Image Cover Sheet

CLASSIFICATION	SYSTEM NUMBER 500711
UNCLASSIFIED	

TITLE

REGISTRATION FOR DATA FUSION BY EXACT MAXIMUM LIKELIHOOD METHOD

System Number:

Patron Number:

Requester:

Notes:

DSIS Use only:

Deliver to:

96-745

COMMUNICATIONS
RESEARCH
LABORATORY

REGISTRATION FOR DATA FUSION BY EXACT MAXIMUM LIKELIHOOD METHOD

by

Yifeng Zhou
and
Patrick Yip

Faculty of Engineering
McMaster University
Hamilton, Ontario
Canada L8S 4K1

CRL Report No. 323
March 1996

Registration

for

Data Fusion

by

Exact Maximum Likelihood

Method

by

Yifeng Zhou & P. Yip

Communications Research Laboratory
McMaster University
Hamilton, Ontario,
Canada, L8S 4K1.

This report applies to DSS Contract No. W7714-4-9827/01-SV

Table of contents:

Abstract:

The combination of data and information from multiple sources or sensors constitutes one of the most important problems in signal processing to-day. This process of data fusion has been defined as one of dealing with the association, correlation and combination of data. In this report, an exact maximum likelihood method is used to solve the problem of registration errors in data fusion. Registration is defined as the co-ordinate conversion of multiple source data. Error-free registration is a pre-requisite in the process of data fusion. In the exact maximum likelihood method (EML), a likelihood function is constructed based on the errors of co-ordinate transformation of the local sensor locations to a common system plane. Optimization is then carried out in a recursive two-step Gauss-Newton type procedure, which produces the correct system biases for proper registration. Simulation results show that EML is more efficient when compared with conventional methods of registration.

Introduction

Data fusion has been defined as "a process dealing with association, correlation and combination of data and information from single and multiple sources to achieve refined position and identity estimates" [1]. This aspect of signal processing has increased in importance in recent years [2] owing to the advance in technology of sensor systems which are becoming more sophisticated and complex. As more of these sensor systems become available, the combination of information from a multitude of sources and the decision on proper reaction turn into a significant problem [3]. This fusion process is usually implemented at three levels. Level 1 includes registration, association, filtering and identification. Level 2 refers to situation assessment and level 3 determines whether a threat is present [4]. In this report, the focus is on level 1 processing and, in particular, on the registration problem [5].

The process of proper co-ordinate conversion of multiple source data is called registration. Generally, this is required because the sensors are not aligned properly for fusion. The major source of registration error are the azimuth bias error with respect to a common reference and the range offset errors of each sensor. These errors in turn will introduce biases in fusion and may generate "ghost" targets in multisensor signal processing [6,7]. To "register" a sensor system, each sensor has to be initialized and then aligned to a common reference frame. Initialization refers to the independent localization of each sensor in the system co-ordinates. Relative alignment is done by using common targets and collecting data points from each sensor. A set of system errors (for proper registration) is then computed and used to adjust subsequent incoming data in the data fusion process.

Several registration methods have been proposed. These include the simple averaging method called Real Time Quality Control (RTQC) [8], the least squares (LS) method [9], the maximum likelihood (ML) solution [10] and the generalized least squares (GLS) technique [5]. The RTQC routine computes the system errors by averaging the measurement data. The LS formulates the registration as an ordinary or unweighted least squares problem, while GLS includes the target and observation error covariance in the formulation. These methods are found to yield reasonable results only when the measurement noise is relatively small. The major drawback is the assumption that the measurements from different sensors are "equal" in a common co-ordinate system. Thus, the measurement noise from individual sensors is not taken into account properly.

This report proposes an exact maximum likelihood method for registration. The likelihood function is derived directly from the measurement data, which are assumed to contain two sets of parameters: the true target positions and the system errors. These two sets of variables are found to be partially separable in the likelihood function, and thus a two-step recursive optimization procedure is possible. An approximation to the Hessian is also made to ensure convergence of the optimization procedure. In the following sections,

we present the conventional formulation of the registration problem, the exact maximum likelihood formulation, the algorithm, computer simulation and real data analysis. Some conclusions are then drawn.

The Registration Problem

Consider two sensors A and B which measure the range and azimuth of a common target. Without loss of generality, let A be located at the origin and B at (u,v) as shown in Figure 1. Let T_k denote the k-th target, and let $\{r_A(k), \theta_A(k)\},\{r_B(k), \theta_B(k)\}$ represent the range and azimuth of the k-th target as measured by A and B respectively. The corresponding sensor biases (system errors) are denoted by $\{\Delta r_A, \Delta\theta_A\}$ and $\{\Delta r_B, \Delta\theta_B\}$. If $\{x'(k), y'(k)\}$ is the true position for the target T_k, we have,

$$x'(k) = [r_A(k) - \Delta r_A] \sin[\theta_A(k) - \Delta\theta_A]$$

$$y'(k) = [r_A(k) - \Delta r_A] \cos[\theta_A(k) - \Delta\theta_A], \qquad (1)$$

for sensor A, and

$$x'(k) = [r_B(k) - \Delta r_B] \sin[\theta_B(k) - \Delta\theta_B] + u$$

$$y'(k) = [r_B(k) - \Delta r_B] \cos[\theta_B(k) - \Delta\theta_B] + v, \qquad (2)$$

for sensor B. These two sets of equations can be combined and be represented in matrix form, giving us,

$$L(k)\,\underline{\eta} = \Delta\underline{X}(k) \qquad (3)$$

where $\underline{\eta} = [\Delta\theta_A, \Delta r_A, \Delta\theta_B, \Delta r_B]^T$,

$$L(k) = \begin{pmatrix} r_A(k)\cos\theta_A(k) & \sin\theta_A(k) & -r_B(k)\cos\theta_B(k) & -\sin\theta_B(k) \\ -r_A(k)\sin\theta_A(k) & \cos\theta_A(k) & r_B(k)\sin\theta_B(k) & -\cos\theta_B(k) \end{pmatrix} \qquad (4)$$

and $$\Delta\underline{X}(k) = \begin{pmatrix} r_A(k)\sin\theta_A(k) - r_B(k)\sin\theta_B(k) - u \\ r_A(k)\cos\theta_A(k) - r_B(k)\cos\theta_B(k) - v \end{pmatrix}. \qquad (5)$$

