 75 J = 1, 4
       CALL URAND(A,B,SEED)
       TTIME1(1,J) = TTIME(1,J) + (A*TTN*TTIME(1,J))
       TTIME1(2,J) = TTIME(2,J) + (B*BTN*TTIME(2,J))
   75 CONTINUE
CC    CALC BUOY POSITS
      DO 80 I = 1, 4
      PHONEX(I) = XBUOY(I)
      PHONEY(I) = YBUOY(I)
   80 CONTINUE
      CALL HYDEPTH (ZCABLE, ZBUOY, NUMSSV, DEPTH,
     +  SSPVEL, HMVEL, BTIME1, PHONEZ)
      CALL RPHONE (SRANGE, HTIME1, PHONEZ, NUMSSV, DEPTH, SSPVEL, HMVEL)
      CALL TWOD (SRANGE, PHONEX, PHONEY, PHONEZ, YBUOY)
CC    CALC TARGET POSITS
      DO 100 J = 1, 2
      TGTZ(J) = 400
      TGTX(J) = 2000
      TGTY(J) = 2000
      CALL TGTRANGE (TTIME1,TGTZ,ZPHONE,NUMSSV,DEPTH,SSPVEL,HMVEL,
     +          TRANGE,J)
      CALL THREED (J,TGTX,TGTY,TGTZ,PHONEX,PHONEY,PHONEZ,TRANGE)
  100 CONTINUE
CC    CALC TRUE ERRORS
      ERFIX = 0.0
      DO 150 J = 1,2
       TERROR(J) = SQRT((XTGT(J)-TGTX(J))**2+(YTGT(J)-TGTY(J))**2+
     +            (ZTGT(J)-TGTZ(J))**2)
       IF (TERROR(J) .GT. 1000.0) ERFIX = 1.0
  150 CONTINUE
      IF (ERFIX .EQ. 1) THEN
        GOTO 79
      ELSE
      DO 200 J =1,2
       TESQ(J) = TESQ(J) + TERROR(J)**2
       TESUM(J) = TESUM(J) + TERROR(J)
  200  CONTINUE
CC    CALC RELATIVE ERRORS
        RERROR = SQRT((XTGT(1)-TGTX(1)-XTGT(2)+TGTX(2))**2+
     +             (YTGT(1)-TGTY(1)-YTGT(2)+TGTY(2))**2+
     +             (ZTGT(1)-TGTZ(1)-ZTGT(2)+TGTZ(2))**2)
        RESQ = RESQ + RERROR**2
        RESUM = RESUM + RERROR
      ENDIF
```

```
  900 CONTINUE
CC    CALC STATS ON TRUE ERRORS
CC       TGT1 RESULTS
      MEAN1 = TESUM(1)/NCOUNT
      VAR1 = SQRT(ABS(NCOUNT*TESQ(1)-TESUM(1)**2))/(NCOUNT*(NCOUNT-1))
CC       TGT2 RESULTS
      MEAN2 = TESUM(2)/NCOUNT
      VAR2 = SQRT(ABS(NCOUNT*TESQ(2)-TESUM(2)**2))/(NCOUNT*(NCOUNT-1))
CC    CALC STATS ON RELATIVE ERRORS
CC       RELATIVE RESULTS
      MEANR = RESUM/NCOUNT
      VARR = SQRT(ABS(NCOUNT*RESQ-RESUM**2))/(NCOUNT*(NCOUNT-1))
      CALL OUTPUT (MEAN1, VAR1, MEAN2, VAR2, MEANR, VARR,
     +            BTN,STN,TTN,NCOUNT)
      END
****************************************************************
*     INPUT SUBROUTINE TO ENTER IN SIMULATION VALUES           *
****************************************************************
      SUBROUTINE INPUT (NCOUNT, BTN, STN, TTN, SEED)
      REAL*8 SEED
      OPEN (UNIT = 1, FILE = 'INPUT.DAT')
      READ (1,*) NCOUNT, BTN, STN, TTN, SEED
      CLOSE (UNIT = 1, STATUS = 'KEEP')
      RETURN
      END

****************************************************************
*     PROFILE SUBROUTINE TO ENTER SOUND VELOCITY PROFILE       *
****************************************************************
      SUBROUTINE PROFILE (MAXSSV, NUMSSV, DEPTH, SSPVEL)
      REAL*4 DEPTH, SSPVEL
      INTEGER*4 MAXSSV, NUMSSV
      DIMENSION DEPTH(40), SSPVEL(40)
      OPEN (UNIT = 1, FILE = 'SSPROF.DAT')
      READ (1,*) NUMSSV
      IF (NUMSSV .GT. MAXSSV) THEN
         WRITE(*,*) '  MAXIMUM NUMBER OF SSP DATA POINTS EXCEEDED'
      ENDIF
      DO 10 I = 1, NUMSSV
         READ (1,*) DEPTH(I), SSPVEL(I)
   10 CONTINUE
      CLOSE (UNIT = 1, STATUS='KEEP')
      RETURN
      END


****************************************************************
*     POSITS SUBROUTINE TO ENTER BUOY, TRANSDUCER AND TARGET   *
****************************************************************
      SUBROUTINE POSITS (XBUOY,YBUOY,ZBUOY,XPHONE,YPHONE,ZPHONE,XTGT,
     +                   YTGT,ZTGT)
      REAL*4 XBUOY,YBUOY,ZBUOY,XPHONE,YPHONE,ZPHONE
      DIMENSION XBUOY(4),YBUOY(4),ZBUOY(4),XPHONE(4),YPHONE(4),
     +    ZPHONE(4),XTGT(2),YTGT(2),ZTGT(2)
      OPEN (UNIT = 2, FILE = 'BUOYLOC.DAT')
      DO 10 I = 1, 4
      READ(2,*) XBUOY(I),YBUOY(I),ZBUOY(I),XPHONE(I),YPHONE(I),
     +         ZPHONE(I)
   10 CONTINUE
      OPEN (UNIT = 1, FILE = 'TARGET.DAT')
      DO 20 J = 1, 2
```

