AN ABSTRACT OF THE THESIS OF

Hansjoerg Haisch for the degree of Master of Science in Electrical and Computer Engineering presented on December 12, 1983.

Title: Passive Source Tracking From Spatially Correlated Angle-of-Arrival Data.

Abstract approved: Redacted for privacy
Rudolf S. Engelbrecht

This thesis presents a simulation technique for the post-detection of a point-source trajectory in an isotropic, stationary random medium. The two-dimensional source location is estimated from spatially correlated angle-of-arrival data which are collected simultaneously at two sensor positions. We assume that the data are collected at discrete, constant time intervals and consist of true (unbiased) source angles plus zero-mean, spatially correlated angular white noise with equal time- and direction-independent variances at both sensors and negligible higher moments. Smoothing of the tracking data by means of an asymmetric smoothing technique gives an optimum source trajectory estimate provided that, for the duration of the smoothing time interval, the source travels

at constant (but unknown) speed along a trajectory having constant (but unknown) radius of curvature.

A mathematical model based on these assumptions is developed for this configuration and tested by simulation.

For a limited range of angular noise variance we achieve good agreement between theory and simulation for cases of negative and positive spatial correlation, as well as for the uncorrelated case.

In this thesis we first examine the case of a non-moving source and then apply the results to the general trajectory estimation of a moving target, testing the new smoothing technique and comparing it with an extended Kalman filtering technique.

Passive Source Tracking From Spatially Correlated Angle-of-Arrival Data

by

Hansjoerg Haisch

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Completed December 12, 1983

Commencement June 1984

APPROVED:

Redacted for privacy

Professor of Electrical and Computer Engineering
in charge of major

Redacted for privacy

Head of Electrical and Computer Engineering Department

Redacted for privacy

Dean of Graduate School

Date thesis is presented December 12, 1983

Typed by researcher for Hansjoerg Haisch

ACKNOWLEDGMENT

I thank my major professor, Rudolf S. Engelbrecht for his great assistance throughout my graduate study. I also thank the other members of my committee J. L. Saugen, T. K. Plant, and O. A. Boedtker for their support. I am also grateful for R. C. Rathja's assistance in mastering the department computer system.

TABLE OF CONTENTS

LIST OF FIGURES

Passive Source Tracking
From Spatially Correlated Angle-of-Arrival Data

I. INTRODUCTION

SONAR and RADAR systems are used in applications where sources have to be located over a large range. SONAR is mainly used for underwater location which not only includes its military implementation for submarine location; fishery and sea-bottom geology are only two examples for its wide range of application.

In passive Sonar systems source location is achieved by detecting the random radiation emitted by a source and processing of the received data. The radiation consists of acoustic or electromagnetic waves of unknown structure.

According to the approach in Ref.1 for the two-dimensional case, estimation requires simultaneous wave-front angle-of-arrival measurements by at least two separate sensors. An equivalent way is using time-of-arrival measurements by at least three sensors.

The angles measured at the sensors can be regarded as random variables which have a mean indicating the true (unbiased) source location and constant, source direction independent variance. The source location, however, estimated from these angular data by direct calculation has a bias and variance depending on the source-sensor geometry

and the spatial data correlation.

It is necessary to determine these quantities before applying a smoothing procedure to get reliable results in estimating the source trajectory.

We concentrate in our investigation on source-sensor geometries where the source is located at distances ranging from very close to the sensors to about three times the sensor separation.

A mathematical model for the stationary target descibing estimator bias and variance and their distribution in the x-y plane is introduced in section II. In section III this model is tested by simulation. In section IV we introduce a new smoothing technique which is testet by simulating a moving target. Section V presents an extended Kalman filter model which is compared with the new smoothing technique by simulation. The content of this thesis is summarized and conclusions are drawn in Section VI. Appendix A gives the derivation of the asymmetry ratio l/u used in the new smoothing technique. Appendix B shows the algorithm for generating N correlated gaussian number pairs. Appendix C lists important parts of the simulation programms.

The simulation was done on the computer system HP 1000 using its built-in random generators, the system plotting software, and the system vector instruction set.

All simulation programs are written in Standard FORTRAN 77 and with minor exceptions are applicable to other computer systems.

II. MATHEMATICAL MODEL

In this section we present equations for the bias and variance of source location estimators calculated from correlated angle-of-arrival data. As can be seen from Figure 1 the two sensors are located at X = +D and X = -D in the x-y plane. The true source location at (x_s, y_s) and (R,θ) in Cartesian and polar coordinates, respectively.

The true source angles at sensors I and II are θ_I and θ_{II} measured counterclockwise from the positive x-direction.

In terms of the true source position coordinates (x_s, y_s) and sensor separation 2D, the true source angles at sensors I and II are defined as follows:

$$\theta_1 = \tan^{-1} \frac{Y_s}{X_s + D} \tag{1}$$

$$\theta_2 = \tan^{-1} \frac{Y_s}{X_s - D} \tag{2}$$

The inversion of (1) and (2) leads to an expression for the true source location coordinates in terms of θ_1, θ_2 and D:

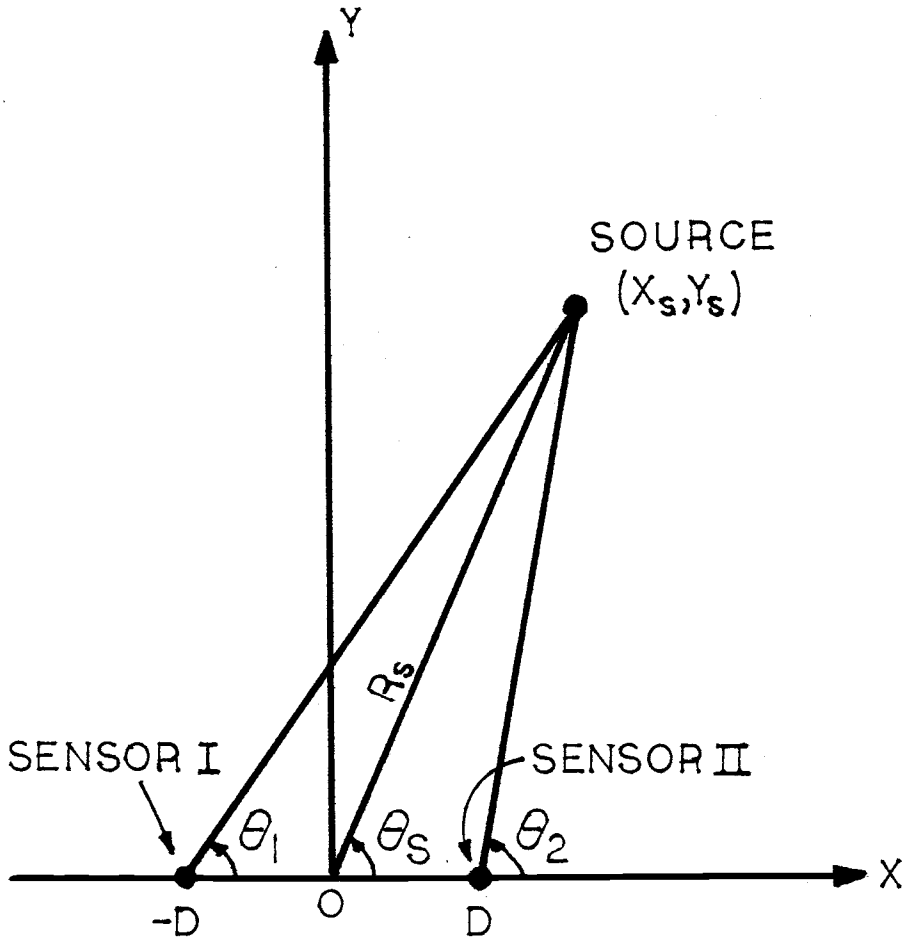

Figure 1. Source-sensor geometry

$$X_s = D\left[\frac{\sin(\theta_2+\theta_1)}{\sin(\theta_2-\theta_1)}\right] \tag{3}$$

$$Y_s = D\left[\frac{\cos(\theta_2-\theta_1)-\cos(\theta_2+\theta_1)}{\sin(\theta_2-\theta_1)}\right] \tag{4}$$

$$R = D\left[\frac{1+\cos^2(\theta_2-\theta_1)-2\cos(\theta_2-\theta_1)\cos(\theta_2+\theta_1)}{\sin^2(\theta_2-\theta_1)}\right] \tag{5}$$

$$\theta = \tan^{-1}\left[\frac{\cos(\theta_2-\theta_1)-\cos(\theta_2+\theta_1)}{\sin(\theta_2+\theta_1)}\right] \tag{6}$$

(Note that (3) - (6) and the following equations are expressed in terms of $(\theta_2-\theta_1)$ and $(\theta_2+\theta_1)$ to simplify their interpretation and application in the simulation procedure. Thus for a "broadside" source, located along the y-axis, $\theta_2+\theta_1=180$ degrees, for a source located on the "unit-circle" at R=D, $\theta_2-\theta_1=90$ degrees).

We now assume, that the source angles, detected at sensor position I and II are "noisy" random variables θ_I and θ_{II} with means θ_1 and θ_2 (i.e. true location), variance σ^2 (same for both sensors and independent of direction and correlation coefficient ρ and negligible higher moments.

The source coordinates (x,y), calculated from these detected angles by (3) and (4), will then also be random variables, the moments of which can be determined in terms of the source-angle moments by expanding (x,y) into a Taylor series about the true source location (x_s, y_s) (Ref. 2).

Neglecting all terms involving moments higher than σ^2 and ρ, mean and variance of x are given by :

$$\bar{x} = x_s + \frac{2D\sigma^2 \sin(\theta_2+\theta_1)}{\sin^3(\theta_2-\theta_1)} \left[\cos^2(\theta_2-\theta_1) - \rho\right] \tag{7}$$

$$\sigma_x^2 = \frac{D^2\sigma^2}{\sin^4(\theta_2-\theta_1)} \left[1 - \cos 2(\theta_2+\theta_1)\cos 2(\theta_2-\theta_1) + \rho\left[\cos 2(\theta_2+\theta_1) - \cos 2(\theta_2-\theta_1)\right]\right] \tag{8}$$

Similarly, mean and variance of y are :

$$\bar{y} = y_s + \frac{2D\sigma^2 \cos(\theta_2-\theta_1)}{\sin^3(\theta_2-\theta_1)} \left[1 - \cos(\theta_2+\theta_1)\cos(\theta_2-\theta_1) + \rho\left[\frac{\cos(\theta_2+\theta_1)}{\cos(\theta_2-\theta_1)} - 1\right]\right] \tag{9}$$

$$\sigma_y^2 = \frac{2D^2\sigma^2}{\sin^4(\theta_2-\theta_1)}\left[\left[1-\cos(\theta_2+\theta_1)\cos(\theta_2-\theta_1)\right]^2 + \sin^2(\theta_2+\theta_1)\sin^2(\theta_2-\theta_1) - \rho\left[\cos(\theta_2-\theta_1)-\cos(\theta_2+\theta_1)\right]^2\right] \quad (10)$$

The covariance of x and y is (using (3) and (4)):

$$C_{xy} = \frac{2\sigma^2D^2\sin(\theta_2+\theta_1)}{\sin^4(\theta_2-\theta_1)}\left[\cos(\theta_2-\theta_1)-\cos(\theta_2+\theta_1)\cos 2(\theta_2-\theta_1) + \rho\left[\cos(\theta_2+\theta_1)-\cos(\theta_2-\theta_1)\right]\right] \quad (11)$$

For convenience, we consider the calculated source coordinates (x,y) as rectangular components of a two-dimensional source location estimator (i.e. a two-dimensional random vector) with mean $(\bar{x},\bar{y})$ and variance S. From (7) and (9) we get the magnitude B of the estimator

bias (i.e., the Euclidean distance between the estimator mean $(\bar{x},\bar{y})$ and the true source location (x_s,y_s)):

$$B = \left[(\bar{x}-x_s)^2 + (\bar{y}-y_s)^2\right]^{1/2} = A\left[E + \rho F + \rho^2 G\right]^{1/2} \qquad (12a)$$

where

$$A = \frac{2D\sigma^2}{\sin^3(\theta_2-\theta_1)}$$

$$E = \cos^2(\theta_2-\theta_1)\left[1-2\cos(\theta_2+\theta_1)\cos(\theta_2-\theta_1)+\cos^2(\theta_2-\theta_1)\right]$$

$$F = 2\cos(\theta_2-\theta_1)\left[\cos(\theta_2+\theta_1)\left[1+\cos^2(\theta_2-\theta_1)\right]-2\cos(\theta_2-\theta_1)\right]$$

$$G = 1-2\cos(\theta_2+\theta_1)\cos(\theta_2-\theta_1)+\cos^2(\theta_2-\theta_1)$$

The angle δ of the estimator bias is, from (7) and (9):

$$\delta = \tan^{-1}\left[\frac{\bar{y}-y_s}{\bar{x}-x_s}\right]$$

$$= \tan^{-1}\left[\frac{(1-\rho)\cos(\theta_2-\theta_1)}{\sin(\theta_2+\theta_1)\left[\cos^2(\theta_2-\theta_1)-\rho\right]} - \cot(\theta_2+\theta_1)\right] \qquad (12b)$$

Similarly from (8) and (10) the estimator variance S (i.e., the mean-squared Euclidean distance between the

estimator (x,y) and the estimator mean ($\bar{x}$,$\bar{y}$)) is obtained as:

$$S = \sigma_x^2 + \sigma_y^2 = H\left[I + \rho J\right] \tag{13}$$

where H, I, J are functions of σ^2, D, θ_1, θ_2.

As seen from (12a) and (13) the square of the estimator bias is a quadratic function of the spatial data correlation coefficient; the estimator variance is a linear function of the spatial data correlation coefficient.

We can get an impression of this strong dependence on ρ from Figure 2 for a particular source-sensor geometry.

If the spatial coherence length of the medium is much larger or much smaller than the sensor separation, we can assume that the data are uncorrelated or fully correlated, currents in the medium or random motion of the (rigid) two-sensor detecting device relative to the propagation medium can result in correlation coefficients of $\rho \simeq -1$ or $\rho \simeq +1$.

The investigation of estimator bias and variance dependencies on ρ and on the source-sensor geometry leads to the graphs in Figures 3-7. We make use of the inverse relations

$$\theta_2 - \theta_1 = \tan^{-1}\left[\frac{\frac{2D}{R}\sin\theta}{1 - \frac{D^2}{R^2}}\right] \tag{14}$$

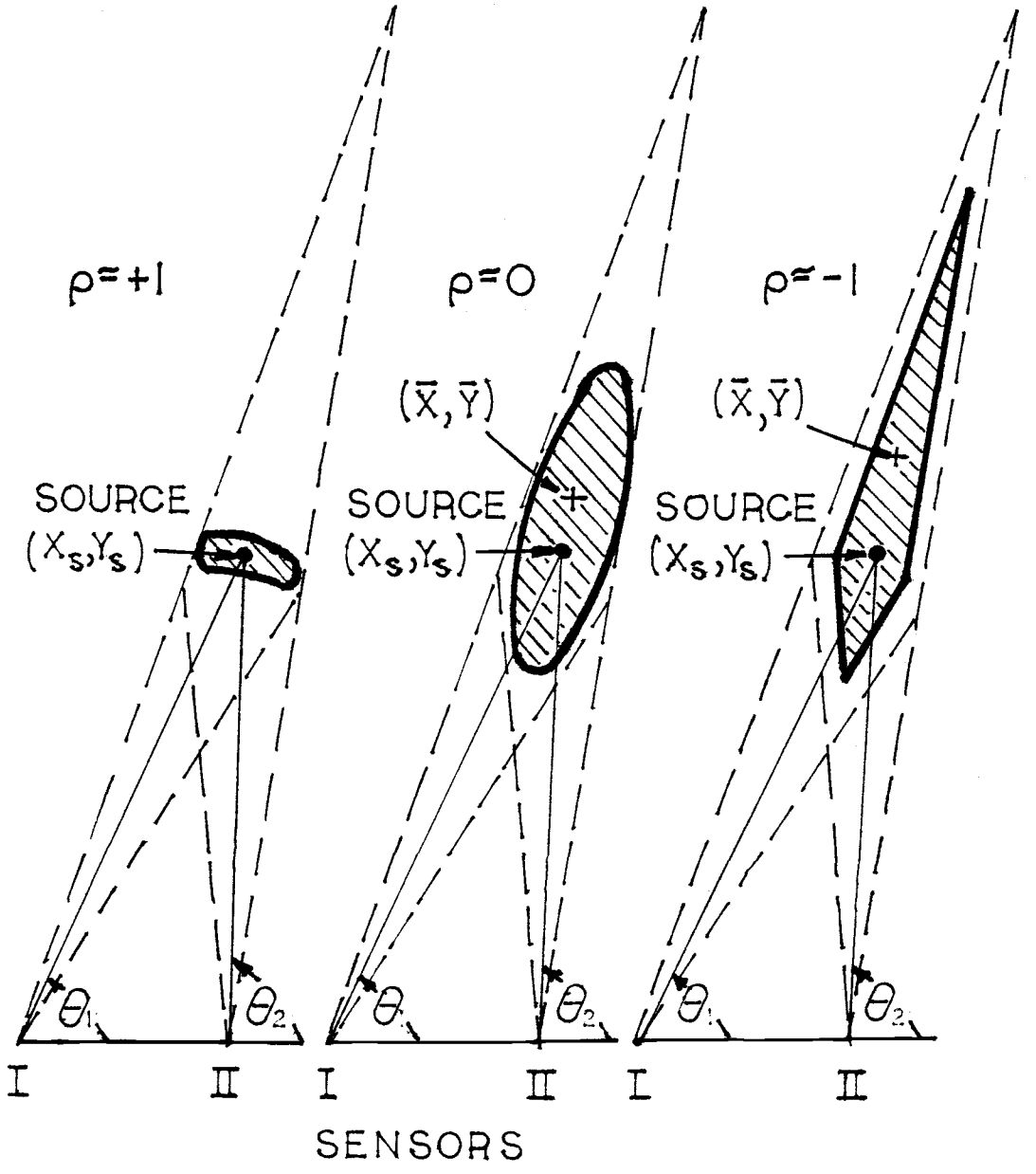

Figure 2: Source-location distributions for a) positively correlated data ($\rho \simeq +1$), b) uncorrelated data ($\rho \simeq 0$), and c) negatively correlated data ($\rho \simeq -1$).

$$\theta_2 + \theta_1 = \tan^{-1}\left[\frac{\sin 2\theta}{\cos 2\theta - \frac{D^2}{R^2}}\right] \qquad (15)$$

to derive (16) - (19).

We first consider the estimator bias B. Differentiation of (12a) with respect to ρ reveals that B is a minimum when:

$$\rho = \rho_{min} = -\frac{F}{2G}$$

$$\rho_{min} = \left[1 - \frac{D^2}{R^2}\right]\left[1 - \frac{D^2}{R^2}\cos 2\theta\right]\left[1 + \frac{D^4}{R^4} - 2\frac{D^2}{R^2}\cos 2\theta\right]^{-1} \qquad (16)$$

In Figure 3 ρ_{min} is plotted against the normalized source coordinates $(x_s/D, y_s/D)$, where $(R/D)^2 = (x_s/D)^2 + (y_s/D)^2$ from Figure 1. Note that $\rho_{min} > 0$ for all source locations except those in the region $\cos 2\theta < R^2/D^2 < 1$.