Equation (3) is the basic registration equation for all conventional methods. The only difference is the way $\underline{\eta}$ is solved. The RTQC routine separates the measurements into two groups by the site line joining the sensors A and B. It then averages the measurements in each group, and the registration error vector $\underline{\eta}$ can then be obtained by

solving a 4×4 matrix [8]. The LS method solves for $\underline{\eta}$ by finding the least squares solution based on K measurements. Thus, equation (3) becomes,

$$L\,\underline{\eta} = \Delta\underline{X} \tag{6}$$

where $L = [L(1)^T, L(2)^T, ..., L(K)^T]^T$, and $\Delta\underline{X} = [\Delta\underline{X}^T(1), \Delta\underline{X}^T(2), ..., \Delta\underline{X}^T(K)]^T$, and

$$\underline{\eta}_{LS} = (L^TL)^{-1}L^T\,\Delta\underline{X}\,. \tag{7}$$

The LS solution is based on the assumption that all measurements are equally weighted [10]. The ML and the GLS methods are introduced to incorporate the covariances of the measurement noise to weight the measurements. Since these two methods do not seem to have much significant improvement over the LS technique, especially in the real-life applications [11], our comparisons will be made with the LS method only.

The Exact Maximum Likelihood Method

The limitations of the conventional formulation for the registration errors as shown by equation (3) are evident when measurement noise are present in the sensors. The formulation, whether solved by simple averaging or least squares, assumes no noise in the measured data, as shown in equations (1) and (2). Any subsequent correction using the computed biases in range and azimuth will further enhance the random error in the measurements. This explains the limited success conventional techniques had with registration.

In the EML method, the random measurement noise of the sensors are included in the formulation. Let $\{x_A(k), y_A(k)\}$ and $\{x_B(k), y_B(k)\}$ denote the Cartesian co-ordinates of the target T_k in the system plane as measured by sensors A and B respectively. The actual range and azimuth measured by A and B are $\{r'_A(k), \theta'_A(k)\}$ and $\{r'_B(k), \theta'_B(k)\}$. Then using $n_i(k)$ as the random noise components, we have

$$x_A(k) = [r'_A(k) + \Delta r_A] \sin[\theta'_A(k)+\Delta\theta_A] + n_1(k)$$

$$y_A(k) = [r'_A(k) + \Delta r_A] \cos[\theta'_A(k)+\Delta\theta_A] + n_2(k), \tag{8}$$

and

$$x_B(k) = [r'_B(k) + \Delta r_B] \sin[\theta'_B(k)+\Delta\theta_B] + u + n_3(k)$$

$$y_B(k) = [r'_B(k) + \Delta r_B] \cos[\theta'_B(k)+\Delta\theta_B] + v + n_4(k). \tag{9}$$

Generally, the system biases $\Delta\theta_A$, Δr_A, $\Delta\theta_B$, Δr_B are small compared to measured values. The equations can be well approximated by retaining first order terms in the biases. Thus,

$$x_A(k) \cong [r'_A(k) + \Delta r_A]\sin\theta'_A(k) + r'_A(k)\cos\theta'_A(k)\Delta\theta_A + n_1(k),$$

$$y_A(k) \cong [r'_A(k) + \Delta r_A]\cos\theta'_A(k) - r'_A(k)\sin\theta'_A(k)\Delta\theta_A + n_2(k),$$

$$x_B(k) \cong [r'_B(k) + \Delta r_B]\sin\theta'_B(k) + r'_B(k)\cos\theta'_B(k)\Delta\theta_B + u + n_3(k),$$

$$y_B(k) \cong [r'_A(k) + \Delta r_A]\cos\theta'_B(k) - r'_B(k)\sin\theta'_B(k)\Delta\theta_B + v + n_4(k). \qquad (10)$$

Now when $\{x'(k), y'(k)\}$ is the actual position of T_k in the system plane, we have

$$x'(k) = r'_A(k)\sin\theta'_A(k) = r'_B(k)\sin\theta'_B(k) + u,$$

$$y'(k) = r'_A(k)\cos\theta'_A(k) = r'_B(k)\cos\theta'_B(k) + v. \qquad (11)$$

These can be substituted into (10) giving,

$$x_A(k) = x'(k) + \frac{\Delta r_A}{r'_A(k)}x'(k) + \Delta\theta_A y'(k) + n_1(k),$$

$$y_A(k) = y'(k) + \frac{\Delta r_A}{r'_A(k)}y'(k) - \Delta\theta_A x'(k) + n_2(k),$$

$$x_B(k) = x'(k) + \frac{\Delta r_B}{r'_B(k)}x'(k) + \Delta\theta_B y'(k) - \frac{\Delta r_B}{r'_B(k)}u - v\,\Delta\theta_B + n_3(k),$$

$$y_B(k) = y'(k) + \frac{\Delta r_B}{r'_B(k)}y'(k) + \Delta\theta_B x'(k) - \frac{\Delta r_B}{r'_B(k)}v + u\,\Delta\theta_B + n_4(k). \qquad (12)$$

In matrix form, equation (12) becomes,

$$\underline{x}(k) = A(k)\,\underline{\eta} + \underline{b}(k) + \underline{n}(k) \qquad (13)$$

where

$$\underline{x}(k) = [x_A(k), y_A(k), x_B(k), y_B(k)]^T,$$

$$\underline{b}(k) = [x'(k), y'(k), x'(k), y'(k)]^T,$$

$$\underline{n}(k) = [n_1(k), n_2(k), n_3(k), n_4(k)]^T,$$

$$\underline{\eta} = [\Delta\theta_A, \Delta r_A, \Delta\theta_B, \Delta r_B]^T, \tag{14}$$

and the matrix A(k) is a block diagonal matrix given by $A(k) = \text{diag}[A_{11}(k), A_{22}(k)]$, where the blocks are defined as

$$A_{11}(k) = \begin{pmatrix} y'(k) & \dfrac{x'(k)}{r'_A(k)} \\ -x'(k) & \dfrac{y'(k)}{r'_A(k)} \end{pmatrix} \quad \text{and} \quad A_{22}(k) = \begin{pmatrix} y'(k)-v & \dfrac{x'(k)-u}{r'_B(k)} \\ -x'(k)+u & \dfrac{y'(k)-v}{r'_B(k)} \end{pmatrix} \tag{15}$$

$\underline{n}(k)$ is the random measurement noise vector with a covariance matrix given by $\sigma_n^2 I$. $\underline{x}(k)$, $\underline{\eta}$ are the target measurement vector and the system bias vector for the target T_k. The matrix A(k) and the vector $\underline{b}(k)$ are independent of the system biases and are determined by the actual location of the target in the system plane.