```
      READ(1,*) XTGT(J), YTGT(J), ZTGT(J)
   20 CONTINUE
      RETURN
      END

***************************************************************
*     HMVELOC SUBROUTINE TO CALCULATE MEAN SOUND VELOCITY     *
***************************************************************
      SUBROUTINE HMVELOC (ZPING, ZRCVR, NUMSSV, DEPTH, SSPVEL,HMVEL)
      REAL*4 ZPING, ZRCVR, DEPTH, SSPVEL, HMVEL, GRADNT, HMV1, HMV2,T
      INTEGER*4 I, J, K, INIT, FINAL, NUMSSV
      DIMENSION DEPTH(NUMSSV), SSPVEL(NUMSSV)
C     ORDER THE DEPTH OF THE RECEIVER AND HYDROPHONE SO THAT THE
C     RECEIVER IS THE DEEPER OF THE TWO DEPTHS.
      T = 0.0
      INTE = 0
      IF (ZPING .GT. ZRCVR) THEN
         TEMP = ZPING
         ZPING = ZRCVR
         ZRCVR = TEMP
         INTE = 1
      ENDIF
C     INTERPOLATE THE SOUND SPEED AT THE PINGER DEPTH
      DO 1 I = 2, NUMSSV
        IF (ZPING .GT. DEPTH(I)) GOTO 1
        GRADNT = (SSPVEL(I) - SSPVEL(I-1))/(DEPTH(I) - DEPTH(I-1))
        HMV1 = SSPVEL(I-1) + GRADNT*(ZPING-DEPTH(I-1))
        INIT= I
        GO TO 2
    1 CONTINUE
C     INTERPOLATE THE SOUND SPEED AT THE HYDROPHONE DEPTH
    2 DO 3 J = 2, NUMSSV
        IF (ZRCVR .GT. DEPTH(J)) GOTO 3
        GRADNT = (SSPVEL(J) - SSPVEL(J-1))/(DEPTH(J) - DEPTH(J-1))
        HMV2 = SSPVEL(J-1) + GRADNT*(ZRCVR-DEPTH(J-1))
        FINAL = J-1
        GO TO 4
    3 CONTINUE
C     CALCULATE THE HARMONIC SOUND SPEED BETWEEN THE PINGER AND THE
C     HYDROPHONE
    4 IF (FINAL .LT. INIT) THEN
C
C     SOURCE AND RECEIVER IN SAME GRADIENT
C     TAKE AVERAGE OF TWO SOUND SPEEDS OF SOURCE AND RECEIVER
C
         GRADNT = (HMV2 - HMV1) / (ZRCVR - ZPING)
         IF (GRADNT .NE. 0.0) T = LOG(HMV2/HMV1)/GRADNT
         IF (GRADNT .EQ. 0.0) T = (ZRCVR - ZPING)/HMV2
         HMVEL = T/(ZRCVR-ZPING)
      ELSEIF (FINAL .EQ. INIT) THEN
C
C     SOURCE AND RECEIVER SPERATED BY ONE DEPTH GRADIENT
C
         GRADNT = (SSPVEL(INIT) - HMV1)/(DEPTH(INIT) - ZPING)
         IF (GRADNT .NE. 0.0) T = LOG(SSPVEL(INIT)/HMV1)/GRADNT
         IF (GRADNT .EQ. 0.0) T = (DEPTH(INIT)-ZPING)/SSPVEL(INIT)
         GRADNT = (HMV2-SSPVEL(FINAL))/(ZRCVR - DEPTH(FINAL))
         IF (GRADNT .NE. 0.0) T = T + LOG(HMV2/SSPVEL(FINAL))/GRADNT
         IF (GRADNT .EQ. 0.0) T = T + (ZRCVR-DEPTH(FINAL))/HMV2
         HMVEL = ABS((ZRCVR-ZPING)/T)
      ELSE
```

```
C
C     SOURCE AND RECEIVER SEPERATED BY MORE THAN ONE DEPTH
C
         GRADNT = (SSPVEL(INIT) - HMV1)/(DEPTH(INIT) - ZPING)
         IF (GRADNT .NE. 0.0) T = LOG(SSPVEL(INIT)/HMV1)/GRADNT
         IF (GRADNT .EQ. 0.0) T = (DEPTH(INIT)-ZPING)/SSPVEL(INIT)
         DO 5 K= INIT + 1, FINAL
           GRADNT = (SSPVEL(K)-SSPVEL(K-1))/(DEPTH(K)-DEPTH(K-1))
           IF (GRADNT .NE. 0.0) T = T+LOG(SSPVEL(K)/SSPVEL(K-1))/GRADNT
           IF (GRADNT .EQ. 0.0) T = T+(DEPTH(K)-DEPTH(K-1))/SSPVEL(K)
    5    CONTINUE
         GRADNT = (HMV2-SSPVEL(FINAL))/(ZRCVR - DEPTH(FINAL))
         IF (GRADNT .NE. 0.0) T = T + LOG(HMV2/SSPVEL(FINAL))/GRADNT
         IF (GRADNT .EQ. 0.0) T = T + (ZRCVR-DEPTH(FINAL))/HMV2
         HMVEL = ABS((ZRCVR-ZPING)/T)
      ENDIF
      IF (INTE .EQ. 1) THEN
        TEMP = ZPING
        ZPING = ZRCVR
        ZRCVR = TEMP
      ENDIF
      RETURN
      END