We substitute (16) into (12a) and obtain the minimum estimator bias:

$$\frac{B_{min}}{B_0} = \sin 2\theta \left[1 + \frac{R^4}{D^4} - 2\frac{R^2}{D^2}\cos 2\theta\right]^{-1/2} \qquad (17)$$

where $B_0 = A[E]^{1/2}$ is the estimator bias obtained for uncorrelated data ($\rho = 0$). In Figure 4 B_{min}/B_0 is plotted

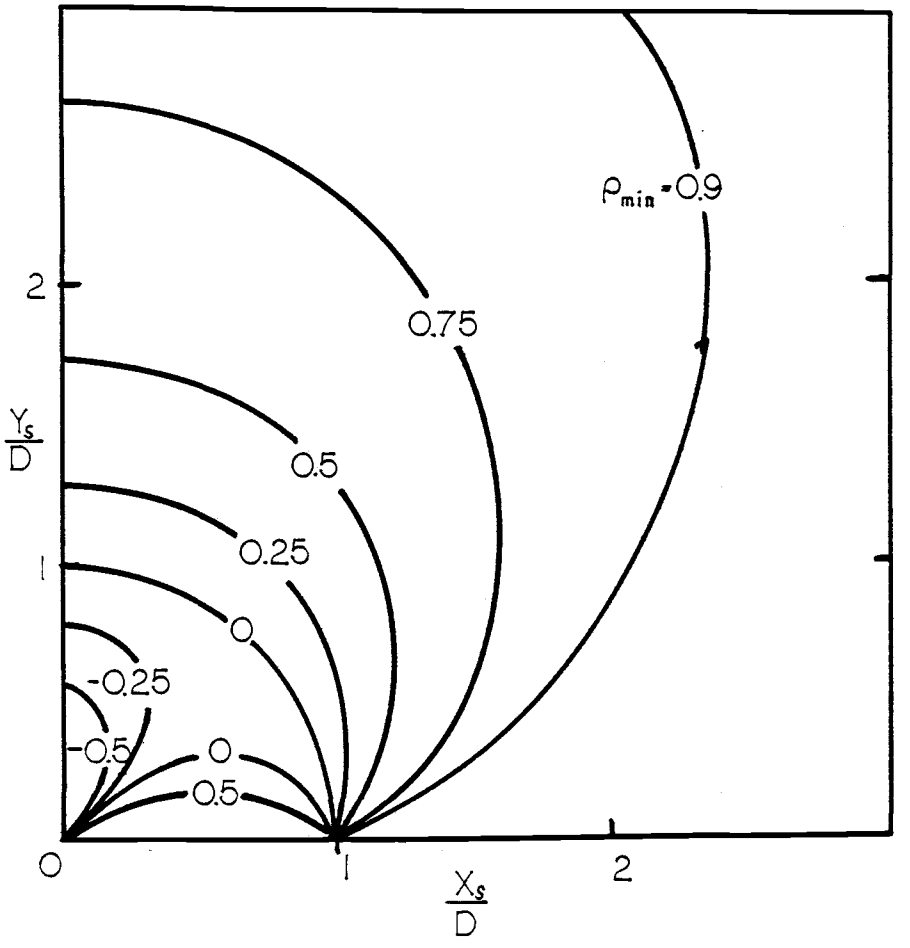

Figure 3: Data correlation coefficient (ρ_{min}) for minimum bias versus source location

against $(x_s/D, y_s/D)$. Note that on the 'Unit circle' (for R=D) B_{min}/B_0 is not defined.

The maximum estimator bias B_{max} is obtained by evaluating (12a) for $\rho=+1$ and $\rho=-1$ according as to whether $\rho_{min}<0$ or $\rho_{min}>0$, respectively.

$$\frac{B_{max}}{B_0} = \left[2(1-\cos 2\theta)\right]^{1/2}\left[1-\frac{R^2}{D^2}\right]^{-1} \qquad \cos 2\theta < R^2/D^2 < 1$$

$$\frac{B_{max}}{B_0} = \left\{2+\frac{2+2\frac{D^2}{R^2}\left[\frac{D^2}{R^2}-2\right]\cos 2\theta}{\left[1-\frac{D^2}{R^2}\right]^2}\right\}^{1/2} \qquad \text{elsewhere} \qquad (18)$$

The corresponding graphs are plotted in Figure 5.

The evaluation of (13) shows that the estimator variance S is a linearly decreasing function of ρ outside the 'Unit circle' ($R^2/D^2>1$) and a linearly increasing function of ρ inside ($R^2/D^2<1$).

For $\rho=-1$ and $\rho=+1$ we obtain the maximum and minimum variance, S_{max} and S_{min}.

$$\frac{S_{max}}{S_0} = \frac{2}{1+\frac{D^2}{R^2}} \quad ; \quad \frac{S_{min}}{S_0} = \frac{2}{1+\frac{R^2}{D^2}} \quad ; \text{ outside} \qquad (19)$$

$$\frac{S_{max}}{S_o} = \frac{2}{1+\frac{R^2}{D^2}} \quad ; \quad \frac{S_{min}}{S_o} = \frac{2}{1+\frac{D^2}{R^2}} \quad ; \text{ inside}$$

In Figures 6 and 7 S_{min}/S_0 and S_{max}/S_0 are plotted against $(x_s/D, y_s/D)$, which are concentric circles around the origin.

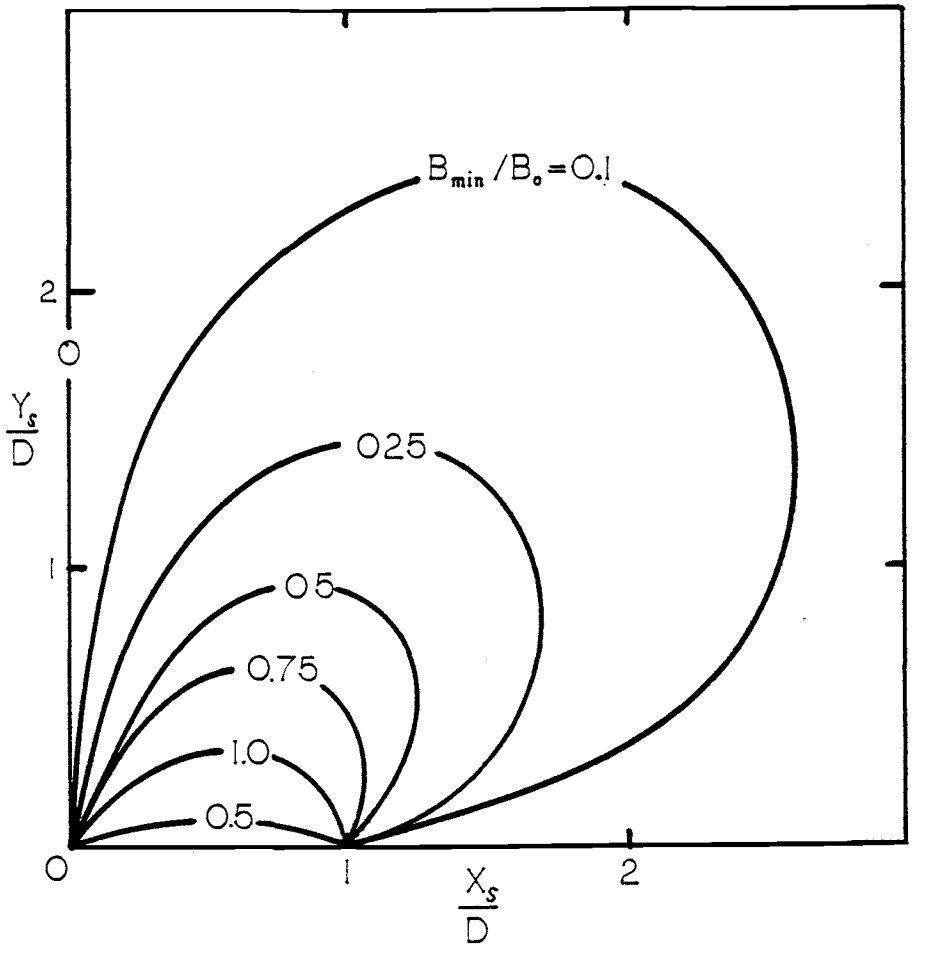

Figure 4: Normalized minimum bias (B_{min}/B_0) versus source location

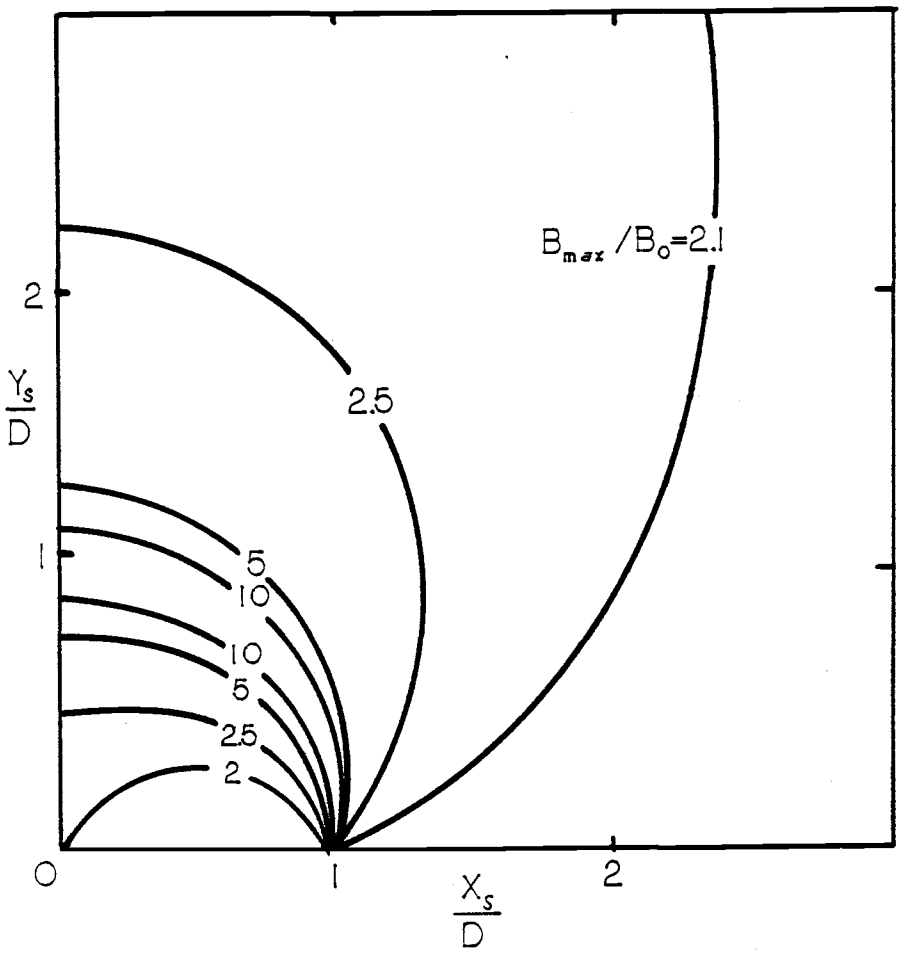

Figure 5: Normalized maximum bias (B_{max}/B_o) versus source location

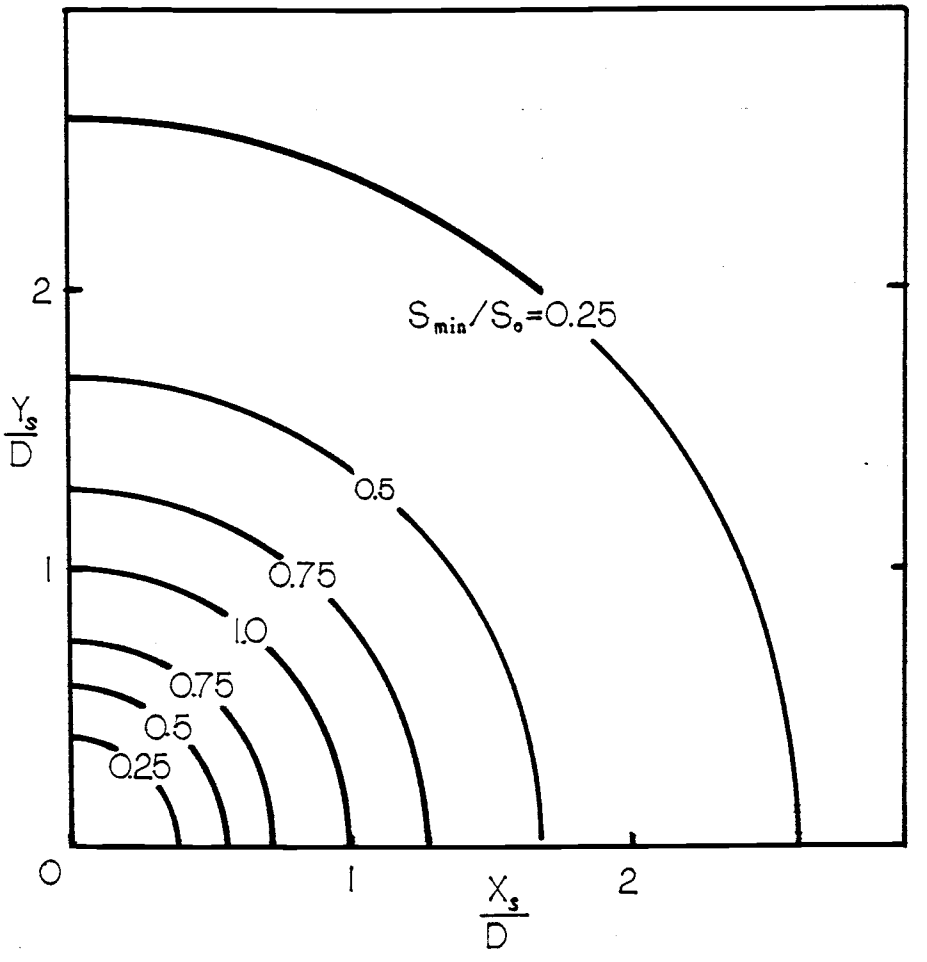

Figure 6: Normalized minimum variance (S_{min}/S_0) versus source location

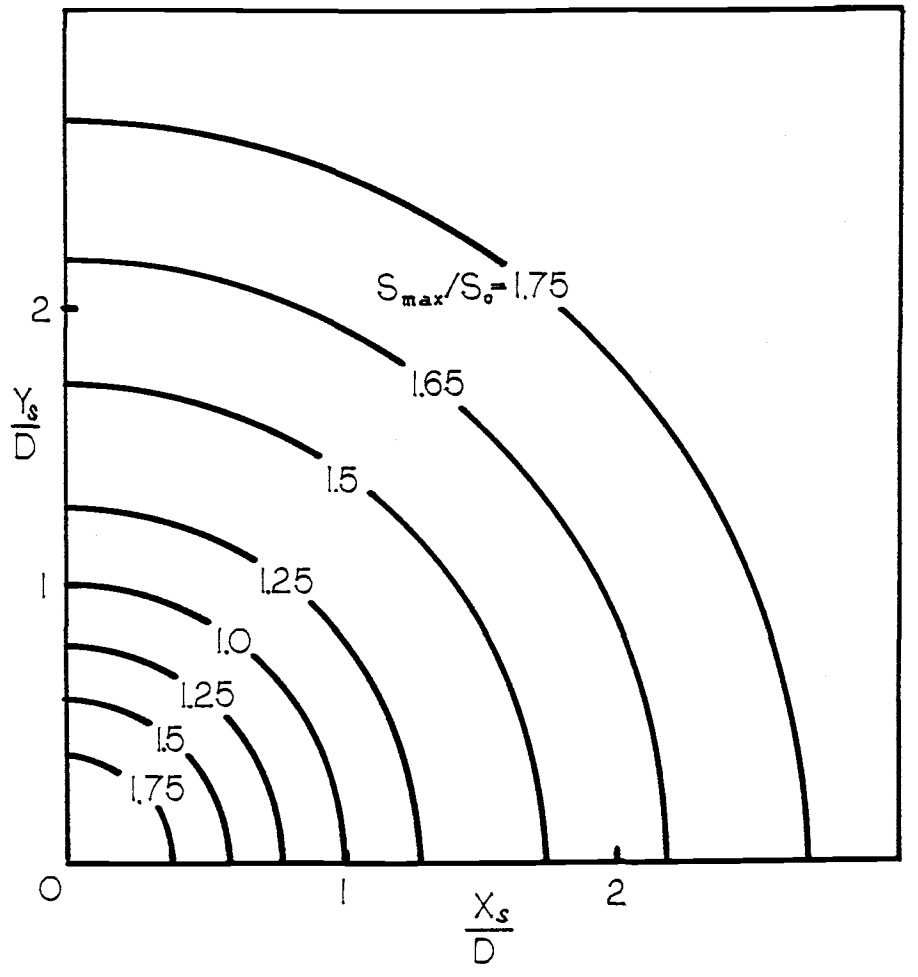

Figure 7: Normalized maximum variance (S_{max}/S_0) versus source location

III. STATIONARY TARGET: SIMULATION MODEL

As mentioned in the Introduction, the angle-of-arrival data are perturbed by various sources. We assume that the perturbation is caused by zero-mean additive noise which is superimposed on the correct angle-of-arrival data.

In our simulation the noise consists of zero-mean gaussian random variables $\Delta\theta_I$ and $\Delta\theta_{II}$, with joint probability density function given by:

$$f_{\Delta\theta_I,\Delta\theta_{II}}(\Delta\theta_I,\Delta\theta_{II}) = \frac{1}{2\pi\sigma^2\sqrt{1-\rho^2}}\exp\left[\frac{1}{2(1-\rho^2)}\left[\frac{\Delta\theta_I^2}{\sigma^2} - \frac{2\rho\Delta\theta_I\Delta\theta_{II}}{\sigma^2} + \frac{\Delta\theta_{II}^2}{\sigma^2}\right]\right] \quad (20)$$

where σ^2 = variances of θ_I, and θ_{II}, and the correlation coefficient is $(-1<\rho<+1)$.

RANDOM POSITION GENERATION

To create the angular noise $(\Delta\theta_I, \Delta\theta_{II})$ we use uncorrelated single precision random numbers (Z_1, Z_2) in the range of -5 to +5 of gaussian distribution with standard deviation $\sigma = 1$ and zero-mean provided by a random generator. Appendix B shows the algorithm for generating N

correlated gaussian number pairs. The desired random position (x,y) is calculated from the random angles θ_I and θ_{II} applying (3) and (4).

SAMPLE TEST

In order to get an impression of the distribution of the random positions around the true source location we calculate the statistical moments.

The expected value in x-direction is estimated by:

$$\bar{X} = E[x] = \frac{1}{N}\sum_{i=1}^{N} X_i \tag{21}$$

The unbiased estimate of the standard deviation in x-direction is given by:

$$S_x = \sqrt{\frac{1}{N-1}\left[\sum_{i=1}^{N} x_i^2 - N E^2[x]\right]} \tag{22}$$

Similarly we get expected value and standard deviation in y-direction.

The correlation of x- and y-positions is estimated by the unbiased correlation coefficient :

$$\rho = \frac{\frac{1}{N-1}\left[\sum_{i=1}^{N} x_i y_i - N E[x]\, E[y]\right]}{S_x\, S_y} \tag{23}$$

The magnitude and direction of the random positions overall displacement from the true source position is given by the empiric estimator bias and estimator angle.

Similar to (12a) and (12b) we get:

$$B = \left[(\bar{x} - x_s)^2 + (\bar{y} - y_s)^2 \right]^{1/2} \qquad (24a)$$

$$\delta = \tan^{-1}\left[\frac{\bar{y} - y_s}{\bar{x} - x_s} \right] \qquad (24b)$$

TEST PROCEDURES

Random Generator

The HP 1000 library software provides a random generator that creates a sequence of gaussian distributed random numbers.

The generator has to be initialized by a positive integer number, called generator seed, that characterizes the created sequence of numbers. This enables us to repeat the simulation under the same (initial) conditons, if wanted. The created sequence supposedly has zero-mean and

unit standard deviation.

By experience we found, that the entire simuation strongly depends on the choice of the generator seeds. In order to keep the computation time within reasonable ranges we have to limit the number of samples. (In the stationary case we usually ran the simulation with 50 samples.) Considering a practical implementation we have to deal with a limited number of samples in a restricted period of time as well. This however, implies that, owing to the imperfections in the random number generation, the random numbers and thus the whole simulation can be biased, the condition of zero-mean noise with unit standard deviation is no longer met.

The following test procedure was applied:

- Creation of random numbers by various calls to a random generator

- Calculation of the moments about the origin and moments about the mean.

Runs for different seeds confirmed that the deviation of mean and standard deviation from the given random generator specifications does not depend on the number of samples but on the choice of the generator seed.

In order to get true zero-mean, unit standard

deviation and thus independence from the generator seed, we apply a linear transformation to the random number sequence:

$$Z_i = (G_i - \bar{G}) / s_G \qquad (25)$$

where G_i is the original sample

$\bar{G}$ is the empiric sample mean

s_G is the unbiased sample standard deviation

thus it follows

Z_i is a random number with zero-mean and unit standard deviation.