Assuming $\{\underline{n}(k)\}$ to be independently and normally distributed, the conditional probability density function of the measurements $\{\underline{x}(k), k=1,2,....,K\}$ is,

$$p([\underline{x}(1), \underline{x}(2), \underline{x}(3), ..., \underline{x}(K)])$$

$$= \prod_{k=1}^{K} \frac{1}{(2\pi)^2 \sigma_n^4} \exp\left\{-\frac{1}{2\sigma_n^2} [\underline{x}(k) - A(k)\underline{\eta} - \underline{b}(k)]^T [\underline{x}(k) - A(k)\underline{\eta} - \underline{b}(k)]\right\} \tag{16}$$

The corresponding negative log likelihood function, without the constant factor has the form,

$$J = -\log p = 2K \log(2\pi\sigma_n^2) + \frac{1}{2\sigma_n^2} \sum_{k=1}^{K} \| \underline{x}(k) - A(k)\underline{\eta} - \underline{b}(k) \|_F^2. \tag{17}$$

where $\| \cdot \|_F$ denotes the Frobenius norm. Based on the principle of maximum likelihood estimation, the estimates of the unknown parameters are obtained by the minimizing arguments of (17) [12]. Let $\underline{\xi}(k) = [x'(k), y'(k)]^T$ be the actual position vector of the target in the system plane. Fixing $\underline{\xi}(k)$ and $\underline{\eta}$, while minimizing J with respect to σ_n^2 gives the estimate for the noise variance:

$$\hat{\sigma}_n^2 = \frac{1}{4K} \sum_{k=1}^{K} \| \underline{x}(k) - A(k)\underline{\eta} - \underline{b}(k) \|_F^2. \tag{18}$$

Substituting this back into (17), the estimates for $\underline{\xi}(k)$ and $\underline{\eta}$ are obtained as the minimizing arguments for the function J:

$$[\hat{\underline{\xi}}(k), \hat{\underline{\eta}}] = \arg\min J, \tag{19}$$

where

$$J = \frac{1}{K} \sum_{k=1}^{K} \| \underline{x}(k) - A(k)\underline{\eta} - \underline{b}(k) \|_F^2. \tag{20}$$

In general, J is a nonlinear function of the two parameter vectors. However, it is seen that the parameters $\underline{\xi}(k)$ and $\underline{\eta}$ are separable in (20), and an alternating optimization technique can be applied [13]. This technique iterates between two steps, each minimizing one of the parameter vectors until convergence is reached.

The Optimization Procedure

Estimation of the system bias vector

The bias vector $\underline{\eta}$ is first estimated while the true target position vector $\underline{\xi}(k)$ is fixed. This means that the estimate is the solution of,

$$\frac{\partial J}{\partial \underline{\eta}} = -2 \sum_{k=1}^{K} A^T(k)[\underline{x}(k) - A(k)\underline{\eta} - \underline{b}(k)] = 0. \tag{21}$$

Solving (21), the estimate of the bias vector is given by,

$$\hat{\underline{\eta}} = \{ \sum_{k=1}^{K} A^T(k)A(k) \}^{-1} \sum_{k=1}^{K} A^T(k)[\underline{x}(k) - \underline{b}(k)]. \tag{22}$$

Estimating the true target position vector

Using the estimate in (22) the true target position vector can be calculated as the minimizing arguments of the likelihood function J. It can be seen that since J is a sum over non-negative terms, each of which relates to a specific k, the minimization can be performed term-by-term so that we obtain,

$$\hat{\underline{\xi}}(k) = \arg\min J_k = \arg\min \| \underline{x}(k) - A(k)\hat{\underline{\eta}} - \underline{b}(k) \|_F^2. \tag{23}$$

The non-linear problem in (23) can be solved numerically by using a Gauss-Newton type method [13]. The method can be represented by the iterative equation,

$$\hat{\underline{\xi}}^{(p+1)}(k) = \hat{\underline{\xi}}^{(p)}(k) - \mu_p H_k^{-1} G_k \tag{24}$$

where μ_p is the p-th iteration step length, H_k is the Hessian matrix for the function J_k with respect to the parameters to be estimated, and G_k is the gradient given by,

$$G_k = 2\, R_k\, [\underline{x}(k) - \underline{b}(k)] = 2R_k\, \underline{\nu}(k). \tag{25}$$

R_k is the Jacobian matrix of $\underline{\nu}(k)$ with respect to the target position vectors $\underline{\xi}(k)$ and can be obtained as,

$$R_k = \begin{pmatrix} -1 - \dfrac{\Delta r_A y'^2(k)}{r'^3_A(k)} & -\Delta\theta_A + \dfrac{\Delta r_A x'(k) y'(k)}{r'^3_A(k)} \\ \Delta\theta_A + \dfrac{\Delta r_A x'(k) y'(k)}{r'^3_A(k)} & -1 - \dfrac{\Delta r_A x'^2(k)}{r'^3_A(k)} \\ -1 - \dfrac{\Delta r_B [y'(k)-v]^2}{r'^3_B(k)} & -\Delta\theta_B + \dfrac{\Delta r_B [x'(k)-u][y'(k)-v]}{r'^3_B(k)} \\ \Delta\theta_B + \dfrac{\Delta r_B [x'(k)-u][y'(k)-v]}{r'^3_B(k)} & -1 - \dfrac{\Delta r_B [x'(k)-u]^2}{r'^3_B(k)} \end{pmatrix}^T \tag{26}$$

The Hessian of J_k can be computed as its second order derivative with respect to the parameters to be estimated. Its role is to modify the gradient in order to achieve faster convergence. At each iteration, the Hessian must be positive definite to guarantee convergence. In practice, the true second order derivative is not guaranteed to be positive definite at each iteration. To avoid non-convergence, the Hessian is approximated by

$$H_k = 2\, R_k R_k^T. \tag{27}$$

Thus the Hessian will be a positive semi-definite matrix. This approximation is justified by observing that the system errors are generally much smaller than the range measurements of the target.