******************************************************************
*     HYDEPTH SUBROUTINE TO CALCULATE TRANSDUCER DEPTHS          *
******************************************************************
      SUBROUTINE HYDEPTH (ZCABLE, ZBUOY, NUMSSV, DEPTH,
     +    SSPVEL, HMVEL, BTIME, PHONEZ)
CC    SUBROUTINE TO CALCULATE THE DEPTH OF THE HYDROPHONE BELOW THE
CC    BUOY.
      REAL*4 ZCABLE, ZBUOY, DEPTH, SSPVEL,HMVEL,BTIME,ZOLD,PHONEZ
      INTEGER*4 NUMSSV
      DIMENSION DEPTH(40), SSPVEL(40),PHONEZ(4), BTIME(4,4),ZBUOY(4)
      DO 5 I = 1, 4
        ZB = ZBUOY(I)
        ZOLD = ZCABLE
        DO 3 KOUNT = 1, 10
      CALL HMVELOC(ZB, ZOLD, NUMSSV, DEPTH, SSPVEL, HMVEL)
      ZNEW = HMVEL * BTIME(I,I) + ZB
          IF (ABS(ZNEW-ZOLD) .LT. 1.0) GO TO 4
          ZOLD = ZNEW
    3   CONTINUE
    4   PHONEZ(I) = ZNEW
    5 CONTINUE
      RETURN
      END

******************************************************************
*     RPHONE SUBROUTINE TO CALCULATE RANGES BETWEEN TRANSDUCERS*
******************************************************************
      SUBROUTINE RPHONE (RANGE, HTIME, PHONEZ, NUMSSV, DEPTH, SSPVEL,
     +    HMVEL)
CC
CC    RANGE(1) = PHONE 1 TO PHONE 2
CC    RANGE(2) = PHONE 3 TO PHONE 2
CC    RANGE(3) = PHONE 1 TO PHONE 4
CC    RANGE(4) = PHONE 3 TO PHONE 4
CC    RANGE(5) = PHONE 2 TO PHONE 4
```

```
CC    RANGE(6) = PHONE 1 TO PHONE 3
      REAL*4 RANGE, HTIME, PHONEZ, DEPTH, SSPVEL,HMVEL,T
      INTEGER NUMSSV
      DIMENSION RANGE(6), HTIME(4,4), PHONEZ(4), DEPTH(40),
     +       SSPVEL(40)
      DO 1 I = 1, 4
    1 CONTINUE
      T = (HTIME(2,1) - HTIME(1,2)/3.0)*1.5
      CALL HMVELOC(PHONEZ(1), PHONEZ(2), NUMSSV, DEPTH, SSPVEL,
     +    HMVEL)
      RANGE(1) = T * HMVEL
      T = 0.5*HTIME(2,3) + 0.5*HTIME(3,2)
      CALL HMVELOC(PHONEZ(2), PHONEZ(3), NUMSSV, DEPTH, SSPVEL,
     +    HMVEL)
      RANGE(2) = T * HMVEL
      T = 0.5*HTIME(1,4) + 0.5*HTIME(4,1)
      CALL HMVELOC(PHONEZ(1), PHONEZ(4), NUMSSV, DEPTH, SSPVEL,
     +    HMVEL)
      RANGE(3) = T * HMVEL
      T = (HTIME(3,4) - HTIME(4,3)/3.0)*1.5
      CALL HMVELOC(PHONEZ(3), PHONEZ(4), NUMSSV, DEPTH, SSPVEL,
     +    HMVEL)
      RANGE(4) = T * HMVEL
      T = 0.75*HTIME(2,4) + 0.25*HTIME(4,2)
      CALL HMVELOC(PHONEZ(2), PHONEZ(4), NUMSSV, DEPTH, SSPVEL,
     +    HMVEL)
      RANGE(5) = T * HMVEL
      T = 0.25*HTIME(1,3) + 0.75*HTIME(3,1)
      CALL HMVELOC(PHONEZ(1), PHONEZ(3), NUMSSV, DEPTH, SSPVEL,
     +    HMVEL)
      RANGE(6) = T * HMVEL
      RETURN
      END

******************************************************************
*     TWOD SUBROUTINE TO CALCULATE TRANSDUCER POSITIONS          *
******************************************************************
      SUBROUTINE TWOD (RANGE, XP, YP, ZP, BUOYY)
      REAL*4 RANGE, BUOYY
      REAL*4 XP, YP, ZP, DX, DY, ANS, XSQR1, XSQR3,
     + YSQR1, YSQR3, RSQR1, RSQR3, RHS, DISCRIM, THETA, PHI
      DIMENSION BUOYY(4), XP(4), YP(4), ZP(4), RANGE(6),
     + ANS(2,2)
      XP(3)=SQRT(RANGE(6)**2 - (ZP(1)-ZP(3))**2)
      YP(3)=0.0
      DX=XP(1)-XP(3)
      DY=YP(1)-YP(3)
      RSQR1=RANGE(1)**2-(ZP(1)-ZP(2))**2
      RSQR3=RANGE(2)**2-(ZP(2)-ZP(3))**2
      XSQR1=XP(1)**2
      XSQR3=XP(3)**2
      YSQR1=YP(1)**2
      YSQR3=YP(3)**2
      RHS = (XSQR1 + YSQR1 - XSQR3 - YSQR3 +RSQR3 - RSQR1)*0.5
      IF (DY .EQ. 0.0) THEN
        THETA = RHS/DX
        A = 1
        B = -2 * YP(1)
        C = (THETA)**2 + XSQR1 + YSQR1 - (2 * THETA * XP(1)) - RSQR1
        DISCRIM = B**2 - (4 * A * C)
```