Simulation of the Stationary Target

In this part we examined the inpact of angular noise power on the results. We tested how large the angular noise variance can be, to maintain reasonable agreement between theory and simulation. In order to get independence from the source range R we define the standard deviation of the noise in terms of the true source angles at detectors I and II:

$$\sigma = A \left[\theta_2 - \theta_1\right] \qquad (26)$$

where A is a normalizing constant used as simulation parameter.

a) Comparison of Theoretical and Simulated Standard Deviation

We calculated the ratio between theoretical and simulated standard deviation (TSDX) and (SDX), respectively, depending on true source location, spatial noise correlation and variance.

Provided that the ratio TSDX/SDX is gaussian distributed, we expect a mean value of:

$$E[TSDX/SDX] = 1 \tag{27}$$

and a standard deviation:

$$\text{Std.dev.}[TSDX/SDX] = \sqrt{2/N} \tag{28}$$

The estimation in y-direction follows respectively.

A representative graph is shown in Figure 8. It shows the simulation results for a source located at the normalized distance R=2 and angle θ=50 degrees for a normalized noise standard deviation A in the range: $0.01<A<0.81$ which for this particular configuration

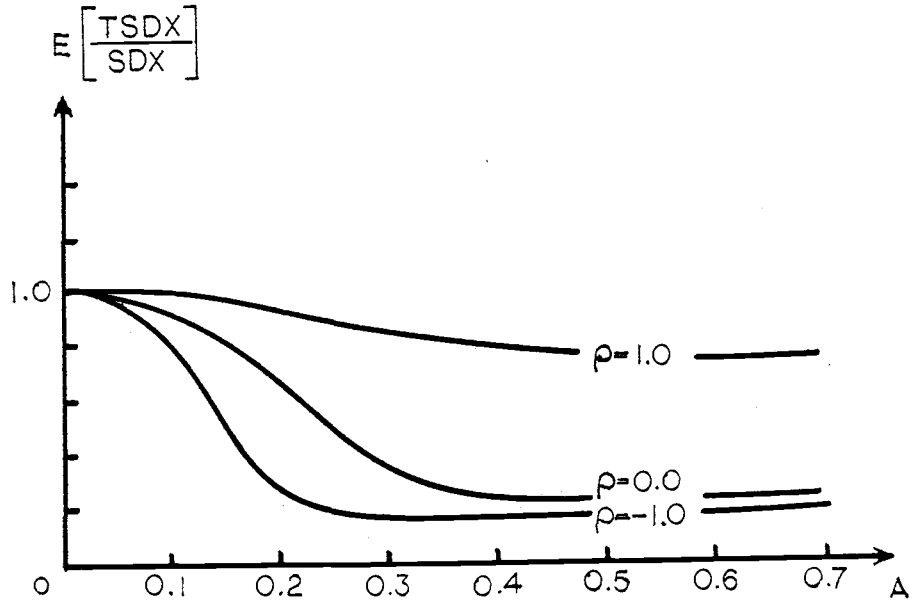

Figure 8: Ratio between theoretical and simulated standard deviation versus normalized spatial noise correlation

corresponds to an average angular deviation from 0.76 degrees to 77.1 degrees and for spatial noise correlation of $\rho=+1$, $\rho=-1$, and $\rho=0$. Note that for decreasing spatial noise correlation the theoretical standard deviation gets too "optimistic".

Tests for various source locations showed that for source locations where R>1 the theoretical estimations are too optimistic for a normalized noise standard deviation A>0.15. For locations within the unit circle (R<1) where the angle difference $(\theta_2 - \theta_1)$ is greater than 90 degrees, there is a significant deterioration of the estimation for A>0.05.

For source locations outside the unit circle we get a significant improvement of the estimation, if we cut off random positions located at larger distances from the sensors. Best results were achieved for cutting off 5% of the farest random positions.

b) Comparison of Theoretical and Simulated Estimator Bias

To get an impression for the deviation of the simulated average position from the theoretical average position, we tested the mean (absolute) deviation, M.D., with respect to the theoretical mean. (Ref. 3)

The mean absolute error (m.a.e) is defined to be:

$E[|XTA - Xa|]$

XTA = theoretical avg. pos.

XA = simulated avg. pos.

For gaussian mean distribution:

$$E[|XTA - Xa|] = SDX \cdot \sqrt{\frac{2}{\pi}} \cdot \frac{1}{\sqrt{N}} \qquad (29)$$

with a standard deviation of

$$\text{std. dev.}[|XTA - Xa|] = SDX \sqrt{1 - \frac{2}{\pi}} \cdot \frac{1}{\sqrt{N}} \qquad (30)$$

SDX = simulated std. dev.

N = # of random positions

(Y-direction follows respectively.)

The simulation conditions correspond to those in the previous subsection.

The representative plot in Figure 9 for a source position at the normalized distance R=2 and source angle θ=50 degrees, for a normalized noise standard deviation A in the range: 0.01<A<0.81 and for spatial noise correlation of ρ=+1, ρ=-1, and ρ=0 shows that the simulation confirms the assumption for the mean deviation for a limited range of angular noise power. Various tests showed that for

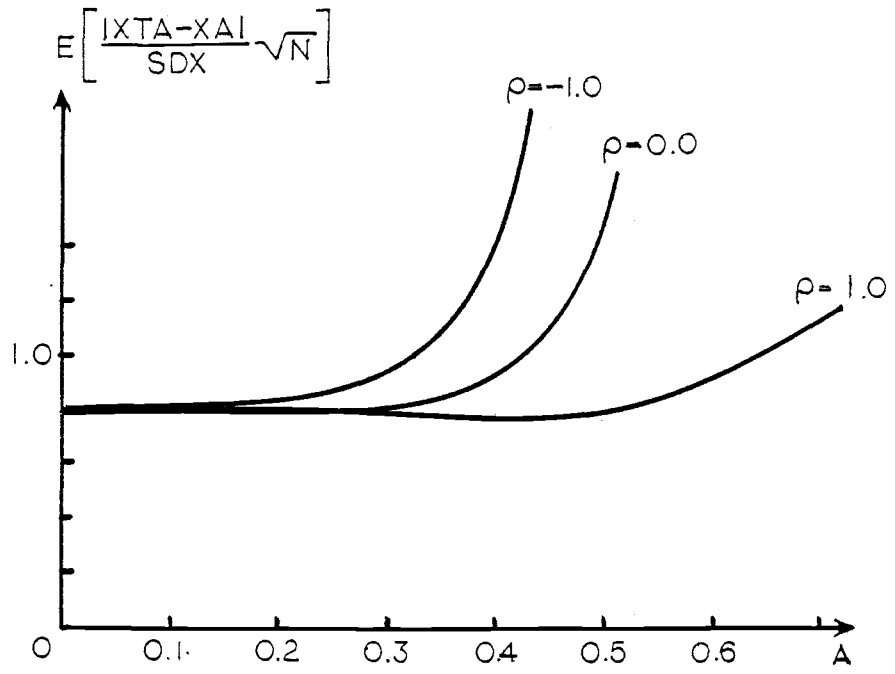

Figure 9: Normalized mean absolute error versus normalized spatial noise standard deviation

source locations outside the unit circle (R>1) the mean absolute error stays approximately constant at the theoretical value given by (29) up to a normalized noise standard deviation of A=0.3. For locations within the unit circle the range for A is slightly smaller, again due to larger difference angles.

Applying cuts at sample positions of large distance from the sensors did not result in a significant improvement of the bias estimation.

c) Summary of the Test Results

We tested the fit of theory and simulation for standard deviation in x-direction and y-direction and bias for different noise variances and spatial correlation at various source locations.

The overall range for $A = \sigma / (\theta_2 - \theta_1)$ to get reliable agreement is:

$A < 0.15$ for $R > 1.0$

$A < 0.05$ for $R < 1.0$

Note that because of the normalization (26) the maximum allowed noise standard deviation depends on the source location.

Bias/Variance Distribution in the X-Y Plane

The following test was used to confirm the theoretical distributions for estimator bias B and estimator variance S as given by (16) - (19).

We tested for:

- Correlation coefficient ϱ_{min} for minimum bias
- normalized minimum bias (B_{min}/B_0)
- normalized maximum bias (B_{max}/B_0)
- normalized minimum variance (S_{min}/S_0)
- normalized maximum variance (S_{max}/S_0)

For every true source location in a x-y plane-array, for fixed noise variance, and N random positions, we calculated

- estimator bias $B(\varrho)$ and variance $S(\varrho)$

  for : $-0.9 < \varrho < +0.9$; $A = 0.1$; $N = 40$

From these data we selected:

- minimum bias $B_{min}(\varrho) = B(\varrho_{min})$
- maximum bias $B_{max}(\varrho)$
- minimum variance $S_{min}(\varrho)$
- maximum variance $S_{max}(\varrho)$

- ϱ_{min} (follows directly from B_{min})

We then normalized these estimates with respect to bias and variance for the uncorrelated case ($B_0 = B(\varrho=0)$, $S_0 = S(\varrho=0)$) and displayed them in the x-y plane. The simulated distributions can be viewed in Figures 10 - 14.

The test showed that the theoretical distribution for estimator bias is very sensitive to biased noise that is received at the detectors. Biased (i.e., non-zero-mean noise) results in a spread of the estimator bias distribution in the x-y plane. The estimator variance distribution, however, stays relatively stable for biased noise.

For the region within the unit circle the simulatiuon could not show clear results due to a not sufficiently large resolution. We can, however assume that the simulation tends to agree with the theoretical distribution.

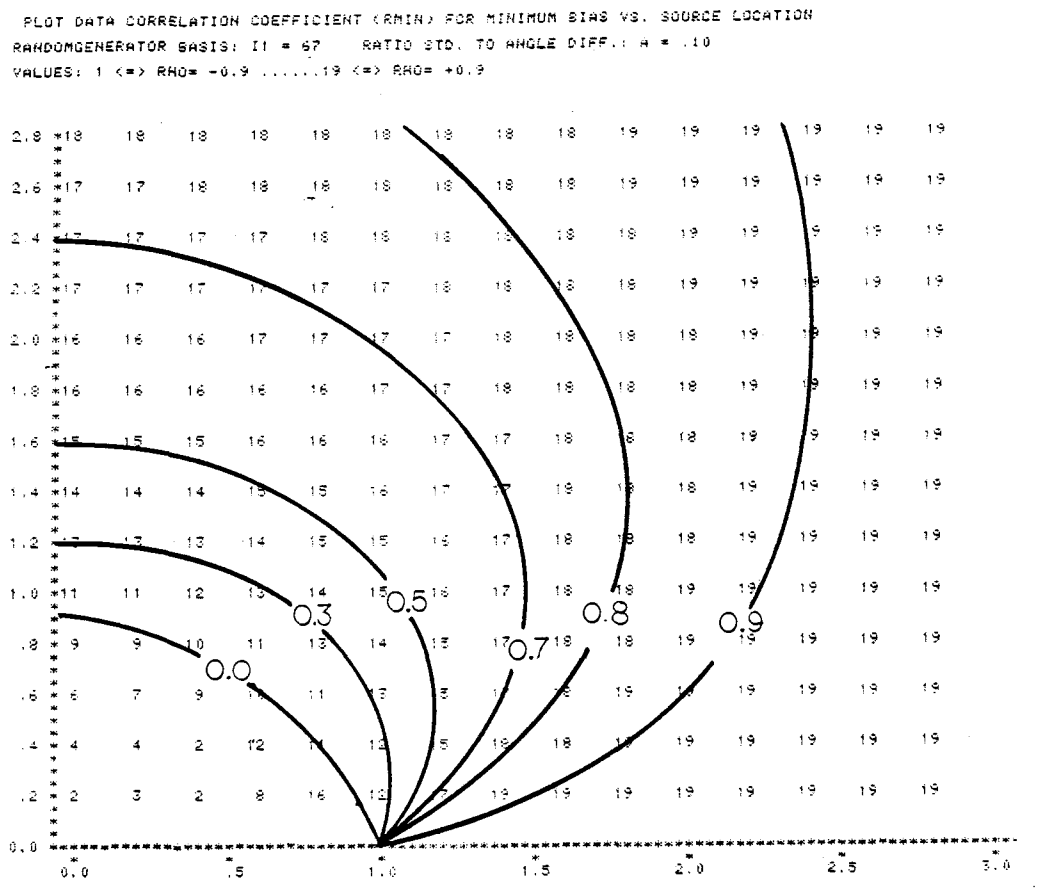

Figure 10: Simulated data correlation coefficient (ρ_{min}) for minimum bias versus source location

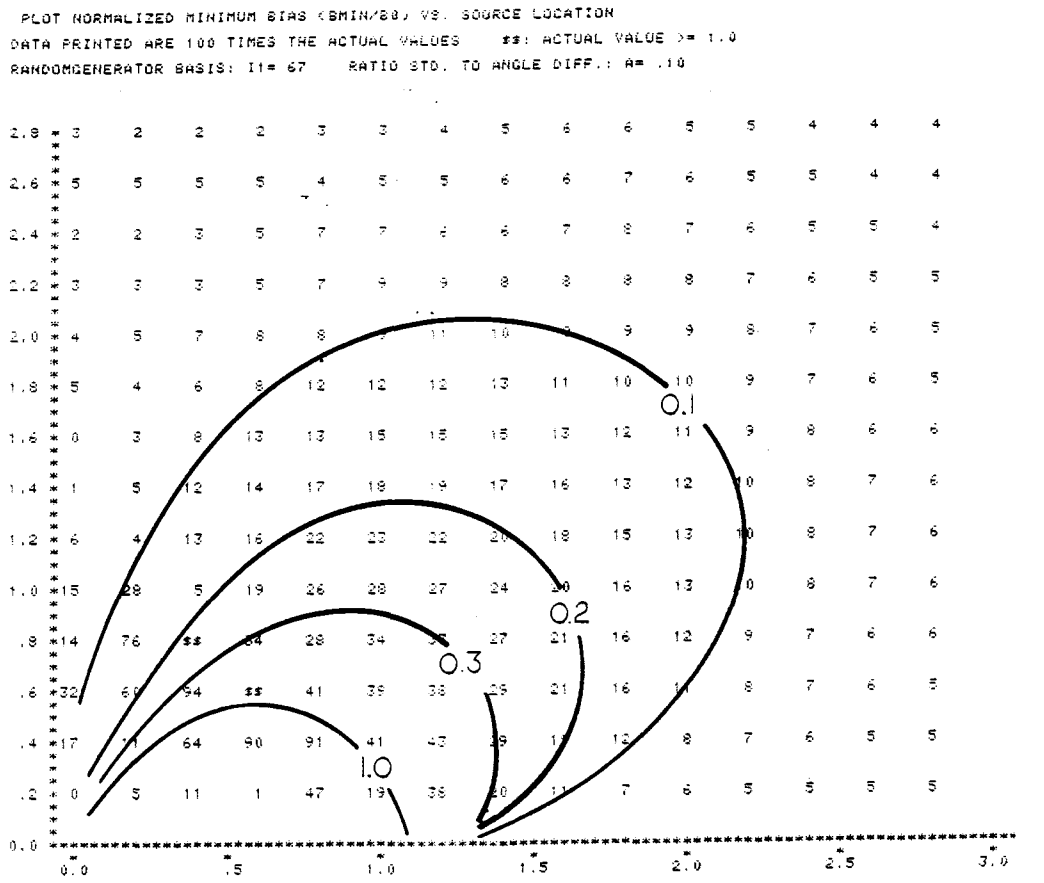

Figure 11: Simulated normalized minimum bias (B_{min}/B_0) versus source location

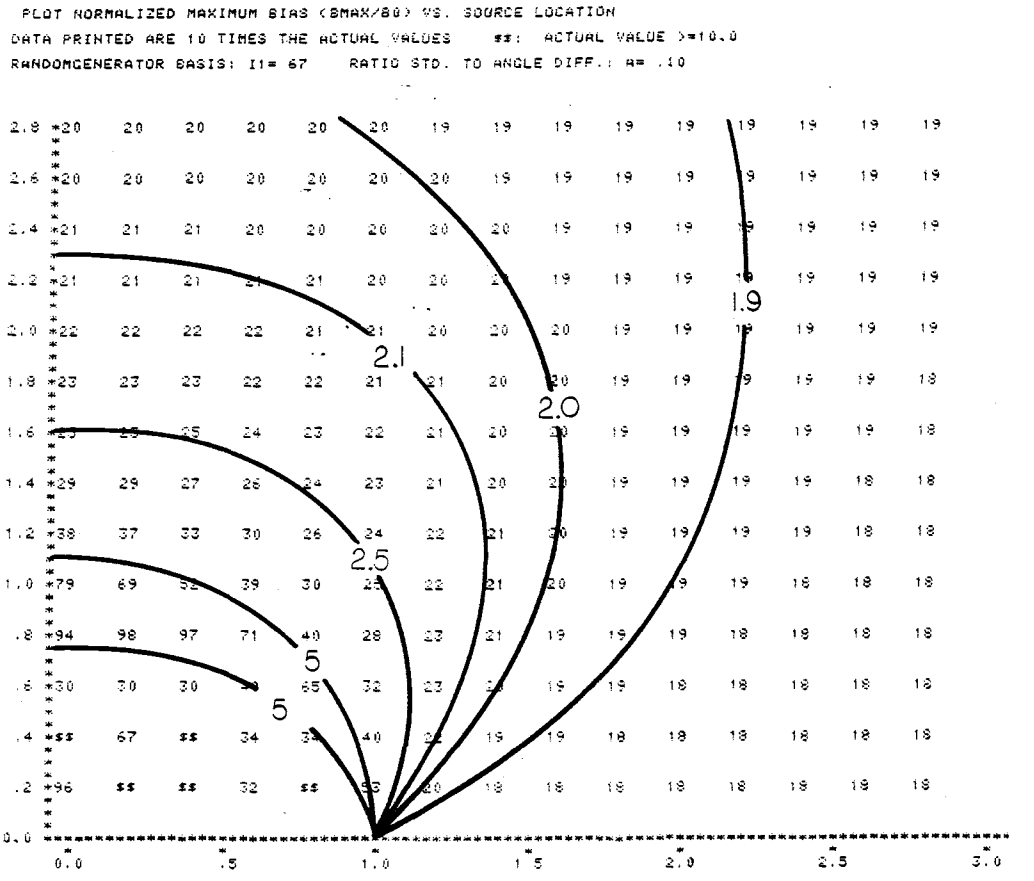

Figure 12: Simulated normalized maximum bias (B_{max}/B_0) versus source location

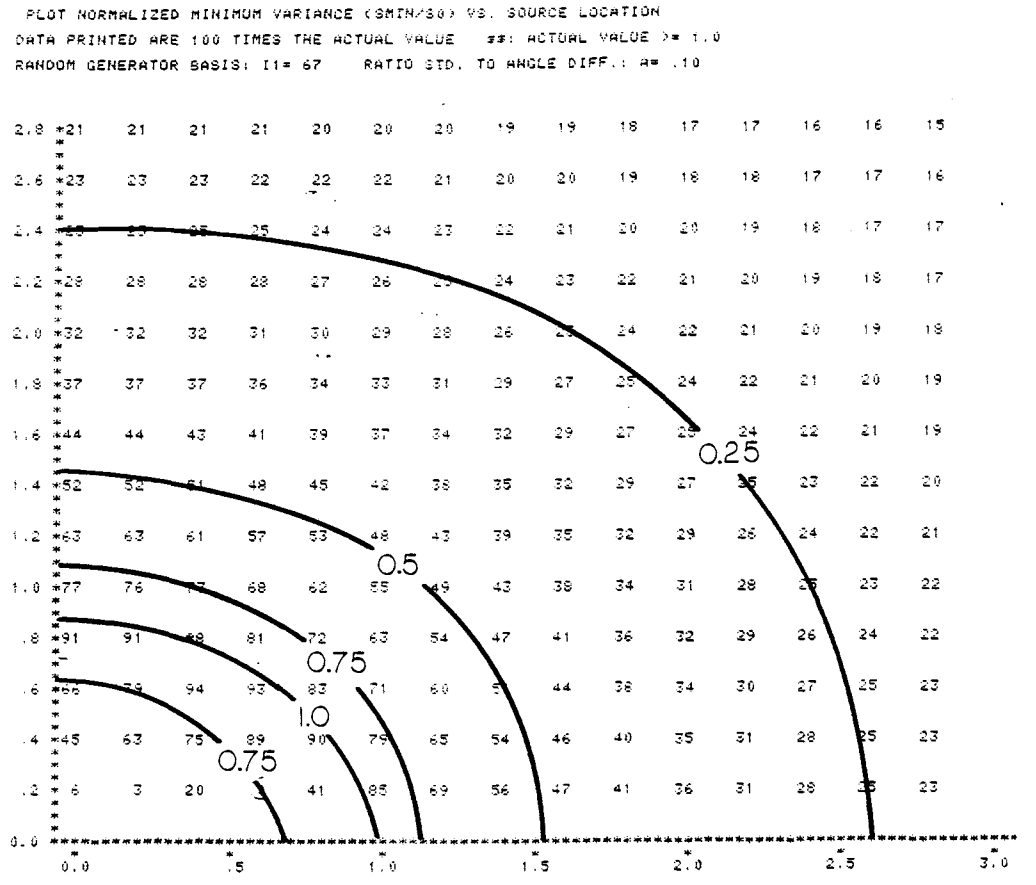

Figure 13: Simulated normalized minimum variance (S_{min}/S_0) versus source location