The EML registration algorithm can be summarized as follows.

- Set a threshold ε and set the iteration index p to zero.
- Obtain an initial estimate of the true target position vectors $\hat{\underline{\xi}}^{(p)}(k)$ for k = 1,2,..., K, and compute the estimate $\hat{\underline{\eta}}^{(p)}$ according to (22).

- Minimize the function J_k , using the approximate Hessian, to obtain the (p+1)-th estimate $\hat{\xi}^{(p+1)}(k)$.

- Estimate the (p+1)-th system bias vector $\hat{\eta}^{(p+1)}$ using (22). Compute the difference norm, $d = \|\hat{\eta}^{(p+1)} - \hat{\eta}^{(p)}\|^2$ and check it against the threshold ε. Repeat from step 3 if $d > \varepsilon$ and set index p to p+1; otherwise stop.

We note that the last two steps can be interchanged in the sense that the threshold can be applied either to the system bias vector or the true target position vector. With the Gauss-Newton method, where the approximated Hessian is used, each separate estimation of the system bias vector and of the target position vector will not increase the objective function J_k . Thus, although convergence to a global minimum is not guaranteed the algorithm is seen to be convergent.

The algorithm can be further simplified, since the matrix A(k) is block diagonal. This means the size of the matrices to be inverted is reduced by a factor of 1/2.

Simulation and real data analysis

In the simulation studies, we examine the performance and robustness of the EML method using track patterns shown in Figure 2. Sensor A is located at the origin and sensor B at (u,v), where u=300 nm (nautical miles) and v=0 nm, in the system plane. Each site is assumed to have a range bias of 1 nm and azimuth bias of 0.0087 radians. The track patterns shown represent typical flight tracks that may be encountered by two radars in a practical situation. In track pattern (a), the target measurements are distributed on both sides of the site line joining sensors A and B, and the track has a slope of 1 relative to the site line. Track (b) is simulated to be parallel to the line connecting the two sites and the measurements are on only one side of the site line. The distance between the track and the site line, d, is set to be 120 nm.

The means and normalized MSE's for the EML and the LS estimates are computed under different noise assumptions. The deviation of the measurement noise varies from 0 to 2 nm. Performance is evaluated through Monte-Carlo method and each test is repeated 100 times to obtain the averaged results. From Tables 1 and 2, we observe that both the EML and the LS estimates are unbiased. Figures 3 and 4 show the variations of the normalized MSE of the registration error estimates via the measurement noise deviation. They clearly demonstrate the improved performance of the EML estimates over the LS estimates, especially for target pattern (b) where the EML algorithm maintains a good estimation performance even in the presence of strong measurement noise while the LS method performs poorly and fails as the measurement noise becomes large.

Figure 5 and 6 show the variations of the normalized MSE's of the registration estimates versus the number of measurements. The deviation of the measurement noise σ_n is set to 0.5 nm, and the number of measurements varies from 10 to 60 with a step size of 10. Again, for each measurement size, 100 trials are repeated to obtain the average. Both EML and LS estimates converge to the true parameter values as the number of measurement increases. However, for small number of measurements, the EML algorithm has a much smaller MSE than the LS method.

The distance between the measurements form different sensors before and after registration is another index for measuring the quality of the registration algorithm. In Table 3, we compute the distances between the original tracks and between the updated tracks using both the EML and the LS methods against the measurement noise deviations. Apparently, both methods can successfully reduce the distance between the updated tracks from different sensors after registration. The EML method outperforms the LS method in the sense that it produces smaller distances between the updated tracks from different sensors for all levels of measurement noise deviations. It should be mentioned that, unlike the LS algorithm, the EML method provides optimal estimates of the target tracks simultaneously with the registration estimates.

To understand the efficiency of the EML method in a practical environment, real-life multiple radar data are collected from an air surveillance network. The radar network is composed of seventeen 24-by-24 foot phased array L-band long range radars located along the coastline of Canada. Tracks of air targets arise from commercial flights. The operating specification of the radars are summarized as follows:

- operating frequency: 1215-1400 Mhz.
- instrument range: 5 - 200 nm
- azimuth coverage: 120 degrees in 12 seconds
- range resolution: 300 meters
- azimuth beamwidth: 2.2 degrees
- probability of detection: 0.75

To remove the effects of false targets (clutters), target ID's provided by the identification of friend and foe (IFF) beacon are used to extract the true aircraft tracks from radar returns for registration calculation. Each site in the system has two satellite dishes for communications, and the communications between the radars are via the Anik satellite. The radar network employs the stereographic projection [14,15] to map the

elliptic earth onto a plane to get the ground range from the slant range of a target. The target azimuth is measured relative to the true north at the radar location and is adjusted so that it is relative to the true north at the origin of the common co-ordinate system. For the data set used in this study, the location of radar A is transformed to the origin and hence radar B is located at (u,v)=(272.6406, 19.3287) nm, as shown in Figure 7.

Table 4 shows the registration estimates obtained using the two methods. The number of measurements varies from 15 to 55 in steps of 10. It is shown that the EML estimates are more consistent than the LS estimates when different number of measurement is used, especially when this number is small. The EML estimates for K=15 are close to those for K=55, while the LS method shows a drastic change in its estimates. This implies that the EML method is capable of satisfactory registration with a relatively small number of measurements. For K=15, the distance between the original tracks from different sensors is 0.7223 nm. After registration using EML, the distance between the updated tracks becomes 0.0970 nm while that for the LS method is 0.6635 nm.