```
  ANS(1,2) = (-B + SQRT(ABS(DISCRIM)))/(2*A)
  ANS(2,2) = (-B/A - ANS(1,2))
  ANS(1,1) = THETA
  ANS(2,1) = THETA
ELSE
  THETA = RHS/DY
  PHI = DX/DY
  A = 1.0 + PHI**2
  B = 2*(YP(1)*PHI-THETA*PHI-XP(1))
  C = XSQR1 + (THETA)**2 + YSQR1 - RSQR1 - 2.0 * THETA * YP(1)
  DISCRIM = B**2 - (4 * A * C)
  ANS(1,1) = (-B +SQRT(DISCRIM))/(2*A)
  ANS(2,1) = (-B/A - ANS(1,1))
  ANS(1,2) = THETA - PHI*ANS(1,1)
  ANS(2,2) = THETA - PHI*ANS(2,1)
ENDIF
  IF (ABS(ANS(1,2) - BUOYY(2)) .LT. 1000) THEN
  XP(2) = ANS(1,1)
  YP(2) = ANS(1,2)
ELSE
  XP(2) = ANS(2,1)
  YP(2) = ANS(2,2)
ENDIF
DX=XP(2)-XP(3)
DY=YP(2)-YP(3)
RSQR1=RANGE(5)**2-(ZP(2)-ZP(4))**2
RSQR3=RANGE(4)**2-(ZP(3)-ZP(4))**2
XSQR1=XP(2)**2
XSQR3=XP(3)**2
YSQR1=YP(2)**2
YSQR3=YP(3)**2
RHS = (XSQR1 + YSQR1 - XSQR3 - YSQR3 +RSQR3 - RSQR1)*0.5
IF (DY .EQ. 0.0) THEN
  THETA = RHS/DX
  A = 1
  B = -2 * YP(2)
  C = (THETA)**2 + XSQR1 + YSQR1 - (2 * THETA * XP(2)) - RSQR1
  DISCRIM = B**2 - (4 * A * C)
  ANS(1,2) = (-B + SQRT(DISCRIM))/(2*A)
  ANS(2,2) = (-B/A - ANS(1,2))
  ANS(1,1) = THETA
  ANS(2,1) = THETA
ELSE
  THETA = RHS/DY
  PHI = DX/DY
  A = 1.0 + PHI**2
  B = 2*(YP(2)*PHI-THETA*PHI-XP(2))
  C = XSQR1 + (THETA)**2 + YSQR1 - RSQR1 - 2.0 * THETA * YP(2)
  DISCRIM = B**2 - (4 * A * C)
  ANS(1,1) = (-B +SQRT(ABS(DISCRIM)))/(2*A)
  ANS(2,1) = (-B/A - ANS(1,1))
  ANS(1,2) = THETA - PHI*ANS(1,1)
  ANS(2,2) = THETA - PHI*ANS(2,1)
  ENDIF
IF (ABS(ANS(1,2) - BUOYY(4)) .LT. 1000) THEN
  XP(4) = ANS(1,1)
  YP(4) = ANS(1,2)
ELSE
```

```
        XP(4) = ANS(2,1)
        YP(4) = ANS(2,2)
      ENDIF
      RETURN
      END

*****************************************************************
*     TGTRANGE SUBROUTINE TO CALCULATE RANGE TO TARGETS         *
*****************************************************************
      SUBROUTINE TGTRANGE (TTIME,TGTZ, ZPHONE,NUMSSV,DEPTH,SSPVEL,
     +          HMVEL, TRANGE, J)
      REAL*4 TGTZ,ZPHONE,DEPTH,SSPVEL,HMVEL,TRANGE,TTIME
      INTEGER*4 J, NUMSSV
      DIMENSION TGTZ(2),ZPHONE(4),DEPTH(40),SSPVEL(40),TRANGE(2,4),
     +           TTIME(2,4)
      DO 10 I = 1, 4
      CALL HMVELOC(TGTZ(J), ZPHONE(I), NUMSSV, DEPTH, SSPVEL, HMVEL)
      TRANGE(J,I) = TTIME(J,I) * HMVEL
   10 CONTINUE
      RETURN
      END

*****************************************************************
*     THREED SUBROUTINE TO CALCULATE TARGET POSITIONS           *
*****************************************************************
      SUBROUTINE THREED (J,TGTX,TGTY,TGTZ,PHONEX,PHONEY,PHONEZ,RM)
      REAL*4 TGTX,TGTY,TGTZ,PHONEX,PHONEY,PHONEZ,PRPX,PRPY,PRPZ,
     + XTO,YTO,ZTO,A1,A2,A3,A4,A5,A6,A7,A8,A9,RO,DR,RM
      INTEGER*4 J, IN
      DIMENSION TGTX(2),TGTY(2),TGTZ(2),PHONEX(4),PHONEY(4),
     + PHONEZ(4),RO(4),RM(2,4)
      IN = 0
    5 CONTINUE
      XTO=TGTX(J)
      YTO=TGTY(J)
      ZTO=TGTZ(J)
      A1=0.0
      A2=0.0
      A3=0.0
      A4=0.0
      A5=0.0
      A6=0.0
      A7=0.0
      A8=0.0
      A9=0.0

      DO 10 I= 1, 4
       RO(I) = SQRT((XTO-PHONEX(I))**2+(YTO-PHONEY(I))**2+
     +        (ZTO-PHONEZ(I))**2)
       DR = RM(J,I) - RO(I)
       PRPX = (XTO - PHONEX(I)) / RO(I)
       PRPY = (YTO - PHONEY(I)) / RO(I)
       PRPZ = (ZTO - PHONEZ(I)) / RO(I)
       A1 = A1 + PRPX*PRPX
       A2 = A2 + PRPX*PRPY
       A3 = A3 + PRPX*PRPZ
       A4 = A4 + PRPY*PRPY
       A5 = A5 + PRPY*PRPZ
       A6 = A6 + PRPZ*PRPZ
       A7 = A7 + DR*PRPX
       A8 = A8 + DR*PRPY
```