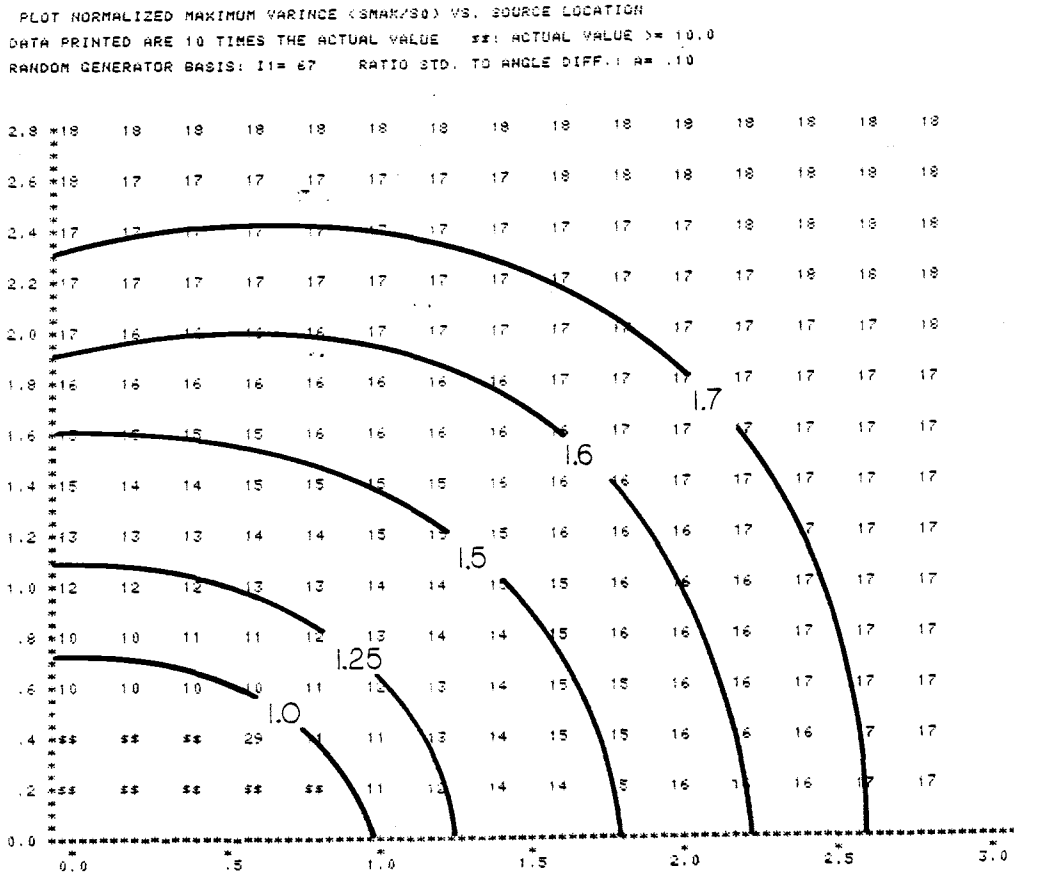

Figure 14: Simulated normalized maximum variance (S_{max}/S_0) versus source location

IV. MOVING TARGET: DATA SMOOTHING

In this section we descibe an optimum trajectory estimation (with minimum mean-squared error). A new flexible smoothing technique is introduced which consists of alternatively smoothing over a symmetric or asymmetric window (Ref. 4). In general the introduced smoothing technique is applied under the following operating conditions:

a) The tracking data are collected at two fixed sensors and at discrete, constant time intervals. They consist of the true source angles plus additive zero-mean gaussian angular white noise. The noise at both sensors has identical time independent variance with known x-y distribution but is possibly spatially correlated between the sensors.

b) For the duration of the smoothing interval the source being tracked travels at a constant (but unknown) speed along a trajectory having constant (but unknown) radius of curvature.

As we know, the angular detection of the source location introduces a bias depending on the spatial noise

correlation between the two sensors. Therefore the data must be corrected in order to get an unbiased trajectory estimate.

Our smoothing procedure consists of two steps:

1) The apparent source locations are calculated for each pair of measured angular data. For these locations the bias is removed, so that the data correspond to the true source locations plus zero-mean two-dimensional noise.

2) Smoothing is now performed over sequences of unbiased noisy data points. The sequences are of constant length, limited by how well condition b) (i.e. constant speed and radius of curvature) is met.

In practice, assuming that the target travels at approximately constant speed and along a circular trajectory, the algorithm performs asymmetric smoothing, thereby permanently calculating the distance travelled from the previously smoothed position to the most recently smoothed target position and comparing it with the average distance travelled between the already smoothed locations during the smoothing interval. This avoids out of range positions caused by the nature of the asymmetric part of the smoothing procedure in the vicinity of points of

inflection (i.e., infinite radius of curvature, for example where the target changes from a clockwise circular trajectory to a counterclockwise one) or in the case where the target travels along a straight line). In the case where the travelled distance exceeds a certain range we apply a conventional symmetric smoothing procedure. (Refer to "Smoothing Procedures".

BIAS REMOVAL

To remove the measurement bias caused by the angular detection, each apparent source location (x_i, y_i) calculated from the angular pair $(\theta_{1i}, \theta_{2i})$ by (3) and (4) must be corrected by subtracting the corresponding bias components calculated for the same data pair by (12a) and (12b). The unbiased source estimates are:

$$X_{oi} = X_i - B_i \cos \delta_i$$
$$Y_{oi} = Y_i - B_i \sin \delta_i \qquad (31a,b)$$

The resulting data point (X_{oi}, Y_{oi}) thus represents the true source location corresponding to the i'th angular data point, plus zero-mean two-dimensional noise.

In order to carry out the bias removal as given in

(31a,b), it should be noted that bias magnitude B and angle δ are functions of the angular noise variance σ^2 and the spatial correlation coefficient ρ. Owing to difficulties in estimating these quantities from the angular data, we have to assume these properties by experience. This can be done by estimating noise variance and spatial correlation for various stationary sources in the vicinity of the sensors thereby taking track of the physical quantities of the transmission medium like sound velocity and water temperature. The x-y distribution of these quantities have to be stored in order to apply them in the bias calculation. Otherwise we have to carry out the bias removal under the assumption that noise variance and spatial correlation are known and constant in the x-y plane.

SMOOTHING PROCEDURES

Conventional smoothing techniques consist of averaging data points in a symmetric "window" about the point of interest, according to some weighing law (e.g. rectangular window). Such symmetric smoothing, however, leads to a new bias due to trajectory curvature.

We make use of a new "asymmetric" smoothing procedure in which two subsets of the symmetric data are averaged

individually in order to obtain two regression lines. By properly proportioning the subsets about the data point of interest, the intersection of these regression lines is, to a very good approximation, an unbiased estimate of the true target location. The residual trajectory bias error for this procedure is about an order of magnitude smaller than the bias obtained with conventinal (symmetric) smoothing, provided that the total smoothed arc does not exceed a complete circle (i.e. 360 degrees). The optimum assymmetry ratio 1/u of the arc length about the point of interest is $\sqrt{3} + 2$, a deviation of about 15% from the optimum does not substantially increase the residual bias. (For the derivation of the asymmetry ratio refer to appendix A.)

To get more insight in the smoothing procdure, assume that the successive data points (X_{oi}, Y_{oi}), calculated from the angle-of-arrival data and corrected for bias, are arranged sequentially in the order in which they were generated in time. Around the point of interest (X_p, Y_p) we devide the calculated target locations into two sequences of data points each l positions long. Sequence 1 runs from i=p-l to i=p+u, and sequence 2 runs from i=p-u to i=p+l, where $l=(\sqrt{3} + 2)u$. For each sequence we then determine its regression line.

$$\text{LINE 1:} \quad (Y-Y_1) = m_1 (X-X_1) \tag{32}$$

where

$$\bar{X}_1 = \frac{1}{\ell+u}\sum_{i=p-\ell}^{p+u} X_{oi} \quad ; \quad \bar{Y}_1 = \frac{1}{\ell+u}\sum_{i=p-\ell}^{p+u} Y_{oi} \tag{33}$$

$$S_{x1}^2 = \frac{1}{\ell+u}\left[\sum_{i=p-\ell}^{p+u} X_{oi}^2\right] - \bar{X}_1^2 \tag{34}$$

$$S_{y1}^2 = \frac{1}{\ell+u}\left[\sum_{i=p-\ell}^{p+u} Y_{oi}^2\right] - \bar{Y}_1^2 \tag{35}$$

$$C_{xy1} = \frac{1}{\ell+u}\left[\sum_{i=p-\ell}^{p+u} X_{oi} Y_{oi}\right] - \bar{X}_1 \bar{Y}_1 \tag{36}$$

and

$$m_1 = \frac{S_{y1}^2 - S_{x1}^2}{2C_{xy1}}\left[1 \pm \left[1 + \frac{4C_{xy1}^2}{(S_{y1}^2 - S_{x1}^2)^2}\right]^{1/2}\right] \tag{37}$$

(The $\pm$ sign in m_1 is choosen such that $m_1/C_{xy1} > 0$.)

Similarly, for sequence 2 we have

$$\text{LINE 2:} \quad (Y-Y_2) = m_2(X-X_2) \tag{38}$$

where Y_2, X_2 and m_2 are calculated by equations identical to (33) - (37), except that the summations run from i=p-u

to i=p+1.

The intersection of LINE 1 and LINE 2 is the estimated source location corresponding to the p'th data point (X_{sp}, Y_{sp}).

Thus,

$$x_{sp} = \frac{(\bar{y}_2 - m_2 \bar{x}_2) - (\bar{y}_1 - m_1 \bar{x}_1)}{m_1 - m_2} \tag{39}$$

$$y_{sp} = \begin{cases} (y_1 - m_1 x_1) + m_1 x_{sp} \\ \text{or} \\ (y_2 - m_2 x_2) + m_2 x_{sp} \end{cases} \tag{40}$$

Schematically, the procedure is shown in Figure 15.

As mentioned above the flexible nature of the proposed smoothing technique takes effect in the vicinity of points of inflection where both regression lines are parallel or nearly so. Thus the intersection of LINE 1 and LINE 2 is far off the point of interest. In this case the conditions for smoothing over a symmetric window by averaging data points around the point of interest are met to get an unbiased estimation of the source location. The window size includes the same number of data points as used for the asymmetric smoothing. Thus for the sequence of data points ranging from i=p-1 to i=p+1 we get:

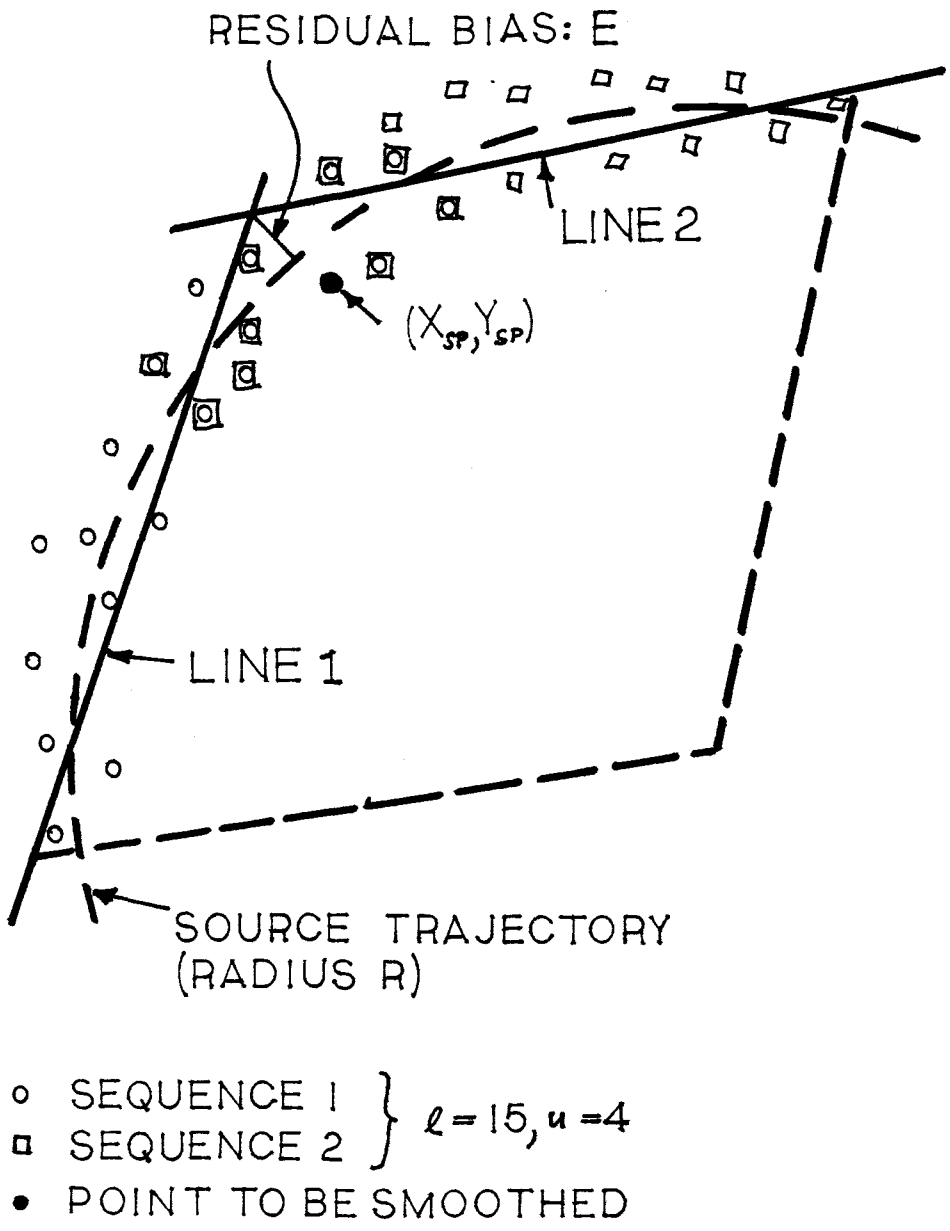

Figure 15: Asymmetric smoothing procedure

$$X_{sp} = \frac{1}{2e+1} \sum_{i=p-e}^{p+e} X_i \qquad (41)$$

$$Y_{sp} = \frac{1}{2e+1} \sum_{i=p-e}^{p+e} Y_i \qquad (42)$$

As an indicator of the point of inflection we control the speed of the target in the smoothing interval by calculating the average distance travelled. Assuming that the target keeps its speed approximately constant, by experience we choose the decision level RM for switching from asymmetric smoothing to symmetric smoothing to RM=2 (i.e., the just travelled distance obtained with asymmetric smoothing D_a exceeds twice the average distance D). Then, in order to avoid wrong decisions we compare both distances (i.e., the distance from the previous smoothed position to the positions, estimated with asymmetric, D_a, and symmetric smoothing D_s) with the average distance D and chose the estimated source location for which the travelled distance is closer to the average distance D. The flowgraph in Figure 16 summarizes the smoothing procedure realized in a computer simulation.

One can show (see Ref. 4) that for the assumed conditions for constant speed and trajectory curvature, the

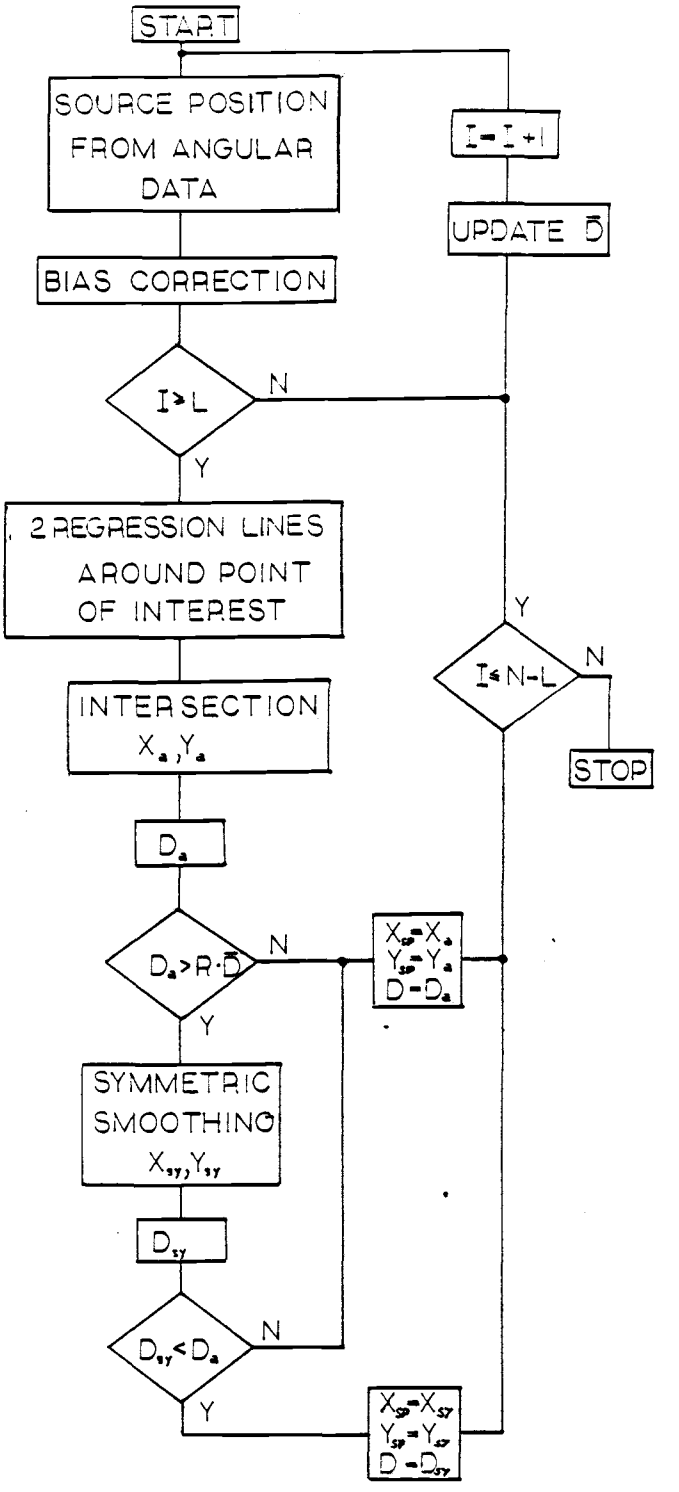

Figure 16: Flow graph: New smoothing algorithm

optimum smoothing arc is that for which the total mean square error

$$e = \frac{S}{2\ell} + \left(\frac{E}{R}\right)^2 \tag{43}$$

is minimum. (S is the variance of the data points from the true trajectory, as given by (13), E/R is the residual bias error.)

For a smoothing arc no longer than 180 degrees and optimum 1/u, differentiation of (43) leads to the condition

$$\ell_{opt} = 3.82\left[\frac{SR^6}{D^8}\right]^{1/9} \tag{44}$$

where D is the average distance travelled by the source in the time interval between successive data points and R is the radius of curvature. If D and R are at least approximately known (44) provides a good estimate of the optimum number of data points ($2l_{opt} + 1$) over which smoothing should be performed. If we do not have this information available or the target violates the condition of constant velocity or it does not follow a true circular trajectory several runs for increasing smoothing arc length

have to be performed and the estimated trajectories displayed in order to judge for the "best fit". Another way is by repeatedly smoothing over already smoothed positions in order to improve the trajectory estimate.