Another important measure of registration algorithm is its generalization capability. More precisely, the registration estimates should also work for those measurement data not used in estimating the registration errors. To evaluate this aspect of the two methods, we use the first 15 measurements to compute the system biases. These estimates are then used in the rest of the data set to see whether the distance between the updated tracks can be reduced. Figure 8 shows the distances between the updated tracks versus the number of measurements. For the EML method, the distances remain at the same level as that for the first 15 measurements. However, when the LS estimates are used, the generalization gets worse as the number of measurements increases and the distances between the updated tracks become larger than that of the original tracks.

Conclusions

A robust registration algorithm called the exact maximum likelihood (EML) method for multiple sensor data fusion is developed in this report. The EML algorithm uses the likelihood function based on direct measurements. A two-step optimization procedure is proposed to estimate the true target positions and the system biases iteratively. Computer simulation and real data analysis show that the EML method outperforms the LS registration algorithm. It can provide satisfactory registration accuracies with a small number of measurements, and achieve consistent registration error estimates under different target distribution patterns. Although the trade-off for this improvement is a higher computational cost, it does not pose any serious problem for registration applications since registration is an off-line pre-processing procedure for a data fusion system.

References

[1] D.L. Hall, *Mathematical Techniques in Multisensor Data Fusion,* Artech House, 1992.

[2] M.A. Abidi and R.C. Gonzalez, ed., *Data Fusion in Robotics and Machine Intelligence,* Academic Press, 1992.

[3] A.D. Stoyenko, P.A. Laplante, R. Harrison and T.J. Marlowe, "Doubling the Engineer's Utility", IEEE Spectrum, pp.32-39, Dec. 1994.

[4] D.D. Freedman and P.A. Smyton, "Overview of Data Fusion Activities", Proc. of the American Control Conf., WP9-14:30, San Francisco, June 1993.

[5] M.P. Dana, "Registration: A pre-requisite for multiple sensor tracking", in Y. Bar-Shalom (ed.), *Multitarget-multisensor Tracking: Advanced Applications,* Artech House, 1990.

[6] Y. Bresler, S.J. Merhav, "Recursive Image Registration with Application to Motion Estimation", IEEE Trans. ASSP-35, pp 70-85, 1987.

[7] S.C.A. Thomopoulos, N.N. Okello, "Distributed and Centralized Multisensor Detection with Misaligned Sensors", Information Sciences, Vol. 77, pp 293-323, 1994.

[8] J.J. Burke, "The SAGE Real Quality Control Fraction and its Interface with BUIC II/BUIC III" MITRE Corporation Technical Report, No. 308, Nov. 1966.

[9] H. Leung, M. Blanchette, C. Harrison, "A Least Squares Fusion of Multiple Radar Data", Proc. RADAR 94, pp 364-369, Paris, 1994.

[10] B.G. Sokkappa, "Registration Computation for RTQC", MITRE Working Paper 7681, Feb. 1971.

[11] H. Leung, M. Blanchette, K. Gault, "Comparison of Registration Error Correction Techniques for Air Surveillance Radar Network", SPIE Proc. on Signal and Data Processing of Small Targets, pp 495-508, San Diego 1995.

[12] T.W. Anderson, *An Introduction to Multivariate Statistical Analysis,* 2nd ed., New York: John Wiley and Sons, 1984.

[13] J.E. Dennisand, R.B. Schnabel, *Numerical Methods for Unconstrained Optimization and Non-linear Equations,* Englewood Cliffs, NJ: Prentice-Hall, 1983.

[14] R.G. Mulholland and D.W. Stout, "Stereographic Projection in the National Airspace System", IEEE Trans. AES-18, pp 48-57, 1982.

[15] K.H. Kim, P.A. Smyton, "Stereographic Projection in Netted Radar System", MITRE Corporation Technical Report No. 10296, May 1988.

Figures and tables

Table 1: **Mean of the registration estimates for flight track (a)**

	EML				LS			
σ_n	$\Delta\theta_A$	Δr_A	$\Delta\theta_B$	Δr_B	$\Delta\theta_A$	Δr_A	$\Delta\theta_B$	Δr_B
2.0	0.0105	1.1517	0.0068	0.8087	0.0090	1.0042	0.0079	1.0232
1.8	0.0070	0.9484	0.0100	1.0300	0.0078	1.1207	0.0089	0.9144
1.6	0.0117	1.0492	0.0066	0.8988	0.0084	0.7027	0.0096	1.2980
1.4	0.0089	0.8861	0.0088	0.9643	0.0082	0.7965	0.0092	1.1004
1.2	0.0084	0.8543	0.0088	1.0838	0.0086	1.0502	0.0083	0.9232
1.0	0.0075	0.8473	0.0101	1.1293	0.0083	1.0116	0.0091	0.9974
0.8	0.0094	1.1117	0.0084	0.8612	0.0084	1.0221	0.0091	0.9802
0.6	0.0090	1.0727	0.0086	0.8840	0.0091	1.1469	0.0083	0.8362
0.4	0.0091	1.0857	0.0086	0.9275	0.0088	1.0765	0.0086	0.9618
0.2	0.0090	1.0252	0.0085	0.9757	0.0088	0.9862	0.0086	1.0389
0.0	0.0087	1.0000	0.0087	1.0000	0.0087	1.0000	0.0087	1.0000