```
       A9 = A9 + DR*PRPZ
   10 CONTINUE
      DETAO = A1*(A4*A6-A5*A5)-A2*(A2*A6-A5*A3)+A3*(A2*A5-A4*A3)
      DETA1 = A7*(A4*A6-A5*A5)-A8*(A2*A6-A5*A3)+A9*(A2*A5-A4*A3)
      DETA2 = A1*(A8*A6-A9*A5)-A2*(A7*A6-A9*A3)+A3*(A7*A5-A8*A3)
      DETA3 = A1*(A4*A9-A5*A8)-A2*(A2*A9-A5*A7)+A3*(A2*A8-A4*A7)
      TGTX(J) = XTO + DETA1/DETAO
      TGTY(J) = YTO + DETA2/DETAO
      TGTZ(J) = ZTO + DETA3/DETAO
      IF (((DETA1**2+DETA2**2+DETA3**2)/DETAO) .LT. 1.0 ) THEN
        GO TO 900
      ENDIF
      IN = IN + 1
      IF (IN .GT. 25) THEN
        GO TO 900
      ENDIF
      GOTO 5
  900 RETURN
      END

****************************************************************
*     OUTPUT SUBROUTINE TO OUTPUT DATA RESULTS                 *
****************************************************************
      SUBROUTINE OUTPUT (MEAN1,VAR1,MEAN2,VAR2,MEANR,VARR,
     +                   BTN,STN,TTN,NCOUNT)
      REAL*4 MEAN1,VAR1,MEAN2,VAR2,MEANR,VARR,BTN,STN,TTN
      INTEGER*4 NCOUNT
      OPEN (UNIT = 8, FILE = 'OUTPUT3.DAT')
      WRITE(8,*) '                                MEAN    STAND DEV '
      WRITE(8,1000)  MEAN1, VAR1
      WRITE(8,1001)  MEAN2, VAR2
      WRITE(8,1002)  MEANR, VARR
      WRITE(8,*) '  NUMBER OF ITERATIONS, BUOY    TRANS    TARGET '
      WRITE(8,1003) NCOUNT, BTN, STN,TTN
 1000 FORMAT (' TRUE ERROR FOR TARGET 1 = ',F8.3, F8.4)
 1001 FORMAT (' TRUE ERROR FOR TARGET 2 = ',F8.3, F8.4)
 1002 FORMAT (' RELATIVE ERROR          = ',F8.3, F8.4)
 1003 FORMAT ( 3X,I6,15X, F8.4, F8.4, F8.4)
      RETURN
      END

****************************************************************
*     URAND SUBROUTINE TO GENERATE RANDOM NUMBERS              *
****************************************************************
      SUBROUTINE URAND(A,B,SEED)
      REAL*8 SEED, PI, R1, R2
      PI=3.14159265358979
      R1=(SEED+PI)**504.D-2
      R1=R1-DINT(R1)
      SEED=R1
      R2=(SEED+PI)**504.D-2
      R2=R2-DINT(R2)
      SEED=R2
      R=SQRT(-2*LOG(R1))
      T=2*PI*R2
      A=R*SIN(T)
      B=R*COS(T)
      RETURN
      END
```

# APPENDIX B

The following is the output data from the simulations conducted for each of the four cases discussed in Chapter IV.

Case number 1:

```
BUOYLOC.DAT
    0.0,      0.0,      5.0,     0.0,      0.0, 1000.0,
 3000.0,   5196.0,      5.0,  3000.0,   5196.0, 1000.0,
 6000.0,      0.0,      5.0,  6000.0,      0.0, 1000.0,
 9000.0,   5196.0,      5.0,  9000.0,   5196.0, 1000.0,
```

```
TARGET.DAT
 2036.0, 2036.0, 400.0,
 1965.0, 1965.0, 400.0,
```

```
OUTPUT.DAT
                                  MEAN    STAND DEV
 TRUE ERROR FOR TARGET 1 =        .002    .0000
 TRUE ERROR FOR TARGET 2 =        .001    .0000
 RELATIVE ERROR          =        .001    .0000
   NUMBER OF ITERATIONS, BUOY    TRANS    TARGET
     1000                .0000   .0000    .0000

                                  MEAN    STAND DEV
 TRUE ERROR FOR TARGET 1 =      19.829    .0122
 TRUE ERROR FOR TARGET 2 =      19.695    .0122
 RELATIVE ERROR          =      14.982    .0084
   NUMBER OF ITERATIONS, BUOY    TRANS    TARGET
     1000                .0010   .0010    .0010

                                  MEAN    STAND DEV
 TRUE ERROR FOR TARGET 1 =      39.791    .0247
 TRUE ERROR FOR TARGET 2 =      39.564    .0248
 RELATIVE ERROR          =      30.101    .0169
   NUMBER OF ITERATIONS, BUOY    TRANS    TARGET
     1000                .0020   .0020    .0020

                                  MEAN    STAND DEV
 TRUE ERROR FOR TARGET 1 =      60.054    .0380
 TRUE ERROR FOR TARGET 2 =      59.780    .0383
 RELATIVE ERROR          =      45.545    .0260
   NUMBER OF ITERATIONS, BUOY    TRANS    TARGET
     1000                .0030   .0030    .0030
```

```
                         MEAN     STAND DEV
TRUE ERROR FOR TARGET 1 =   80.887    .0530
TRUE ERROR FOR TARGET 2 =   80.616    .0537
RELATIVE ERROR          =   61.654    .0362
  NUMBER OF ITERATIONS, BUOY     TRANS     TARGET
    1000                .0040    .0040    .0040

                          MEAN     STAND DEV
TRUE ERROR FOR TARGET 1 =  102.065    .0688
TRUE ERROR FOR TARGET 2 =  102.172    .0718
RELATIVE ERROR          =   78.586    .0486
  NUMBER OF ITERATIONS, BUOY     TRANS     TARGET
    1000                .0050    .0050    .0050