SIMULATION RESULTS

The smoothing technique was tested in several simulation runs for different trajectories. Figures 17, 18, and 19 show the simulation for a true circular trajectory (R=2.0, 2 circle segments of Al=80 degrees each, stepwidth DAL=2.0 degrees) in the case of positively correlated, uncorrelated, and negatively correlated data. It can be seen, that in general the estimated trajectory is a good approximation of the true trajectory. Figure 20 shows the recorded data of the simulation run in Figure 19. Numbers in row BML not equal to zero indicate where the algorithm switched from asymmetric smoothing to smoothing over a symmetric window in order to avoid positions out of range (which for instance would have occured at target position 16).

The simulation run also shows some inaccuracies (like a suddenly backwards travelling target) which could be excluded in a more sophisticated algorithm. A more detailed investigation of apparently "backward" travelling

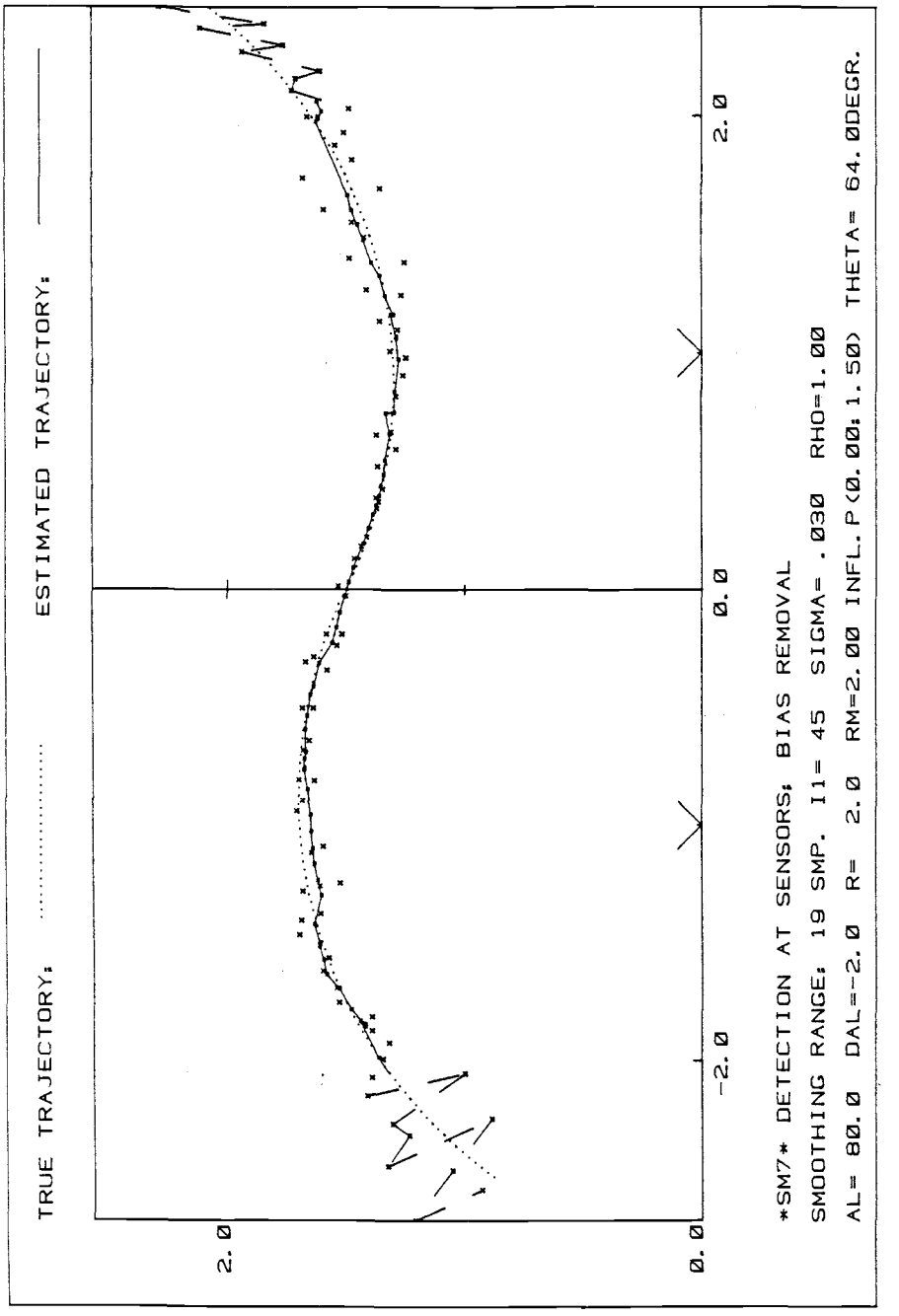

Figure 17: New smoothing algorithm: True circular trajectory: σ = 0.03, ρ = +1.0

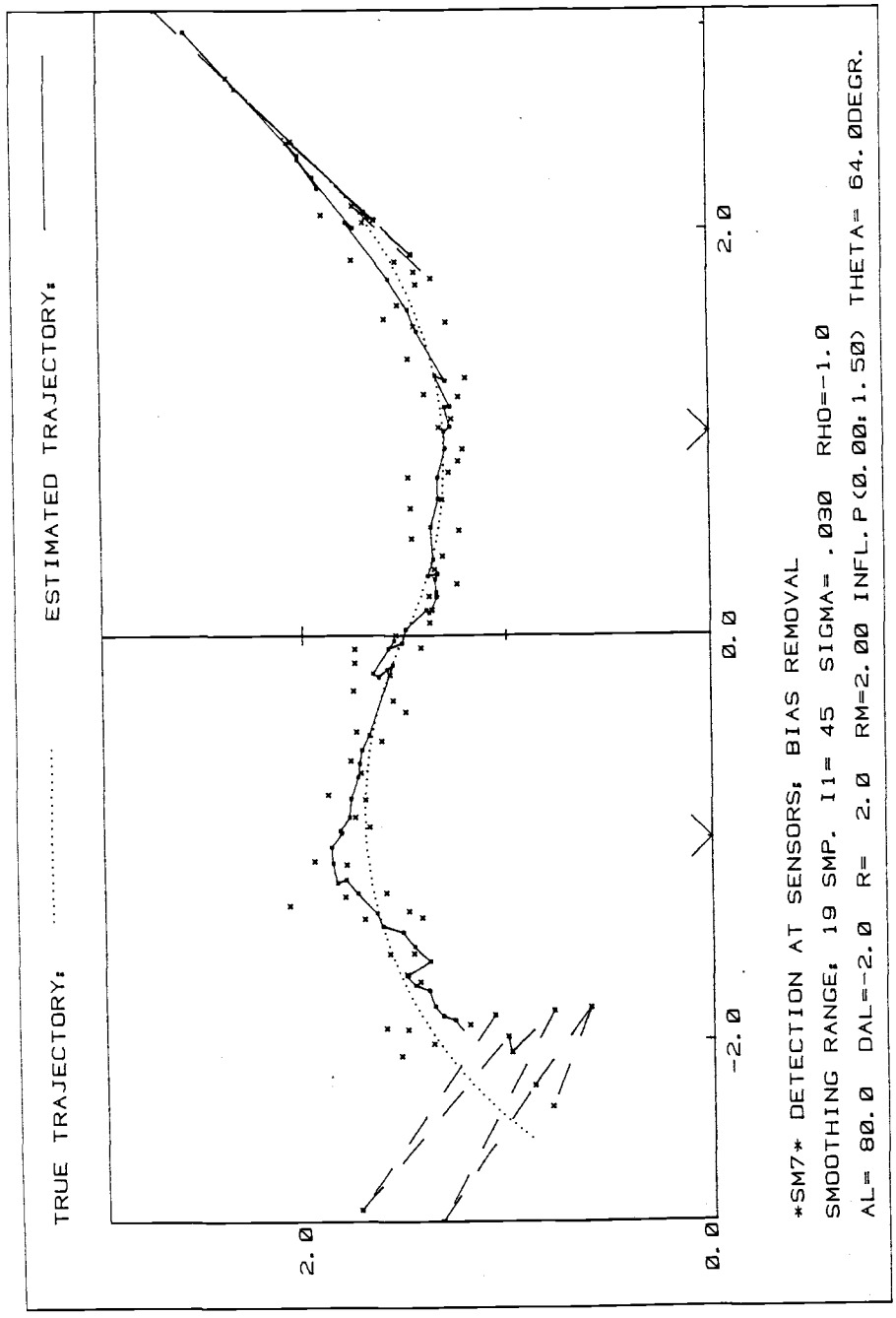

Figure 18: New smoothing algorithm: True circular trajectory: σ = 0.03, ρ = -1.0

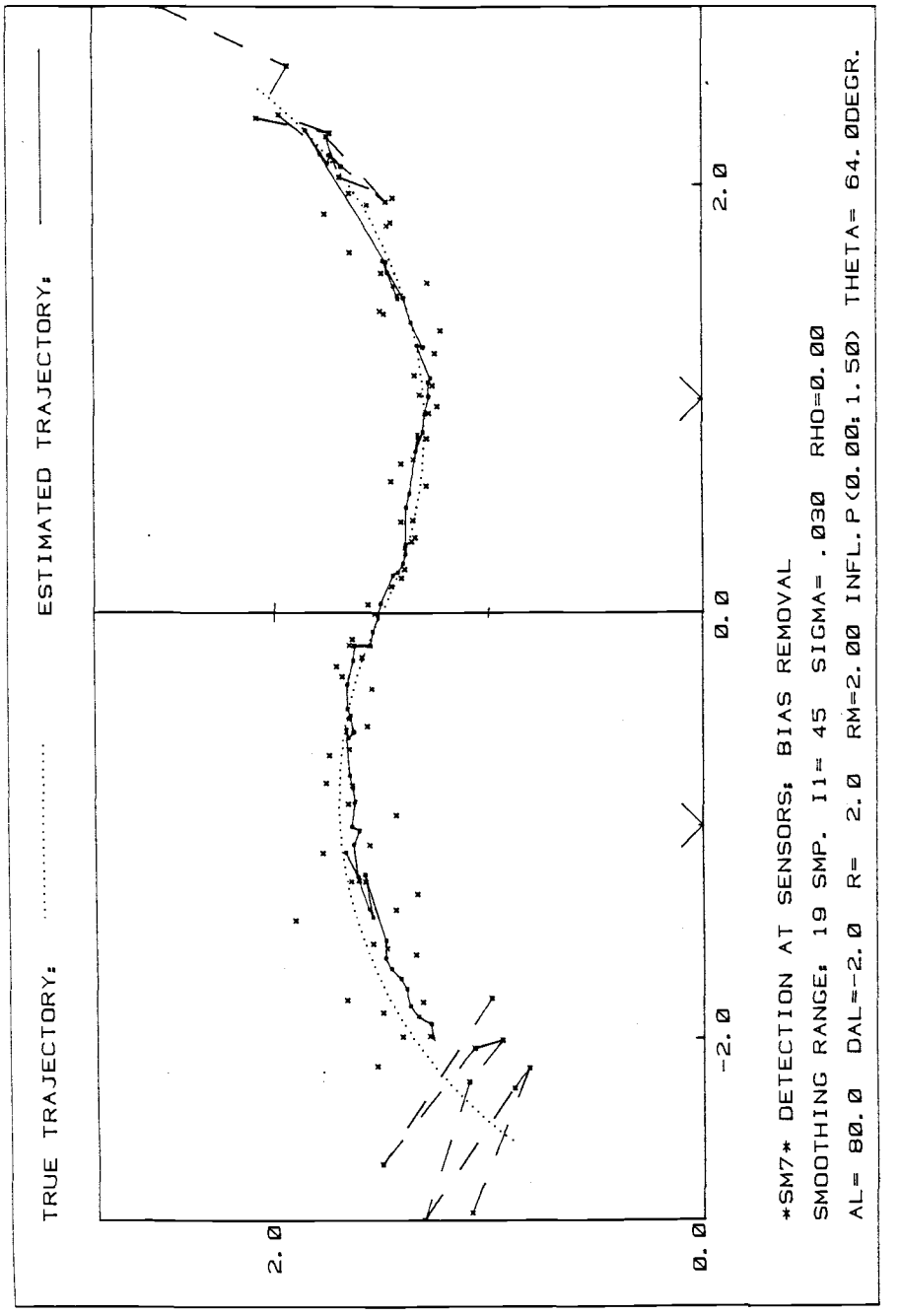

Figure 19: New smoothing algorithm: True circular trajectory: σ = 0.03, ρ = 0.0

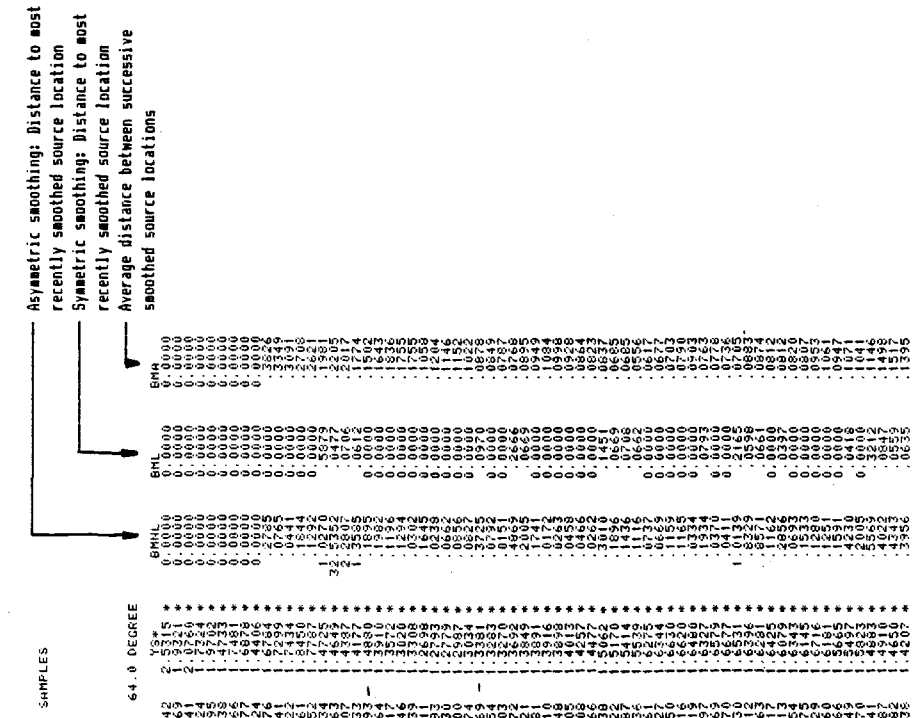

Figure 20: Simulation data listing for the trajectory in Figure 19.

targets (e.g. at target position 63) showed that these erroneous positions only appear when the algorithm switches from smoothing over a symmetric window to asymmetric smoothing and vice versa.

We further tested the adaptability of the algorithm concentrating on trajectories where the radius of curvature changes constantly. As a practical example we assume a sensor configuration where several sensors are arranged along a straight line, each sensor 200 m apart. The sensors active for source location are those for which the target distance is a minimum. Thus when a target leaves the area of one sensor pair, detection is handed over to the next sensor pair. Additionally, assume a target travelling at a velocity of 10 m/s along a fixed trajectory. (For the observer target velocity and trajectory are unknown.) The data are collected at a fixed rate and were smoothed using a constant smoothing interval.

For the simulation we asumed a trajectory given by

$$Y = -0.08X^3 + 0.04X^2 + 0.42X + 1.5 \qquad (45)$$

(all distances are normalized with respect to half the sensor separation.) Concentrating only on a two-sensor configuration, the area of interest was chosen slightly larger than required by a multisensor configuration. Also, the nature of the asymmetric smoothing technique requires

enough source positions ahead of the first smoothed position. Thus we chose the area of interest to $-2.0<X<+2.5$. The sampling rate was 2.5/sec. Figures 21, 22, and 23 show the simulated run for constant angular noise standard deviation and constant spatial correlation coefficient :

$\rho = +1.0$	$\sigma = 0.03$	Figure 21
$\rho = 0.0$	$\sigma = 0.03$	Figure 22
$\rho = -1.0$	$\sigma = 0.03$	Figure 23

(The dashed lines connected the unsmoothed source positions in the order they were detected in time.) Note, that the dashed lines actually indicate an unsmoothed trajectory, "seen" from the sensors. In all three cases, in the vicinity of X=1.4 the estimated trajectory deviates from the true trajectory because in this range the smoothing algorithm mainly used symmetric averaging. However, if we consider the distribution of the detected positions around the true trajectory, we can conclude that the smoothed trajectory is a sufficiently accurate estimate of the true trajectory.