Table 2: **Mean of the registration estimates for flight track (b)**

	EML				LS			
σ_n	$\Delta\theta_A$	Δr_A	$\Delta\theta_B$	Δr_B	$\Delta\theta_A$	Δr_A	$\Delta\theta_B$	Δr_B
2.0	0.0056	1.5598	0.0121	1.4078	0.0023	2.4367	0.0169	1.8839
1.8	0.0017	1.9003	0.0146	2.0610	-0.0013	2.2074	0.0164	2.6112
1.6	0.0060	1.2538	0.0112	1.5121	0.0049	1.2035	0.0109	1.6879
1.4	0.0075	0.9480	0.0092	1.1438	0.0044	1.1989	0.0109	1.6879
1.2	0.0086	0.7097	0.0073	1.0020	0.0053	1.5171	0.0119	1.4776
1.0	0.0092	0.6456	0.0070	0.9788	0.0048	1.1887	0.0105	1.7124
0.8	0.0092	0.6873	0.0073	0.9727	0.0061	1.1769	0.0102	1.4752
0.6	0.0090	0.7451	0.0076	0.9787	0.0079	1.4986	0.0114	1.0370
0.4	0.0089	0.8183	0.0078	1.0190	0.0081	1.2191	0.0098	1.1019
0.2	0.0089	0.8543	0.0080	1.0051	0.0093	1.0329	0.0086	0.8972
0.0	0.0087	1.0000	0.0087	1.0000	0.0087	1.0000	0.0087	1.0000

Table 3: **Calculation of the distance between the updated tracks**

	TRACK (a)		TRACK (b)	
σ_n	EML	LS	EML	LS
2.0	0.6004	0.8629	0.6142	0.9454
1.8	0.5409	0.8210	0.4851	0.8318
1.6	0.4923	0.7412	0.4505	0.7702
1.4	0.3825	0.6267	0.3925	0.6601
1.2	0.3177	0.5091	0.2988	0.5423
1.0	0.2508	0.4273	0.2401	0.4605
0.8	0.1928	0.3469	0.1811	0.3683
0.6	0.1348	0.2619	0.1354	0.2700
0.4	0.0879	0.1721	0.0888	0.1818
0.2	0.0434	0.0912	0.0440	0.0947
0.0	0.0000	0.0000	0.0000	0.0000

Table 4: **Registration results using real radar data**

	EML				LS			
K	$\Delta\theta_A$	Δr_A	$\Delta\theta_B$	Δr_B	$\Delta\theta_A$	Δr_A	$\Delta\theta_B$	Δr_B
15	-0.0046	0.0594	-0.0208	-0.2895	-0.0150	62.6223	0.0577	-62.5132
25	-0.0069	0.0504	-0.0163	-0.2177	-0.0015	6.4926	-0.0148	-6.6899
35	-0.0059	-0.0158	-0.0184	-0.2116	-0.0032	1.8650	-0.0196	-2.0925
45	-0.0064	-0.0360	-0.0177	-0.2035	-0.0029	2.9533	-0.0185	-3.1937
55	-0.0062	-0.0675	-0.0180	-0.1993	-0.0043	1.5275	-0.0184	-1.7672

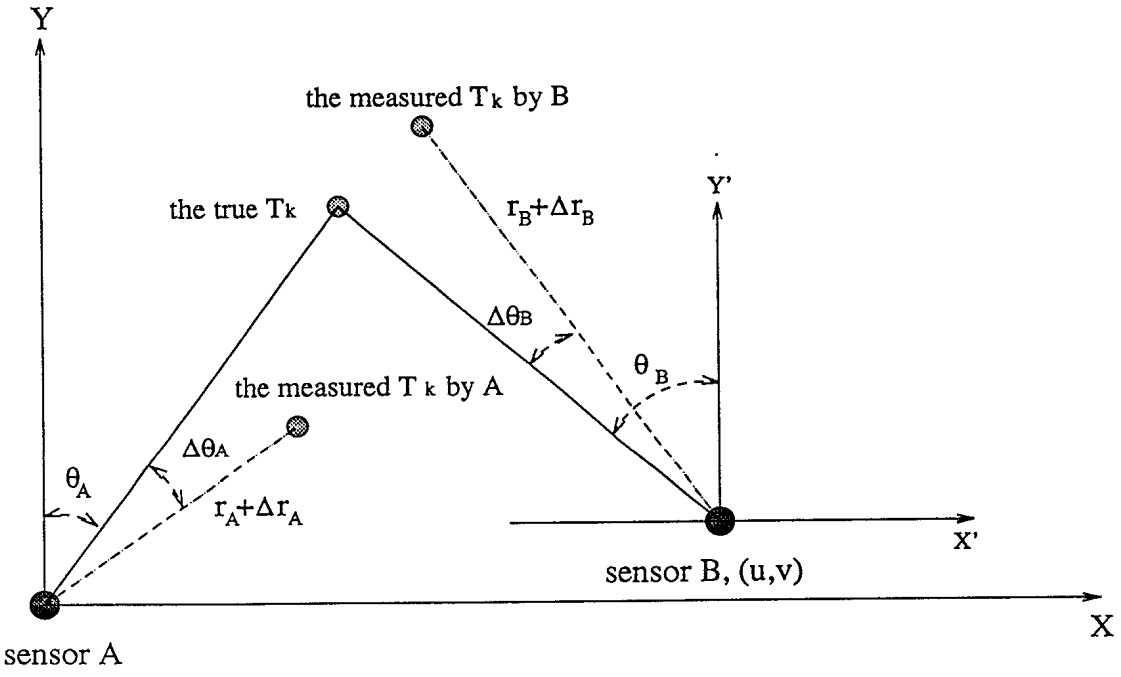
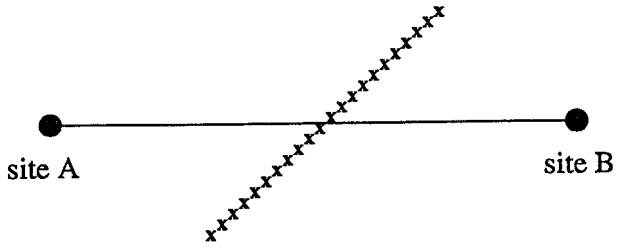
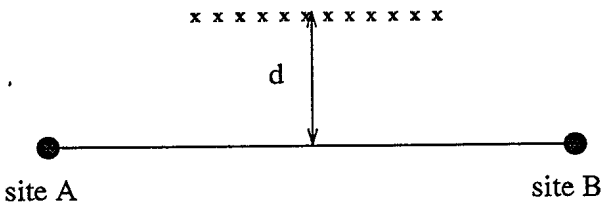


Figure 1: The geometry of registration errors



track pattern (a)



track pattern (b)

Figure 2: The simulated flight track patterns for registration.