                          MEAN     STAND DEV
TRUE ERROR FOR TARGET 1 =  123.875    .0910
TRUE ERROR FOR TARGET 2 =  123.837    .0939
RELATIVE ERROR          =   98.655    .0726
  NUMBER OF ITERATIONS, BUOY     TRANS     TARGET
    1000                .0060    .0060    .0060

                          MEAN     STAND DEV
TRUE ERROR FOR TARGET 1 =  145.096    .1044
TRUE ERROR FOR TARGET 2 =  146.658    .1133
RELATIVE ERROR          =  118.392    .0931
  NUMBER OF ITERATIONS, BUOY     TRANS     TARGET
    1000                .0070    .0070    .0070

                          MEAN     STAND DEV
TRUE ERROR FOR TARGET 1 =  164.285    .1207
TRUE ERROR FOR TARGET 2 =  165.526    .1290
RELATIVE ERROR          =  138.519    .1075
  NUMBER OF ITERATIONS, BUOY     TRANS     TARGET
    1000                .0080    .0080    .0080

                          MEAN     STAND DEV
TRUE ERROR FOR TARGET 1 =  183.286    .1335
TRUE ERROR FOR TARGET 2 =  179.448    .1264
RELATIVE ERROR          =  154.021    .1197
  NUMBER OF ITERATIONS, BUOY     TRANS     TARGET
    1000                .0090    .0090    .0090

                          MEAN     STAND DEV
TRUE ERROR FOR TARGET 1 =  198.663    .1406
TRUE ERROR FOR TARGET 2 =  199.931    .1478
RELATIVE ERROR          =  172.593    .1288
  NUMBER OF ITERATIONS, BUOY     TRANS     TARGET
    1000                .0100    .0100    .0100
```

Case number 2:

```
BUOYLOC.DAT
    0.0,      0.0,     5.0,    0.0,      0.0, 1000.0,
 3000.0,   5196.0,     5.0, 3000.0,   5196.0, 1000.0,
 6000.0,      0.0,     5.0, 6000.0,      0.0, 1000.0,
 9000.0,   5196.0,     5.0, 9000.0,   5196.0, 1000.0,

TARGET.DAT
 1000.0, 2000.0, 400.0,
 3000.0, 4236.0, 400.0,

OUTPUT.DAT
                                    MEAN    STAND DEV
 TRUE ERROR FOR TARGET 1 =          .001    .0000
 TRUE ERROR FOR TARGET 2 =          .002    .0000
 RELATIVE ERROR          =          .001    .0000
   NUMBER OF ITERATIONS, BUOY     TRANS     TARGET
     1000                  .0000    .0000    .0000

                                    MEAN    STAND DEV
 TRUE ERROR FOR TARGET 1 =        23.314    .0147
 TRUE ERROR FOR TARGET 2 =        12.332    .0058
 RELATIVE ERROR          =        19.779    .0111
   NUMBER OF ITERATIONS, BUOY     TRANS     TARGET
     1000                  .0010    .0010    .0010

                                    MEAN    STAND DEV
 TRUE ERROR FOR TARGET 1 =        46.881    .0300
 TRUE ERROR FOR TARGET 2 =        24.694    .0116
 RELATIVE ERROR          =        39.783    .0226
   NUMBER OF ITERATIONS, BUOY     TRANS     TARGET
     1000                  .0020    .0020    .0020

                                    MEAN    STAND DEV
 TRUE ERROR FOR TARGET 1 =        71.010    .0467
 TRUE ERROR FOR TARGET 2 =        37.106    .0175
 RELATIVE ERROR          =        60.300    .0355
   NUMBER OF ITERATIONS, BUOY     TRANS     TARGET
     1000                  .0030    .0030    .0030

                                    MEAN    STAND DEV
 TRUE ERROR FOR TARGET 1 =        96.323    .0673
 TRUE ERROR FOR TARGET 2 =        49.592    .0236
 RELATIVE ERROR          =        81.926    .0521
   NUMBER OF ITERATIONS, BUOY     TRANS     TARGET
     1000                  .0040    .0040    .0040
```

```
                                MEAN    STAND DEV
TRUE ERROR FOR TARGET 1 =  122.966    .0913
TRUE ERROR FOR TARGET 2 =   62.178    .0299
RELATIVE ERROR          =  104.788    .0721
  NUMBER OF ITERATIONS, BUOY    TRANS    TARGET
     1000                .0050   .0050   .0050

                                MEAN    STAND DEV
TRUE ERROR FOR TARGET 1 =  148.598    .1136
TRUE ERROR FOR TARGET 2 =   74.547    .0364
RELATIVE ERROR          =  127.339    .0906
  NUMBER OF ITERATIONS, BUOY    TRANS    TARGET
     1000                .0060   .0060   .0060

                                MEAN    STAND DEV
TRUE ERROR FOR TARGET 1 =  174.871    .1362
TRUE ERROR FOR TARGET 2 =   87.047    .0428
RELATIVE ERROR          =  150.839    .1110
  NUMBER OF ITERATIONS, BUOY    TRANS    TARGET
     1000                .0070   .0070   .0070

                                MEAN    STAND DEV
TRUE ERROR FOR TARGET 1 =  194.483    .1489
TRUE ERROR FOR TARGET 2 =   99.075    .0498
RELATIVE ERROR          =  169.387    .1218
  NUMBER OF ITERATIONS, BUOY    TRANS    TARGET
     1000                .0080   .0080   .0080

                                MEAN    STAND DEV
TRUE ERROR FOR TARGET 1 =  209.383    .1497
TRUE ERROR FOR TARGET 2 =  111.316    .0572
RELATIVE ERROR          =  183.449    .1207
  NUMBER OF ITERATIONS, BUOY    TRANS    TARGET
     1000                .0090   .0090   .0090