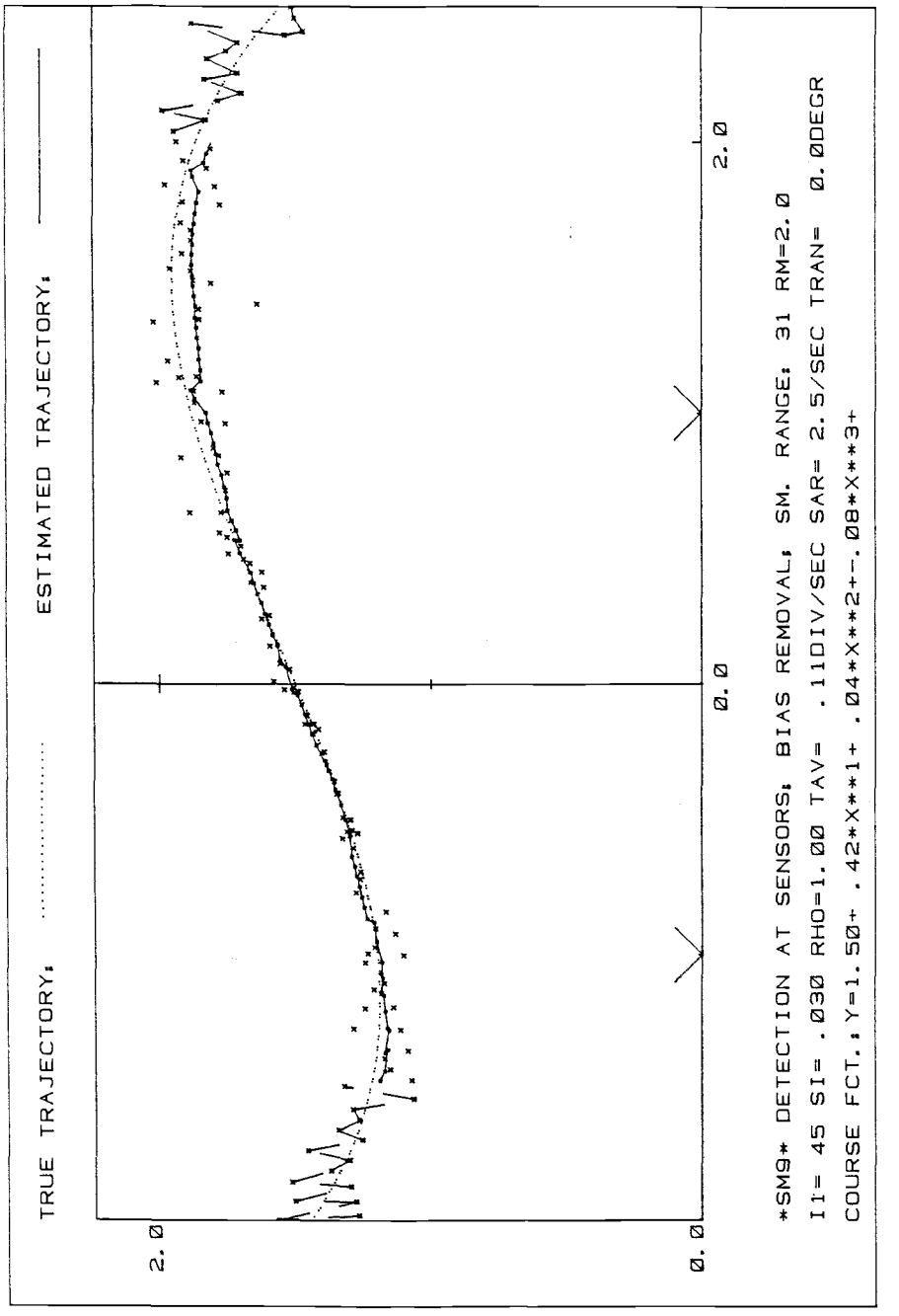

Figure 21: New smoothing algorithm: General trajectory:
σ = 0.03, ρ = +1.0

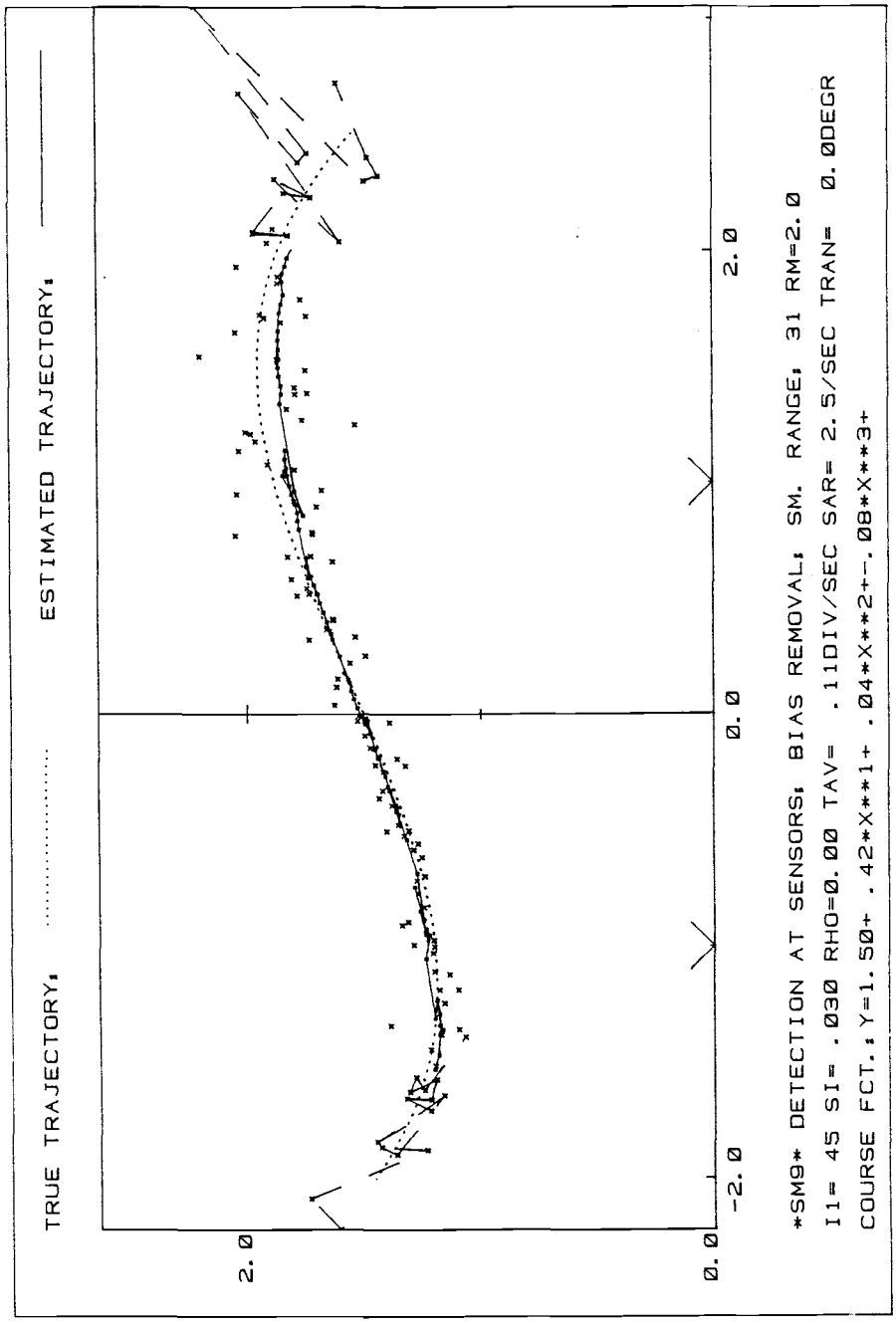

Figure 22: New smoothing algorithm: General trajectory:
σ = 0.03, ρ = 0.0

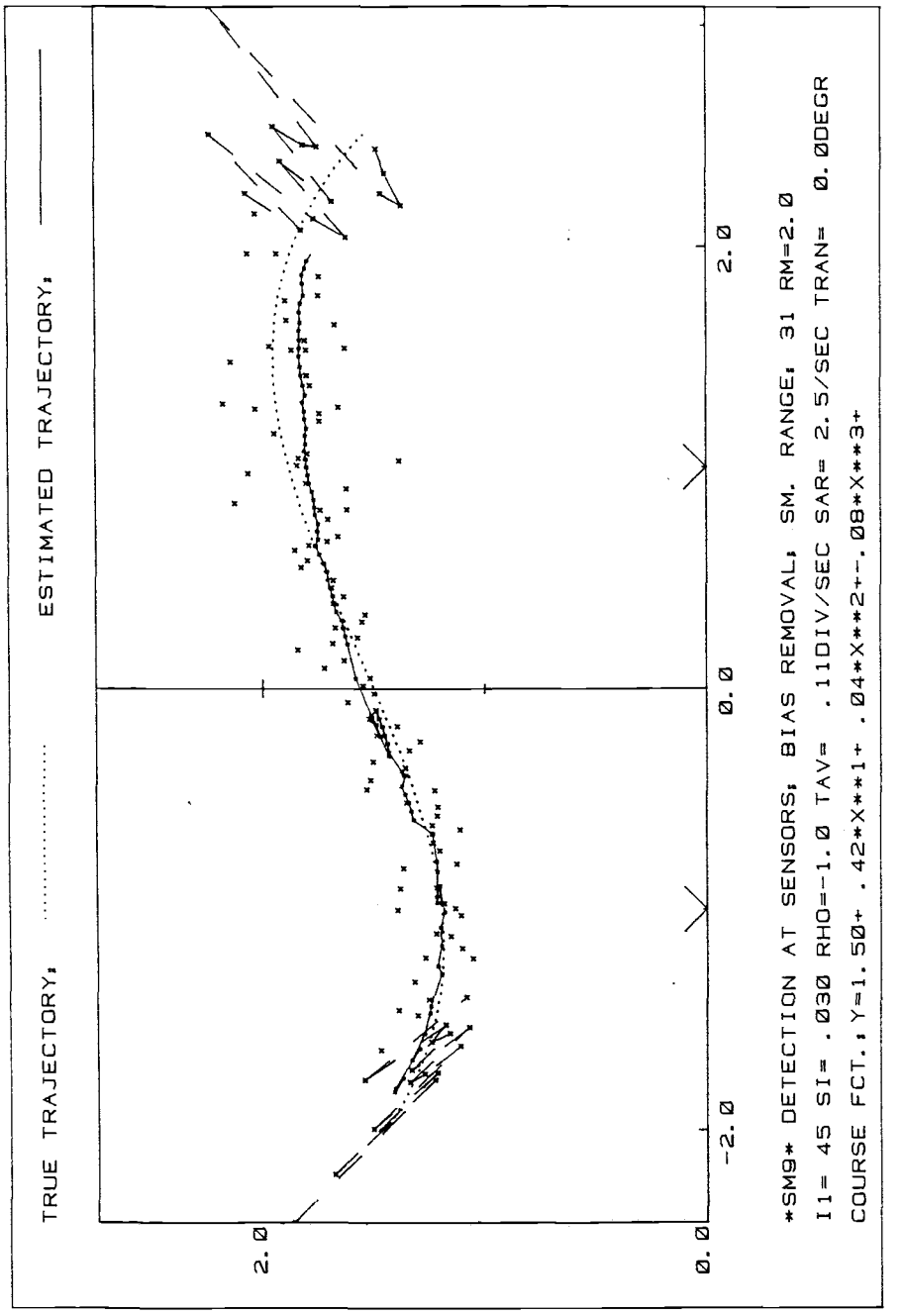

Figure 23: New smoothing algorithm: General trajectory:
σ = 0.03, ρ = -1.0

V. KALMAN FILTERING

To evaluate the performance of the new tracking algorithm, we compare it with Kalman filtering.

We assume the same operating conditions as for the above tracking procedure. This leads to a discrete time signal model with the state vector consisting of the x-y positions, the trajectory angle, the rate of change of the trajectory angle, and the target velocity. Thus, we select as a state vector:

$$\tilde{X}_k = \begin{bmatrix} x_k \\ y_k \\ \beta_k \\ \dot{\beta}_k \\ v_k \end{bmatrix} \tag{46}$$

where x_k, y_k denote the position coordinates of the target at time t_k.

β_k is the trajectory angle at time t_k measured counterclockwise with respect to the positive x-axis.

$\dot{\beta}_k$ is the rate of change of the trajectory angle at time t_k.

v_k is the target velocity along the trajectory at time t_k.

In order to meet the condition of constant radius of curvature and constant target velocity over the time interval T_s (i.e. the smoothing interval for the new smoothing method) $\dot{\beta}$ and v have to stay approximately constant. But since we include all of the past history for evaluating the Kalman filter parameters and at the same time assume that the target is maneuvering, we have to allow the trajectory parameters to change slightly. This is done by introducing an artifical error to the states consisting of gaussian white noise, which is uncorrelated between the states.

Under these conditions the process can be modeled as follows:

$$x_{k+1} = x_k + v_k \Delta T \cos\beta_k + w_x \tag{47a}$$

$$y_{k+1} = y_k + v_k \Delta T \sin\beta_k + w_y \tag{47b}$$

$$\beta_{k+1} = \beta_k + \dot{\beta}\Delta T + w_\beta \tag{47c}$$

$$\dot{\beta}_{k+1} = \dot{\beta}_k + w_{\dot{\beta}} \tag{47d}$$

$$v_{k+1} = v_k + w_v \tag{47e}$$

where ΔT is the time interval between measurements. The covariance matrix Q for the $\tilde{w}_k$ vector is given by:

$$E[\tilde{w}_k \tilde{w}_i^T] = \begin{cases} Q_k & i=k \\ 0 & i\neq k \end{cases} \tag{48}$$

The measured target positions are calculated from the

measurement data at sensors I and II by (3) and (4) and are corrected for the measurement bias by (31a) and (31b). Thus the measured position used for the trajectory estimation represents the true source location at time t_k plus zero-mean two-dimensional noise. We choose as the data vector:

$$\tilde{z}_k = \begin{bmatrix} x_{ok} \\ y_{ok} \end{bmatrix} \tag{49}$$

The measurement can be modeled as follows:

$$\tilde{z}_k = H\tilde{x}_k + \tilde{r}_k \tag{50}$$

where

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{51}$$

and r_k denotes the noise perturbation of the measurement of $H\tilde{x}_k$. The covariance matrix R for the $\tilde{r}_k$ vector is given by:

$$R = \begin{bmatrix} \sigma_x^2 & C_{xy} \\ C_{xy} & \sigma_y^2 \end{bmatrix} \tag{52}$$

where σ_x^2, σ_y^2 and C_{xy} are calculated by (8), (10), and (11), respectively. Since these parameters practically vary with

the position data used for their calculation, the question may arise, which source location should be used. There are three types of source locations available: the random position, calculated from the angle-of-arrival data at sensors I and II; the source location, obtained after bias correction by (31a) and (31b); the source position, predicted by means of the Kalman filter equations, which depend on the past process information. Chosing type one where the parameters of the covariance matrix are calculated from the raw angular data detected by sensors I and II and assuming a sufficiently smooth target trajectory, we introduce a larger uncertainity about the measured source locations. This leads to a decreased Kalman gain, by means of which the filter puts less weight on the measurements. This however, also diminishes the adaptability of the filter to changing trajectory conditions. In practice this could mean that once the filter has chosen a erroneous trajectory it cannot readily get back to the correct trajectory. Despite that, the chosen approach promises to further increase the "filtering" features of the Kalman filter algorithm, especially in a Sonar implementation in which we have to deal with data that were received in an inhomogeneous medium.

Equations (47) and (50) lead to the following nonlinear state space model:

$$\tilde{x}_{k+1} = f_k(\tilde{x}_k) + \tilde{w}_k \qquad (53a,b)$$

$$\tilde{z}_k = H\,\tilde{x}_k + \tilde{r}_k$$

where $\tilde{x}_k$ = (5x1) process state vector at time t_k.

$f_k(\tilde{x}_k)$ = the state transition function at time t_k.

$\tilde{w}_k$ = (5x1) noise vector, assumed to be white (uncorrelated) noise with known covariance structure and constant for any time t_k.

$\tilde{z}_k$ = (2x1) measurement vector at time t_k.

H_k = (2x5) matrix giving the ideal (noiseless) connection between the measurement and the state vector

Following the linearization approach in Ref. 5, $f_k(\tilde{x}_k)$ can be expanded into a Taylor series around the conditional means $\hat{x}_k$:

$$f_k(\tilde{x}_k) = f_k(\hat{x}_k) + F_k(\tilde{x}_k - \hat{x}_k) + \ldots \qquad (54)$$

where

$$F_k = \left.\frac{\partial f_k(\tilde{x})}{\partial \tilde{x}}\right|_{\tilde{x}=\hat{x}_k} \qquad (55)$$

Neglecting higher order terms leads to the following linearized state equations.

$$\tilde{x}_{k+1} = F_k \tilde{x}_k + f_k(\hat{x}_k) - F_k \hat{x}_k + \tilde{w}_k$$

$$\tilde{z}_k = H \tilde{x}_k + \tilde{r}_k \qquad (56a,b)$$

Thus the Kalman filter recursive equations can be derived (Ref. 6):

Kalman gain:

$$K_k = P_k^- H^T \left[H P_k^- H^T + R_k \right]^{-1} \qquad (57)$$

Updating the estimate with measurement $\tilde{z}_k$:

$$\hat{x}_k = \hat{x}_k^- + K_k \left[\tilde{z}_k - H \hat{x}_k^- \right] \qquad (58)$$

Error covariance for updated estimate:

$$P_k = \left[I - K_k H \right] P_k^- \qquad (59)$$

Projecting ahead:

$$\hat{x}_{k+1}^- = f_k(\hat{x}_k) \qquad (60)$$

$$\bar{P}_{k+1} = F_k P_k F_k^T + Q_k \tag{61}$$

The Kalman filter algorithm has to be entered with a prior estimate $\hat{x}_k^-$ and its error covariance $\bar{P}_k$.

Assuming that the target has minimum turning radius, we introduce an additional controlling feature to the Kalman filter model. For that purpose, we calculate the trajectory radius of curvature at time t_k directly after the state vector was updated by the measurement. The radius of curvature is determined by:

$$R_k = \frac{v_k \Delta T}{2 \sin \dot{\beta}_k \frac{\Delta T}{2}} \tag{62}$$

where v_k and $\dot{\beta}_k$ are the updated states 4 and 5 at time t_k.

When the calculated radius of curvature falls below a certain limit (which is specified by the minimum radius of curvature R_{min}) $\dot{\beta}_k$ (state 4) is set to a temporary maximum value calculated by means of the inverse relation (using a Power series expansion of $\sin^{-1}$ and neglecting terms of 5th order and higher (Ref. 7)):

$$\dot{\beta}_k = \frac{v_k}{R_{min}} + \frac{v_k^3 (\Delta T)^2}{24 R_{min}^3} \tag{63}$$

This additional feature should lead to a much smoother trajectory estimate.

SIMULATION RESULTS

In order to get a direct comparison with the smoothing procedure introduced in section IV we apply the Kalman filter under the same operating conditions. Again we make use of the sufficiently smooth trajectory which was given by (45) and performed the simulation for the correlation coefficients of +1.0, 0.0, and -1.0.

As already mentioned, the chosen state model cannot precisely describe every trajectory parameter. Thus, we have to allow the trajectory parameters to change slightly. The Covariance matrix Q for the state model is set to:

$$Q_{11} = 2 \cdot 10^{-5}$$
$$Q_{22} = 2 \cdot 10^{-5}$$
$$Q_{33} = 2 \cdot 10^{-5}$$
$$Q_{44} = 8 \cdot 10^{-5}$$
$$Q_{55} = 2 \cdot 10^{-5}$$

As a prior estimate for $\hat{x}_k^-$, make the following assumptions:

a) States 1 and 2 are set to the first detected source location, which was corrected for bias

b) State 3 (the trajectory angle) is set to zero

c) State 4 (the rate of change of the trajectory angle) is set to a maximum value, determined by means of an assumed maximum target velocity v_{max} and R_{min}

d) State 5 (the target speed) is set to v_{max}

As a prior estimate of the error covariance matrix $\bar{P}_k$, we assume the error to be zero at the beginning of the process.

Thus,

$$\hat{X}_1^- = \begin{bmatrix} X_{01} \\ Y_{01} \\ 0 \\ \frac{V_{max}}{R_{min}} \\ V_{max} \end{bmatrix} \qquad P_1^- = 0$$

The simulation results can be viewed in Figures 24, 25, and 26.

The simulation for $\rho = +1.0$ and $\sigma = 0.03$ (Figure 24) shows that after a settling time the filter gives an unbiased trajectory estimate comparable to the trajectory estimate we obtained with the smoothing technique.

The simulation for $\rho = 0.0$ and $\sigma = 0.03$ (Figure 25) shows that for the uncorrelated case the filter needs much more

time to settle. Despite of the fact that the introduced additional damping factor by means of a minimum radius already leads to a slight improvement of smoothness, the estimated trajectory is still very coarse.

In order to get fairly reasonable simulation results for the case of negative spatial noise correlation, the angular noise had to be limited to σ=0.01. Despite of relative small noise the filter starts to become unstable in the vicinity of X=0.7 (Figure 26).