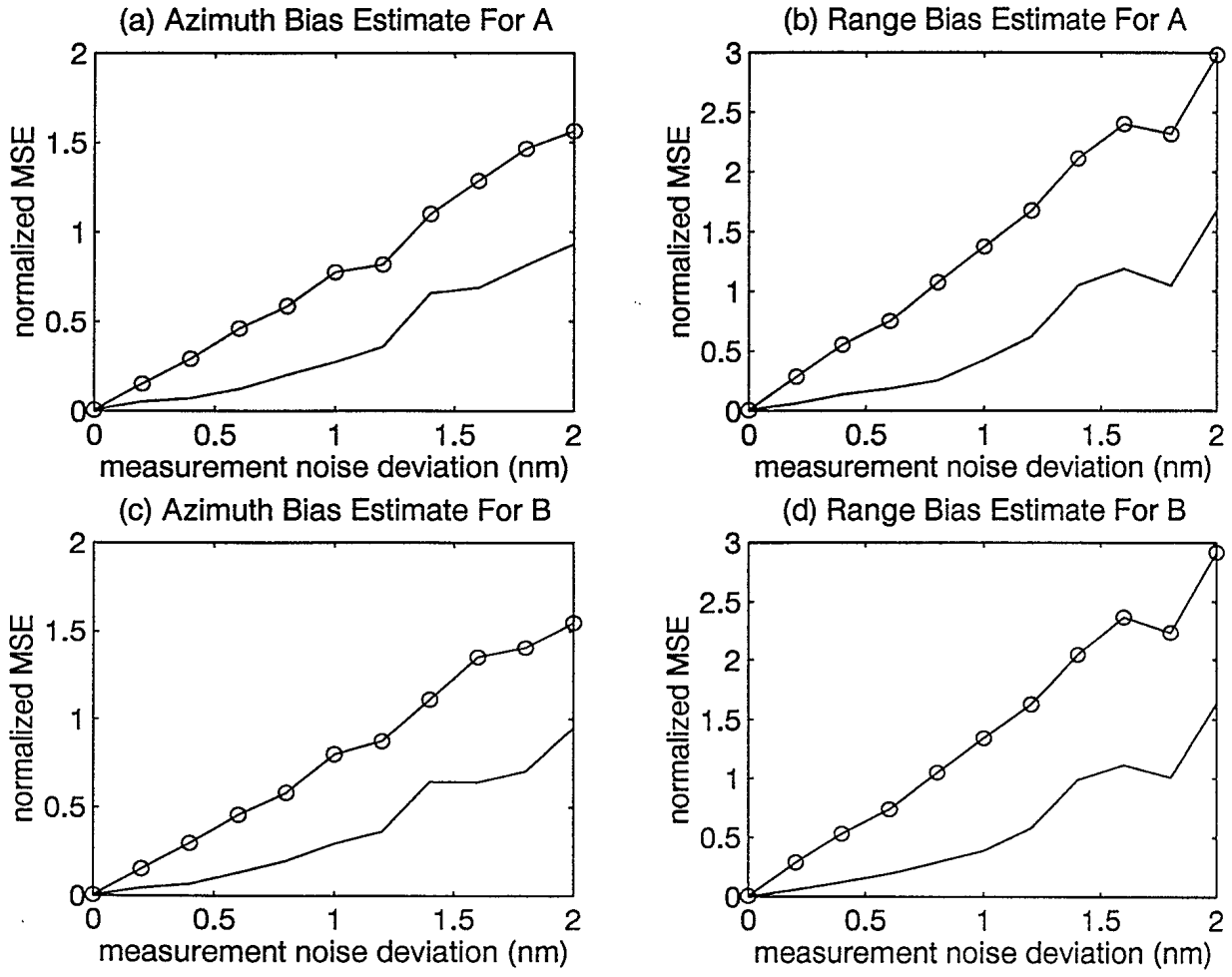


Figure 3: The variation of the normalised MSE's of the registration estimates versus the measurement noise deviation for track pattern (a). The solid line represents the EML results, and the line labeled 'o' denotes the results obtained by the LS method.