                                MEAN    STAND DEV
TRUE ERROR FOR TARGET 1 =  232.755    .1665
TRUE ERROR FOR TARGET 2 =  123.187    .0622
RELATIVE ERROR          =  206.640    .1412
  NUMBER OF ITERATIONS, BUOY    TRANS    TARGET
     1000                .0100   .0100   .0100
```

Case number 3:

```
BUOYLOC.DAT
   0.0,      0.0,     5.0,    0.0,      0.0, 1000.0,
 3000.0,  5196.0,     5.0, 3000.0,   5196.0,  900.0,
 6000.0,     0.0,     5.0, 6000.0,      0.0,  800.0,
 9000.0,  5196.0,     5.0, 9000.0,   5196.0,  700.0,

TARGET.DAT
 2036.0, 2036.0, 400.0,
 1965.0, 1965.0, 400.0,

OUTPUT.DAT
                                   MEAN    STAND DEV
 TRUE ERROR FOR TARGET 1 =         .002    .0000
 TRUE ERROR FOR TARGET 2 =         .001    .0000
 RELATIVE ERROR          =         .001    .0000
   NUMBER OF ITERATIONS, BUOY    TRANS    TARGET
     1000                .0000   .0000    .0000

                                   MEAN    STAND DEV
 TRUE ERROR FOR TARGET 1 =      21.890    .0139
 TRUE ERROR FOR TARGET 2 =      21.668    .0138
 RELATIVE ERROR          =      16.342    .0095
   NUMBER OF ITERATIONS, BUOY    TRANS    TARGET
     1000                .0010   .0010    .0010

                                   MEAN    STAND DEV
 TRUE ERROR FOR TARGET 1 =      44.028    .0283
 TRUE ERROR FOR TARGET 2 =      43.639    .0283
 RELATIVE ERROR          =      32.941    .0193
   NUMBER OF ITERATIONS, BUOY    TRANS    TARGET
     1000                .0020   .0020    .0020

                                   MEAN    STAND DEV
 TRUE ERROR FOR TARGET 1 =      66.771    .0445
 TRUE ERROR FOR TARGET 2 =      66.268    .0443
 RELATIVE ERROR          =      50.214    .0303
   NUMBER OF ITERATIONS, BUOY    TRANS    TARGET
     1000                .0030   .0030    .0030

                                   MEAN    STAND DEV
 TRUE ERROR FOR TARGET 1 =      90.263    .0629
 TRUE ERROR FOR TARGET 2 =      89.303    .0620
 RELATIVE ERROR          =      68.667    .0436
   NUMBER OF ITERATIONS, BUOY    TRANS    TARGET
     1000                .0040   .0040    .0040
```

```
                          MEAN     STAND DEV
TRUE ERROR FOR TARGET 1 =  114.024    .0834
TRUE ERROR FOR TARGET 2 =  114.166    .0862
RELATIVE ERROR          =   89.502    .0640
  NUMBER OF ITERATIONS, BUOY     TRANS     TARGET
    1000                .0050    .0050    .0050

                          MEAN     STAND DEV
TRUE ERROR FOR TARGET 1 =  137.825    .1066
TRUE ERROR FOR TARGET 2 =  136.027    .1048
RELATIVE ERROR          =  111.802    .0864
  NUMBER OF ITERATIONS, BUOY     TRANS     TARGET
    1000                .0060    .0060    .0060

                          MEAN     STAND DEV
TRUE ERROR FOR TARGET 1 =  164.585    .1340
TRUE ERROR FOR TARGET 2 =  159.147    .1289
RELATIVE ERROR          =  136.492    .1275
  NUMBER OF ITERATIONS, BUOY     TRANS     TARGET
    1000                .0070    .0070    .0070

                          MEAN     STAND DEV
TRUE ERROR FOR TARGET 1 =  178.254    .1318
TRUE ERROR FOR TARGET 2 =  174.554    .1306
RELATIVE ERROR          =  146.033    .1085
  NUMBER OF ITERATIONS, BUOY     TRANS     TARGET
    1000                .0080    .0080    .0080

                          MEAN     STAND DEV
TRUE ERROR FOR TARGET 1 =  201.941    .1519
TRUE ERROR FOR TARGET 2 =  198.101    .1477
RELATIVE ERROR          =  178.767    .1563
  NUMBER OF ITERATIONS, BUOY     TRANS     TARGET
    1000                .0090    .0090    .0090

                          MEAN     STAND DEV
TRUE ERROR FOR TARGET 1 =  217.521    .1554
TRUE ERROR FOR TARGET 2 =  212.118    .1509
RELATIVE ERROR          =  186.845    .1463
  NUMBER OF ITERATIONS, BUOY     TRANS     TARGET
    1000                .0100    .0100    .0100
```

Case number 4:

BUOYLOC.DAT

```
    0.0,      0.0,     5.0,     0.0,      0.0, 1000.0,
 3000.0,   5196.0,     5.0, 3000.0,   5196.0,  900.0,
 6000.0,      0.0,     5.0, 6000.0,      0.0,  800.0,
 9000.0,   5196.0,     5.0, 9000.0,   5196.0,  700.0,
```

TARGET.DAT

```
 1000.0, 2000.0, 400.0,
 3000.0, 4236.0, 400.0,
```

OUTPUT.DAT

```
                                MEAN    STAND DEV
 TRUE ERROR FOR TARGET 1 =        .001   .0000
 TRUE ERROR FOR TARGET 2 =        .001   .0000
 RELATIVE ERROR          =        .001   .0000
   NUMBER OF ITERATIONS, BUOY     TRANS    TARGET
     1000                .0000    .0000    .0000

                                MEAN    STAND DEV
 TRUE ERROR FOR TARGET 1 =      24.415   .0156
 TRUE ERROR FOR TARGET 2 =      13.655   .0068
 RELATIVE ERROR          =      20.218   .0114
   NUMBER OF ITERATIONS, BUOY     TRANS    TARGET
     1000                .0010    .0010    .0010