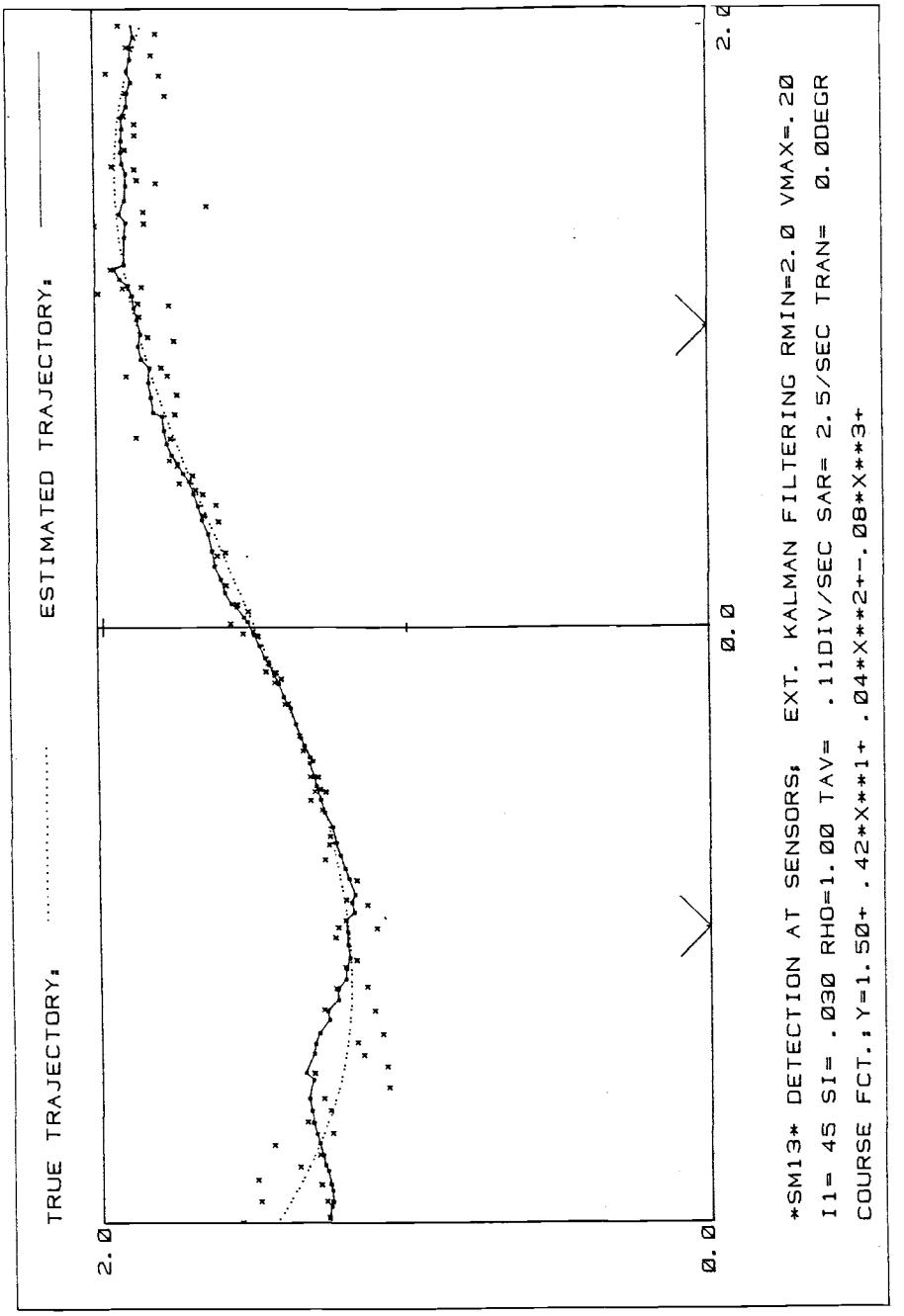

Figure 24: Kalman filtering: General trajectory:
σ = 0.03, ρ = +1.0

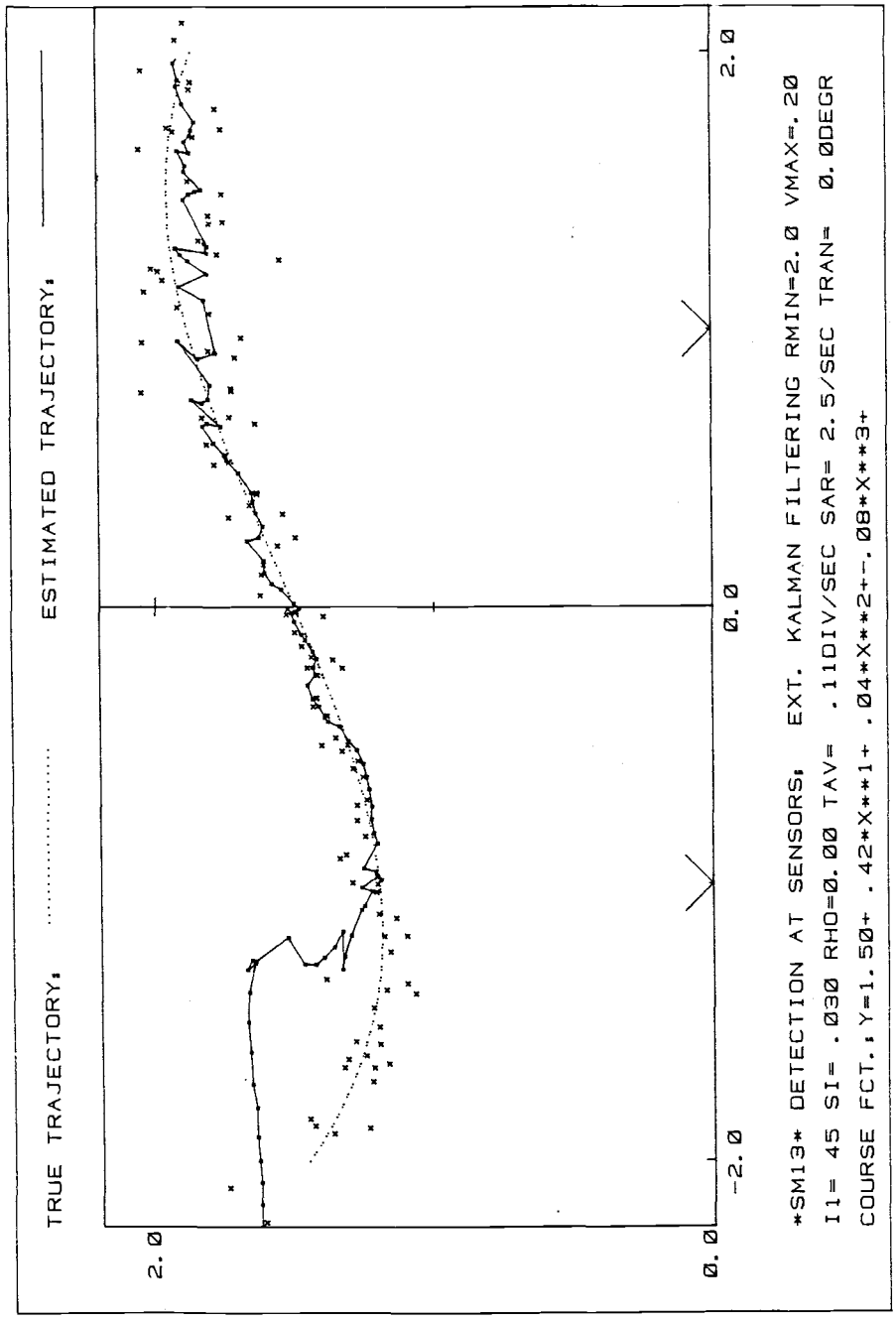

Figure 25: Kalman filtering: General trajectory:
$\sigma = 0.03$, $\rho = 0.0$

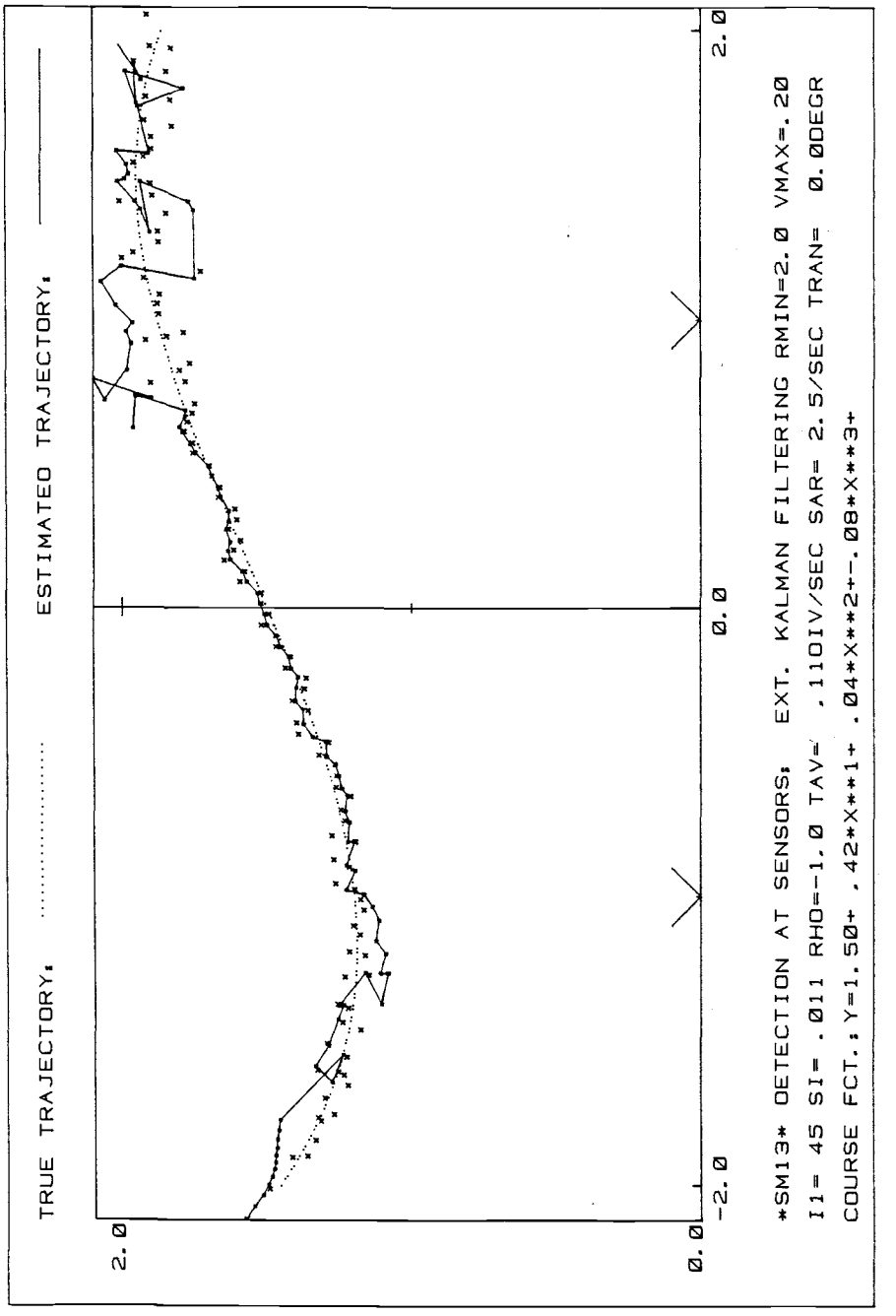

Figure 26: Kalman filtering: General trajectory:
$\sigma = 0.01$, $\rho = -1.0$

VI. SUMMARY AND CONCLUSIONS

We derived and tested an approach for the post-detection of a point-source trajectory in an isotropic, stationary medium. A mathematical model, describing stationary target locations which are estimated from angle-of-arrival measurements detected at two sensors, was described and tested. The angular data were regarded as random variables whose mean indicates the true (unbiased) source location and whose variance is constant, direction independent. The angular noise may be spatially correlated between the sensors with correlation coefficients ranging from $-1.0<\rho<+1.0$. The source location estimated from the angular data has a bias and variance depending on the source-sensor geometry and spatial noise correlation.

We first examined the mathematical model by simulating a stationary target at various locations in the simulation plane.

We then introduced a tracking algorithm which operates under the assumption that the target trajectory has constant but unknown radius of curvature and the target is travelling at constant but unknown speed. The algorithm alternatively uses asymmetric and conventional symmetric smoothing to determine the target trajectories. Asymmetric smoothing is obtained by individually averaging two subsets

of symmetric data in order to obtain two regression lines. The intersection of these two regression lines leads to an unbiased estimate of the true target location. This algorithm can also to a limited extend (i.e. with a time delay) be applied to on-line tracking.

We finally performed extended Kalman filtering in order to compare it with the introduced smoothing algorithm for trajectory estimation.

The conclusions are as follows:

1) The simulation showed that for a limited range of angular noise standard deviation, the mathematical model is a realistic description of a source-sensor configuration in an isotropic, stationary medium.

2) The introduced smoothing algorithm proved to perform a very good trajectory estimate. The simulation further showed that the smoothing algorithm is very robust to strong angular noise perturbations received by the two sensors.

3) Comparing the trajectory estimates achieved from the introduced smoothing method and from a conventional extended Kalman filter shows that despite the striking simplicity of the smoothing method, the introduced smoothing algorithm surpasses a relatively sophisticated Kalman filter model.

To continue the investigation of passive source tracking from spatially correlated angle-of-arrival data, we could analyse the impact of higher moments of noise variance and correlation coefficient on estimator bias and variance. Chosing a different approach for the stationary case, estimating the source position by means of the median value (i.e., arranging N source locations calculated from the angle-of-arrival data in the order of increasing distance from the detectors and then taking the N/2-th position as the estimated source location) promises to make the estimation of bias and variance less sensitive to positions out of range.

The smoothing procedure could be further improved by an algorithm that eliminates positions that indicate "backwards" travelling targets. Repeatedly smoothing over already smoothed positions should lead to a much smoother trajectory estimate.

BIBLIOGRAPHY

[1] Rudolf S. Engelbrecht, "Passive Source Localization from Spatially Correlated Angle-of-Arrival Data", IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-31, no. 4, pp. 842-846, 1983

[2] A. Papoulis, Probability, Random Variables, and Stochastic Processes. New York: Mc Graw-Hill, 1965, p. 212.

[3] R.S. Burlington and D.C. May, Handbook of Probability and Statistics with Tables. Ohio: Handbook Publishers, 1953, p. 84

[4] Rudolf S. Engelbrecht,"Passive Tracking Data Update Algorithm", Report to NUWES, March 1983.

[5] B.D.O. Anderson, J.B. Moore, Optimal Filtering. New Jersey: Prentice-Hall, 1979, pp. 193-205.

[6] R.G. Brown, Introduction to Random Signal Analysis and Kalman Filtering. New York: John Wiley & Sons, 1983, pp. 188-200.

[7] I. Bronstein, K. Semendjajew, Taschenbuch der Mathematik, Frankfurt/Main: Verlag Harri Deutsch, 1969, p. 279/281

Appendices

APPENDIX A: Derivation of the Ratio 1/u

Assume the configuration of the circle segment shown in Figure 27. The following derivation leads to the ratio 1/u of the distances AB and BC, such that the circle segment is approximated by a regression line with minimum mean squared error.

A Cartesian coordinate system is chosen, such that the circle center is located at Y=0 and the segment center of gravity is located at X=0. Under these assumptions the regression line can be expressed by the following equation:

$$Y_r = m = R \cdot \cos\beta \tag{64}$$

where m = const, R = radius of curvature, β = angle between the y-axis and point B.

Define the error as the distance between a particular location on the circle segment and the regression line measured perpendicular to the regression line:

$$e = y - m \tag{65}$$

The mean squared error

$$E[e^2] = E[y^2] - 2m\,E[y] + m^2 \tag{66}$$

is minimum for

$$E[y] = m \tag{67}$$

Under the condition that the target locations are uniformly distributed along the trajectory, E[y] can be expressed as

$$E[y] = \int_0^{\alpha} R\cos\phi \, \frac{1}{\alpha} \, d\phi \tag{68}$$

leading to (using (64))

$$E[y] = \frac{R}{\alpha} \sin\alpha = R\cos\beta \tag{69}$$

Expanding $\sin^{-1}$ and $\cos^{-1}$ into a Power series and neglecting terms of forth order and higher (Ref. 7)

$$\frac{\sin\alpha}{\alpha} = \frac{1}{\alpha}\left[\alpha - \frac{\alpha^3}{3!}\right] = \cos\beta = \left[1 - \frac{\beta^2}{2!}\right] \tag{70}$$

leads to the condition

$$\frac{\ell}{u} = \frac{\alpha+\beta}{\alpha-\beta} = \sqrt{3} + 2 \tag{71}$$

and finally (using (71)) to the asymmetry ratio l/u of

$$\frac{\alpha^2}{\beta^2} = 3 \tag{72}$$

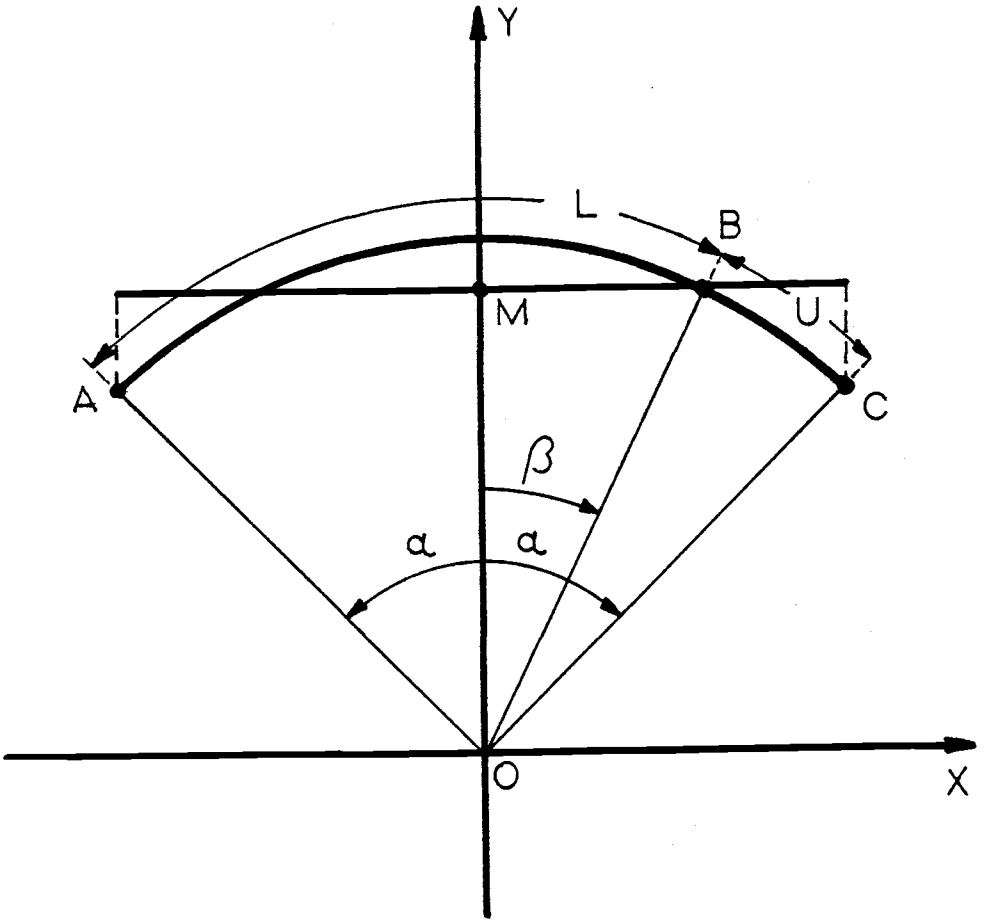

Figure 27: Approximation of a circle segment by a regression line

APPENDIX B: Generation of Correlated Gaussian Number Pairs

The following algorithm generates N correlated Gaussian number pairs:

Given: $\overline{\Delta\theta}_I = \overline{\Delta\theta}_{II} = 0$

$\sigma_{\Delta\theta_I} = \sigma_{\Delta\theta_{II}} = \sigma$

correlation coefficient ρ

1) Generate random numbers:

Z1 ... Z1 $\qquad \overline{Z1} = \overline{Z2} = 0$

Z2 ... Z2 $\qquad \sigma_{Z1} = \sigma_{Z2} = 1$

(Generation by call to subroutine GAUSS) *

2) Generate correlated random numbers:

R1 ... R1

R2 ... R2

$$R1 = \sqrt{1 - \rho^2}\ Z2 + \rho Z1$$

$$R2 = Z1$$

3) Generate random noise

$$\Delta\theta_I = \sigma \cdot R1$$

$$\Delta\theta_{II} = \sigma \cdot R2$$

4) Calculate random angles

$$\theta_I = \theta_1 + \Delta\theta_I$$
$$\theta_{II} = \theta_2 + \Delta\theta_{II}$$

*) Subroutine GAUSS creates two sequences of random numbers each created by various calls to a random generator. The sequences are normalized to zero-mean and unit standard deviation. (Refer to section "Test procedures" and to the listing of subroutine GAUSS in Appendix C.)