                                MEAN    STAND DEV
 TRUE ERROR FOR TARGET 1 =      49.162   .0319
 TRUE ERROR FOR TARGET 2 =      27.372   .0136
 RELATIVE ERROR          =      40.721   .0233
   NUMBER OF ITERATIONS, BUOY     TRANS    TARGET
     1000                .0020    .0020    .0020

                                MEAN    STAND DEV
 TRUE ERROR FOR TARGET 1 =      74.694   .0504
 TRUE ERROR FOR TARGET 2 =      41.198   .0208
 RELATIVE ERROR          =      61.921   .0372
   NUMBER OF ITERATIONS, BUOY     TRANS    TARGET
     1000                .0030    .0030    .0030

                                MEAN    STAND DEV
 TRUE ERROR FOR TARGET 1 =     102.194   .0764
 TRUE ERROR FOR TARGET 2 =      55.192   .0283
 RELATIVE ERROR          =      84.955   .0585
   NUMBER OF ITERATIONS, BUOY     TRANS    TARGET
     1000                .0040    .0040    .0040
```

```
                                  MEAN    STAND DEV
TRUE ERROR FOR TARGET 1 =  130.319   .1024
TRUE ERROR FOR TARGET 2 =   69.366   .0364
RELATIVE ERROR          =  109.312   .0842
  NUMBER OF ITERATIONS, BUOY    TRANS    TARGET
    1000                .0050   .0050   .0050

                                  MEAN    STAND DEV
TRUE ERROR FOR TARGET 1 =  154.690   .1222
TRUE ERROR FOR TARGET 2 =   83.145   .0447
RELATIVE ERROR          =  130.410   .0952
  NUMBER OF ITERATIONS, BUOY    TRANS    TARGET
    1000                .0060   .0060   .0060

                                  MEAN    STAND DEV
TRUE ERROR FOR TARGET 1 =  183.675   .1501
TRUE ERROR FOR TARGET 2 =   97.235   .0547
RELATIVE ERROR          =  156.063   .1203
  NUMBER OF ITERATIONS, BUOY    TRANS    TARGET
    1000                .0070   .0070   .0070

                                  MEAN    STAND DEV
TRUE ERROR FOR TARGET 1 =  202.344   .1559
TRUE ERROR FOR TARGET 2 =  112.773   .0695
RELATIVE ERROR          =  173.353   .1238
  NUMBER OF ITERATIONS, BUOY    TRANS    TARGET
    1000                .0080   .0080   .0080

                                  MEAN    STAND DEV
TRUE ERROR FOR TARGET 1 =  221.332   .1593
TRUE ERROR FOR TARGET 2 =  126.445   .0794
RELATIVE ERROR          =  194.134   .1413
  NUMBER OF ITERATIONS, BUOY    TRANS    TARGET
    1000                .0090   .0090   .0090

                                  MEAN    STAND DEV
TRUE ERROR FOR TARGET 1 =  239.836   .1740
TRUE ERROR FOR TARGET 2 =  140.192   .0861
RELATIVE ERROR          =  210.705   .1459
  NUMBER OF ITERATIONS, BUOY    TRANS    TARGET
    1000                .0100   .0100   .0100
```

# LIST OF REFERENCES

1. Kinsler, L. E., and others, *Fundamentals of Acoustics*, 3rd ed., John Wiley and Sons, 1982.

# BIBLIOGRAPHY

Pacific Missile Test Range Technical Note 3430-45-73., *An Improved Formula for Underwater Effective Sound Velocity*, by Larry A. Anderson, December, 1973.

Anderson, L. A., *Tracking Algorithms for the Auxiliary Tracking System (ATS)*, paper prepared for evaluation of ATS design review, 28 June, 1990.

Anton, H., and Rorres, C., *Elementary Linear Algebra with Applications*, John Wiley and Sons, 1987.

Naval Undersea Warfare Engineering Station Report 16623-100006, *Programmable Pinger Theory of Operation*, by M. Barlow, July, 1985.

Naval Undersea Warfare Engineering Station Report 16623-100147, *SFSK Receiver Theory of Operation*, by M. Barlow, July, 1985.

Hoover, S. V., and Perry, R. F., *Simulation: A Problem-Solving Approach*, Addison-Wesley Publishing Co., 1989.

Walpole, R. E., and Myers, R. H., *Probability and Statistics for Engineers and Scientists*, 4th ed., Macmillan Publishing Co., 1989.

Naval Undersea Warfare Engineering Station Program, *ATS Survey Software Operating Procedures on HP 9000/550*, June, 1990.

## INITIAL DISTRIBUTION LIST

| | | |
|---|---|---|
| 1. | Defense Technical Information Center<br>Cameron Station<br>Alexandria, VA 22304-6145 | 2 |
| 2. | Library, Code 52<br>Naval Postgraduate School<br>Monterey, CA 93943-5002 | 2 |
| 3. | Superintendent<br>Attn: Prof R.R. Read, Code OR/Re<br>Naval Postgraduate School<br>Monterey, CA 93943-5000 | 1 |
| 4. | Superintendent<br>Attn: Prof A.B. Coppens, Code PH/Cz<br>Naval Postgraduate School<br>Monterey, CA 93943-5000 | 1 |
| 5. | Commanding Officer<br>Attn: John Palsha, Code 70E3<br>Naval Undersea Warfare Engineering Station<br>Keyport, WA 98345-5000 | 1 |
| 6. | Commanding Officer<br>Attn: George Case, Code 70D1<br>Naval Undersea Warfare Engineering Station<br>Keyport, WA 98345-5000 | 1 |
| 7. | Commanding Officer<br>Attn: Larry Anderson, Code 70E2<br>Naval Undersea Warfare Engineering Station<br>Keyport, WA 98345-5000 | 1 |