APPENDIX C

The following program was used to create the distributions of estimator bias and variance in the x-y plane.

```
FTN7X,I
      PROGRAM PLOT
      REAL Z1(100),Z2(100),B(19),YD(30), S(19),XM(6)
      INTEGER XN,YN,R(30,30),BMI(30,30),BMA(30,30),
     1SMI(30,30),SMA(30,30)
      PI=3.141592654
C
C **** DATA INPUT ***************************************
C
1000  WRITE(1,1)
1     FORMAT("RANGE: 0 < X <= XMAX   XMAX <= 2.8",/,8X,
     1"0 < Y <="," YMAX   YMAX <= 2.8",/,"XMAX = ?")
      READ*,XMAX
      WRITE(1,2)
2     FORMAT("YMAX = ?")
      READ*,YMAX
      WRITE(1,3)
3     FORMAT("ENTER STEPWIDTH: STEP >= 0.1")
      READ*,STEP
      WRITE(1,4)
4     FORMAT("ENTER RATIO STANDARD DEV. TO ANGLE"
     1" DIFFERENCE: A = ?")
      READ*,A
      WRITE(1,5)
5     FORMAT("ENTER SAMPLE SIZE: N = ?")
      READ*,N
      WRITE(1,6)
6     FORMAT("ENTER RANDOM GENERATOR BASIS: I1 = ?")
      READ*,I1
C
C **** INITALIZATION ************************************
C
      CALL GAUSS(I1,Z1,Z2,N)
      XN=INT(XMAX/STEP)
      YN=INT(YMAX/STEP)
C
C **** MAIN CALCULATION LOOP ****************************
C
      DO 100 I=1,YN
      Y=FLOAT(I)*STEP
      WRITE(1,101)Y
```

```
101   FORMAT("  Y = ",F4.2)
      DO 200 J=0,XN
      X=FLOAT(J)*STEP
      TO1=ATAN(Y/(1.+X))
      TO2=ATAN(Y/(X-1.))
      IF(X .LT. 1.0) TO2=TO2+PI
      SIGMA=A*(TO2-TO1)
      DO 300 K=1,19
      RHO=-1.+FLOAT(K)*.1
      CALL RADAT(Z1,Z2,SIGMA,TO1,TO2, BIAS,SDX,SDY,X,Y,RHO,
     1N)
      B(K)=BIAS
300   S(K)=SDX*SDX+SDY*SDY
      BMIN=B(1)
      BMAX=B(1)
      RMIN=1
      SMIN=S(1)
      SMAX=S(1)
      DO 400 K=2,19
      IF(BMIN .LE. B(K)) GO TO 401
      BMIN=B(K)
      RMIN=K
401   IF(BMAX .LT. B(K)) BMAX=B(K)
      IF(SMAX .LT. S(K)) SMAX=S(K)
400   IF SMIN .GT. S(K)) SMIN=S(K)
      R(J,I)=RMIN
      BMI(J,I)=INT(100.*BMIN/B(10))
      BMA(J,I)=INT(10.*BMAX/B(10))
      SMI(J,I)=INT(100.*SMIN/S(10))
      SMA(J,I)=INT(10.*SMAX/S(10))
200   CONTINUE
100   CONTINUE
      WRITE(6,500)I1,A
500   FORMAT(1H1,"PLOT DATA CORRELATION COEFFICIENT (RMIN)"
     1" FOR MINIMUM BIAS VS. SOURCE LOCATION",//," RANDOM "
     2"GENERATOR BASIS: I1 =",I3,"  RATIO STD. TO ANGLE "
     3"DIFF.: A =",F4.2,//,"VALUES: 1 <=> RHO= -0.9 ....."
     4" 19 <=> RHO= +0.9",////)
      DO 510 I=YN,1,-1
      YD(I)=STEP*FLOAT(I)
      WRITE(6,520)YD(I),R(J,I),J=0,XN)
520   FORMAT(1X,F3.1," *",15(I2,4X))
      WRITE(6,525)
525   FORMAT(5X,"*",/,5X,"*",/,5X,"*")
510   CONTINUE
      DO 515 I=1,6
      XM(I)=FLOAT(I)*2.5*STEP
515   CONTINUE
      WRITE(6,530)(XM(I),I=1,6)
530   FORMAT(1X,"0.0",1X,120("*"),/,7X,"*",6(14X,"*"),/,6X,
     1"0.0",6(12X,F3.1))
      WRITE(6,540)I1,A
```

```
540   FORMAT(1H1,"PLOT NORMALIZED MINIMUM BIAS (BMIN/B0)"
     1" VS. SOURCE LOCATION",//," DATA PRINTED ARE 100 "
     2"TIMES THE ACTUAL VALUES     $$: ACTUAL VAULE >=1.0"
     3//," RANDOM GENERATOR BASIS: I1=",I3,/"   RATIO STD."
     4" TO ANGLE DIFF.: A=",F4.2,////)
      DO 550 I=YN,1,-1
      WRITE(6,520)YD(I),(BMI(J,I),J=0,XN)
      WRITE(6,525)
550   CONTINUE
      WRITE(6,530)(XM(I),I=1,6)
      WRITE(6,560)I1,A
560   FORMAT(1H1,"PLOT NORMALIZED MAXIMUM BIAS (BMAX/B0)"
     1" VS. SOURCE LOCATION",//," DATA PRINTED ARE 10 "
     2"TIMES THE ACTUAL VALUES     $$: ACTUAL VALUE >= "
     3"1.0",//," RANDOM GENERATOR BASIS:",I3,/"   RATIO "
     4"STD. TO ANGLE DIFF.: A=",F4.2,////)
      DO 570 I=YN,1,-1
      WRITE(6,520)YD(I),(BMA(J,I),J=0,XN)
      WRITE(6,525)
570   CONTINUE
      WRITE(6,530)(XM(I),I=1,6)
      WRITE(6,580)I1,A
580   FORMAT(1H1,"PLOT NORMALIZED MINIMUM VARIANCE "
     1"(SMIN/S0) VS. SOURCE LOACTION",//," DATA PRINTED"
     2"  ARE 100 TIMES THE ACTUAL VALUES    $$: ACTUAL "
     3"VALUE >= 1.0",//," RANDOM GENERATOR BASIS: I1=",I3,
     4"    RATIO STD. TO ANGLE DIFF.: A=",4.2,////)
      DO 581 I=YN,1,-1
      WRITE(6,520)YD(I),(SMI(J,I),J=0,XN)
      WRITE(6,525)
581   CONTINUE
      WRITE(6,530)(XM(I),I=1,6)
      WRITE(6,590)I1,A
590   FORMAT(1H1,"PLOT NORMALIZED MAXIMUM VARIANCE "
     1"(SMAX/S0) VS. SOURCE LOCATION",//," DATA PRINTED"
     2" ARE 10 TIMES THE ACTUAL VALUES    $$: ACTUAL "
     3"VALUE >= 10.0"//," RANDOM GENERATOR BASIS: I1=",I3,
     4"    RATIO STD. TO ANGLE DIFF.: A=",F4.2,////)
      DO 591 I=YN,1,-1
      WRITE(6,520) YD(I),(SMA(J,I),J=0,XN)
      WRITE(6,525)
591   CONTINUE
      WRITE(6,530)(XM(I),I=1,6)
      WRITE(1,600)
600   FORMAT(" NEW DATA ?  YES = 1  NO = 0")
      READ*,L
      IF(L)610,610,1000
610   STOP
      END
```

```
C
C **** RANDOM POSITION GENERATION AND CALCULATION OF BIAS *
C       AND VARIANCE
C
      SUBROUTINE RADAT(Z1,Z2,SIGMA,TO1,TO2,BIAS,SDX,SDY,XS,
     1YS,RHO,N)
      REAL X(100),Y(100),Z1(100),Z(200)
C * INITIALIZATION *
      SUMX=0.
      SUMY=0.
      SUMSX=0.
      SUMSY=0.
C * CREATION OF RANDOM POSITIONS *
      DO 10 I=1,N
      D=SQRT(1.0-RHO*RHO)
      R1=D*Z2(I)+RHO*Z1(I)
      DT1=Z1(I)*SIGMA
      DT2=R1*SIGMA
      T1=TO1+DT1
      T2=TO2+DT2
      X(I)=SIN(T1+T2)/SIN(T2-T1)
10    Y(I)=(COS(T2-T1)-COS(T2+T1))/SIN(T2-T1)
C * CALCULATION OF BIAS AND VARIANCE *
      DO 20 I=1,N
      SUMX=SUMX+X(I)
      SUMY=SUMY+Y(I)
      SUMSX=SUMSX+X(I)*X(I)
20    SUMSY=SUMSY+Y(I)*Y(I)
      XA=SUMX/FLOAT(N)
      YA=SUMY/FLOAT(N)
      SDX=SQRT((SUMSX-XA*XA*FLOAT(N))/FLOAT(N-1))
      SDY=SQRT((SUMSY-YA*YA*FLOAT(N))/FLOAT(N-1))
      BIAS=SQRT((XA-XS)**2+(YA-YS)**2)
      RETURN
      END

C
C **** GENERATION OF UNBIASED RANDOM NUMBERS *************
C       WITH STD. DEV. = 1.0
C
      SUBROUTINE GAUSS(I1,Z1,Z2,N)
      REAL G1(100),G2(100),Z1(100),Z2(100)
      INTEGER I1,N
C * INTIALIZING GENERATORS *
      CALL SSEED(I1)
C * GENERATION OF RANDOM NUMBERS *
      S1=.0
      S2=.0
      SS1=.0
      SS2=.0
      DO 10 I=1,N
      G1(I)=GRAN(1)
```

```
      G2(I)=GRAN(2)
      S1=S1+G1(I)
      S2=S2+G2(I)
      SS1=SS1+G1(I)*G1(I)
10    SS2=SS2+G2(I)*G2(I)
      GA1=S1/FLOAT(N)
      GA2=S2/FLOAT(N)
      GS1=(SS1-FLOAT(N)*GA1*GA1)/FLOAT(N-1)
      GS2=(SS2-FLOAT(N)*GA2*GA2)/FLOAT(N-1)
      DO 20 I=1,N
      Z1(I)=G1(I)/SQRT(GS1)-GA1
20    Z2(I)=G2(I)/SQRT(GS2)-GA2
      RETURN
      END
```

The flexible smoothing technique introduced in section IV uses the following FORTRAN commands:

```
C
C **** SMOOTHING PROCEDURE **********************************
C
      B=SQRT(3.0)+2.0
      WRITE(1,13)
13    FORMAT(" # OF LOWER POSITIONS TO BE SMOOTHED ?  "
     1"M>=4 ")
      READ*,M
      NB=INT(FLOAT(M)/B)
      WRITE(1,14)
14    FORMAT(" RATIO DISTANCE TO PREVIOUS POSITION "
     1"TO AVER.DISTANCE: RM=? ")
      READ*,RM
      DO 250 I=1,M
      XS(I)=XR(I)
      YS(I)=YR(I)
250   CONTINUE
      DO 260 I=N-M+1,N
      XS(I)=XR(I)
      YS(I)=YR(I)
260   CONTINUE
      DO 270 I=1,N
270   BML(I)=0.0
      BS=.0
      DO 280 I=2,M
      BM=SQRT((XS(I)-XS(I-1))**2+(YS(I)-YS(I-1))**2)
      BS=BS+BM
 280  CONTINUE
C
C *** ASYMMETRIC SMOOTHING: ********************************
C     INTERSECTION OF TWO REGRESSIONLINES
```

```
C
      DO 300 I=M+1,N-M
      L=I-M
      H=I+NB
      CALL RLINE (XR,YR,L,H,M1,XA1,YA1,SIGMA,RHO)
      H=I+M
      L=I-NB
      CALL RLINE (XR,YR,L,H,M2,XA2,YA2,SIGMA,RHO)
      XNL=(M1*XA1-M2*XA2+YA2-YA1)/(M1-M2)
      YNL=YA1-M1*XA1+M1*XNL
C * SPEED CONTROL: DECISION LEVEL *
      BMNL(I)=SQRT((XNL-XS(I-1))**2+(YNL-YS(I-1))**2)
      BMA(I)=BS/FLOAT(M-1)
      IF(BMNL(I) .LT. RM*BMA(I)) GOTO 320
C
C **** SMOOTHING OVER A SYMMETRIC WINDOW ****************
C
      SUMX=.0
      SUMY=.0
      DO 310 L=I-M,I+M
      CALL BIREM(L,RHO,SIGMA,XUB,YUB,XR,YR)
      SUMX=SUMX+XUB
      SUMY=SUMY+YUB
 310  CONTINUE
      J=2*M+1
      XL=SUMX/FLOAT(J)
      YL=SUMY/FLOAT(J)
C * SECOND SPEED CONTROL:
C   DECISION FOR ASYMMETRIC OR SYMMETRIC SMOOTHING
      BML(I)=SQRT((XL-XS(I-1))**2+(YL-YS(I-1))**2)
      IF(BML(I) .GT. BMNL(I)) GOTO 320
      XS(I)=XL
      YS(I)=YL
      BS=BS+BML(I)
      GOTO 325
 320  XS(I)=XNL
      YS(I)=YNL
      BS=BS+BMNL(I)
 325  IS=I-M
      ISP=IS+1
      BS=BS-SQRT((XS(ISP)-XS(IS))**2+(YS(ISP)-YS(IS))**2)
 300  CONTINUE

C
C **** REGRESSION LINE ***********************************
C
      SUBROUTINE RLINE (XR,YR,L,H,MS,XA,YA,SIGMA,RHO)
      REAL XR(300),YR(300),MS,XA,YA,RHO,SIGMA
      INTEGER L,H,J,IS
      SUMX=.0
      SUMY=.0
      SUMXY=.0
```

```
      SUMSX=.0
      SUMSY=.0
      DO 120 IS=L,H
      CALL BIREM(IS,RHO,SIGMA,XUB,YUB,XR,YR)
      SUMX=SUMX+XUB
      SUMY=SUMY+YUB
      SUMXY=SUMXY+XUB*YUB
      SUMSX=SUMSX+XUB*XUB
      SUMSY=SUMSY+YUB*YUB
  120 CONTINUE
      J=H-L+1
      XA=SUMX/FLOAT(J)
      YA=SUMY/FLOAT(J)
      SSX=SUMSX/FLOAT(J)-XA**2
      SSY=SUMSY/FLOAT(J)-YA**2
      RSXY=SUMXY/FLOAT(J)-XA*YA
      D=SSY-SSX
      C=SQRT(1.+(2.*RSXY/D)**2)
      IF(D .LE. 0.0) C=-1.*C
      MS=D*(1.0+C)/(2.*RSXY)
      RETURN
      END

C
C **** BIAS REMOVAL ****************************************
C
      SUBROUTINE BIREM(I,RHO,SIGMA,XUB,YUB,XR,YR)
      REAL T1,T2,XR(300),YR(300),XUB,YUB,RHO,SIGMA
      PI=3.141592654
      T1=ATAN(YR(I)/(1.+XR(I)))
      T2=ATAN(YR(I)/(XR(I)-1.))
      IF((XR(I) .LT. 1.) .AND. (YR(I) .GE. .0))T2=T2+PI
      IF((XR(I) .LT. 1.) .AND. (YR(I) .LT. .0))T2=T2-PI
      IF((XR(I) .LT. -1.) .AND. (YR(I) .GE. .0))T1=T1+PI
      IF((XR(I) .LT. -1.) .AND. (YR(I) .GE. .0))T1=T1-PI
      DI=T2-T1
      SU=T2+T1
      B=2.*SIGMA**2/(SIN(DI))**3
      E=(COS(DI))**2*(1.-2.*(COS(SU))*COS(DI)+(COS(DI))**2)
      F=2.*COS(DI)*(COS(SU)*(1.+(COS(DI))**2)-2.*COS(DI))
      G=1.-2.*COS(SU)*COS(DI)+(COS(DI))**2
      BIAS=B*SQRT(E+RHO*F+RHO*RHO*G)
      CON=(1.-RHO)*COS(DI)
      DIN=SIN(SU)*((COS(DI))**2-RHO)
      DEL=ATAN(CON/DIN-1./TAN(SU))
      XUB=XR(I)-BIAS*COS(DEL)
      YUB=YR(I)-BIAS*SIN(DEL)
      RETURN
      END
```

The Kalman filter introduced in section V consists of the following FORTRAN commands:

```
C
C **** KALMAN FILTERING ************************************
C
      SUBROUTINE KALM (N,RHO,SIGMA,SR,XS,YS,XR,YR,RMIN,
     1VMAX)
      REAL PH(5,5),PHT(5,5),U5(5,5),R(2,2),Q(5,5),P(5,5),
     1PM(5,5),KA(5,2),Z(2),X(5),XM(5),XR(301),YR(301),
     2XS(301),YS(301),DUM(5,5),DUV(2),DU52(5,2),DU22(2,2),
     3DU22H(2,2),DUV5(5),H(2,5),HT(5,2),DUMH(5,5)
C * PREPARATION OF MATRICIES **********
      CALL VMOV(0.,0,U5,1,25)
      CALL VMOV(1.,0,U5,6,5)
      CALL VMOV(0.,0,PH,1,25)
      CALL VMOV(1.,0,PH,6,5)
      PH(1,3)=1./SR
      PHT(4,3)=PH(3,4)
      CALL VMOV(0.,0,Q,1,25)
      Q(1,1)=.00002
      Q(2,2)=.00002
      Q(3,3)=.00002
      Q(4,4)=.00008
      Q(5,5)=.00002
      CALL VMOV(0.,0,H,1,10)
      H(1,1)=1.
      H(2,2)=1.
      CALL VMOV(0.,0,HT,1,10)
      HT(1,1)=1.
      HT(2,2)=1.
C * INITIALIZATION *********************
      K=1
      CALL BIREM(K,RHO,SIGMA,XUB,YUB,XR,YR)
      XM(1)=XUB
      XM(2)=YUB
      XM(3)=0.
      XM(4)=VMAX/RMIN
      XM(5)=VMAX
      CALL VMOV(0.,0,PM,1,25)
      DO 100 K=1,N
C * KALMAN GAIN ************************
      CALL MULT(PM,HT,DU52,5,5,2)
      CALL MULT(H,DU52,DU22,2,5,2)
      CALL COV(K,RHO,SIGMA,R,XR,YR)
      CALL VADD(DU22,1,R,1,DU22,1,4)
      S=DU22(1,1)*DU22(2,2)-DU22(1,2)*DU22(2,1)
      DU22H(1,1)=DU22(2,2)
      DU22H(1,2)=-DU22(1,2)
      DU22H(2,1)=-DU22(2,1)
```

```
      DU22H(2,2)=DU22(1,1)
      SI=1./S
      CALL VSMY(SI,DU22H,1,DU22,1,4)
      CALL MULT(HT,DU22,DU52,5,2,2)
      CALL MULT(PM,DU52,KA,5,5,2)
C * UPDATE ESTIMATE ********************
      CALL MULT(H,XM,DUV,2,5,1)
      CALL BIREM(K,RHO,SIGMA,XUB,YUB,XR,YR)
      Z(1)=XUB
      Z(2)=YUB
      CALL VSUB(Z,1,DUV,1,DUV,1,2)
      CALL MULT(KA,DUV,DUV5,5,2,1)
      CALL VADD(XM,1,DUV5,1,X,1,5)
      RAD=X(5)*PH(3,4)*.5/SIN(.5*X(4)*PH(3,4))
      IF(RAD .LE. RMIN) X(4)=X(5)/RMIN+X(5)**3*PH(3,4)**3/
     1RMIN**3/24.
C * ERROR COVARIANCE ********************
      CALL MULT(KA,H,DUM,5,2,5)
      CALL VSUB(U5,1,DUM,1,DUM,1,25)
      CALL MULT(DUM,PM,P,5,5,5)
C * PROJECT AHEAD ***********************
      PH(2,5)=PH(3,4)*SIN(X(3))
      PH(1,5)=PH(3,4)*COS(X(3))
      PH(1,3)=-X(5)*PH(2,5)
      PH(2,3)=X(5)*PH(1,5)
      PHT(5,1)=PH(1,5)
      PHT(5,2)=PH(2,5)
      PHT(3,1)=PH(1,3)
      PHT(3,2)=PH(2,3)
      XM(1)=X(1)+X(5)*PH(3,4)*COS(X(3))
      XM(2)=X(2)+X(5)*PH(3,4)*SIN(X(3))
      XM(3)=X(3)+X(4)*PH(3,4)
      XM(4)=X(4)
      XM(5)=X(5)
      CALL MULT(P,PHT,DUM,5,5,5)
      CALL MULT(PH,DUM,DUMH,5,5,5)
      CALL VADD(DUMH,1,Q,1,PM,1,25)
      XS(K)=X(1)
100   YS(K)=X(2)
      RETURN
      END
C
C **** MATRIX MULTIPLICATION ******************************
C
      SUBROUTINE MULT(A,B,C,N,M,L)
      REAL A(N,M),B(M,L),C(N,L),DUM(5,5)
      DO 100 J=1,L
      DO 200 I=1,N
200   CALL VMPY(A(I,1),N,B(1,J),1,DUM(I,1),4,M)
      DO 100 I=1,N
100   CALL VSUM(C(I,J),DUM(I,1),4,M)
      RETURN
```

```
      END
C
C **** MEASUREMENT COVARIANCE MATRIX **********************
C
      SUBROUTINE COV(K,RHO,SIGMA,R,XR,YR)
      REAL R(2,2),XR(301),YR(301)
      CALL BIREM(K,RHO,SIGMA,XUB,YUB,XR,YR)
      SU=ATAN(2.*XUB*YUB/(XUB*XUB-YUB*YUB-1.))
      DI=ATAN(2.*YUB/(XUB*XUB+YUB*YUB-1.))
      SP=SIN(SU)
      SM=SIN(DI)
      CP=COS(SU)
      CM=COS(DI)
      C2P=COS(2.*SU)
      C2M=COS(2.*DI)
      SM3=SM**3
      SM4=SM*SM3
      SSGMA=SIGMA*SIGMA
      R(1,1)=SSGMA/SM4*(1.-C2P*C2M+RHO*(C2P-C2M))
      R(2,2)=2.*SSGMA/SM4*((1.-CP*CM)**2+(SP*SM)**2-RHO*
     1((CM-CP)**2))
      R(2,1)=2.*SP/SM4*(CM-CP*C2M+RHO*(CP-CM))*SSGMA
      R(1,2)=R(2,1)
      RETURN
      END
```