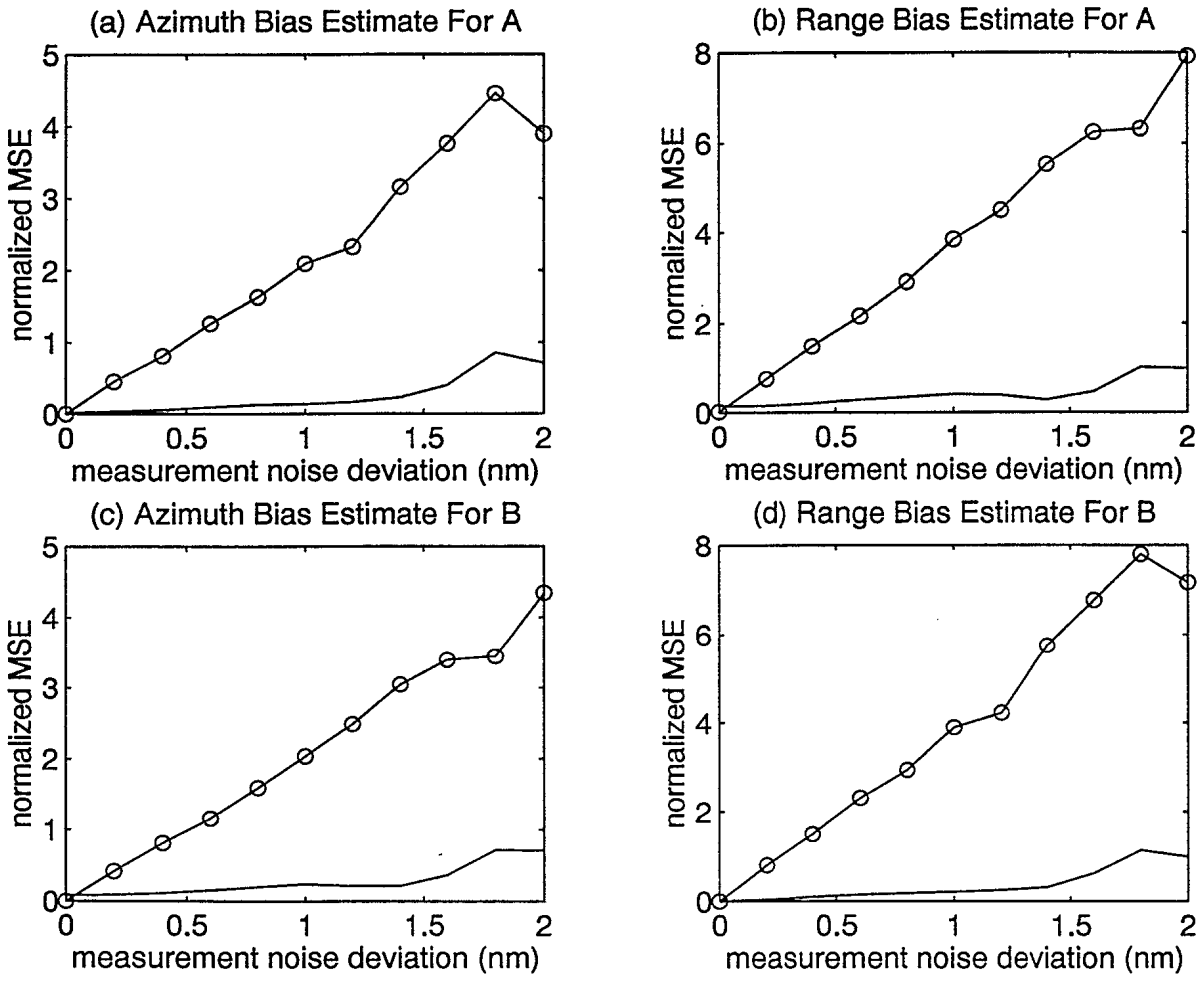


Figure 4: The variation of the normalised MSE's of the registration estimates versus the measurement noise deviation for track pattern (b). The solid line represents the EML results, and the line labeled 'o' denotes the results obtained by the LS method.

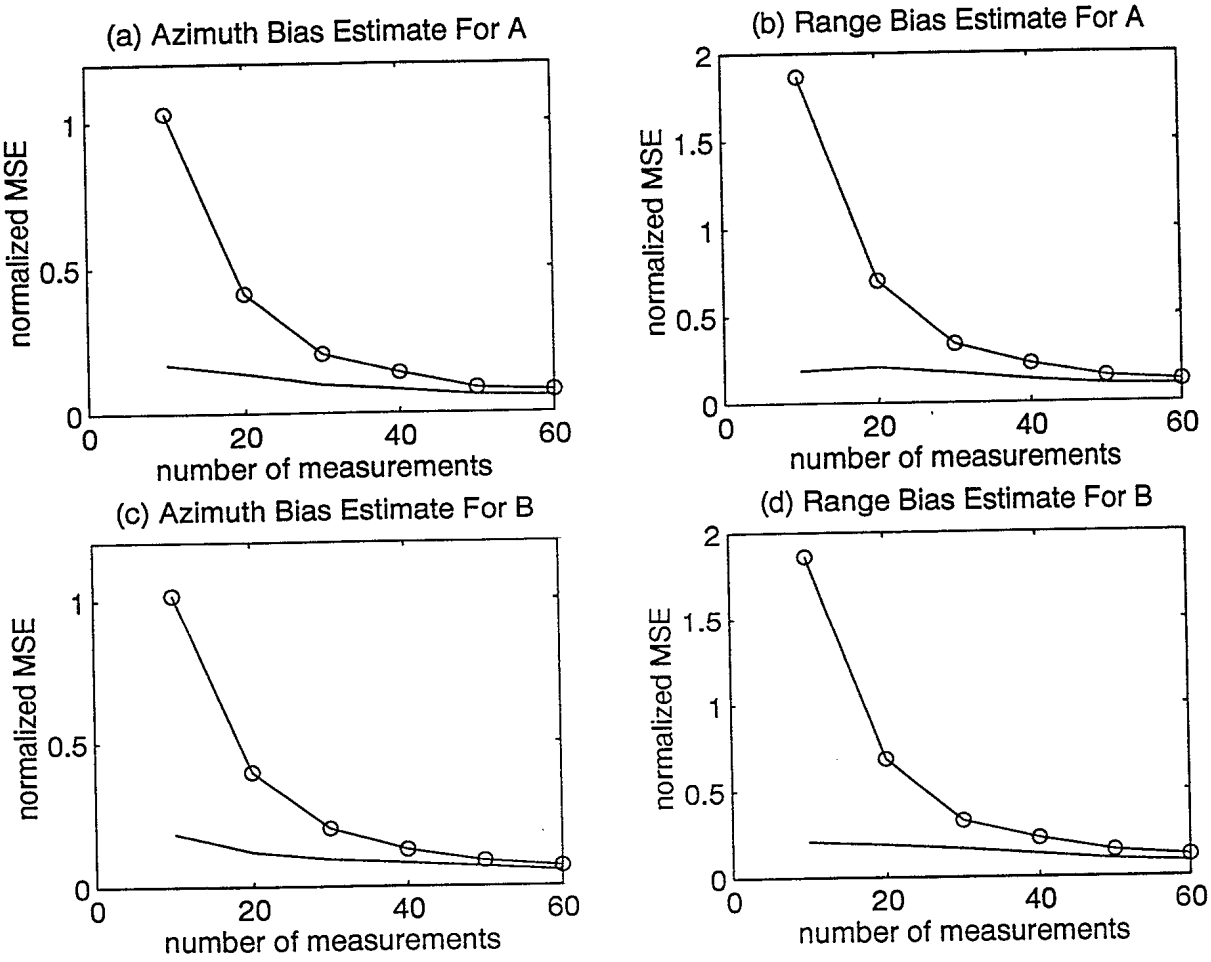


Figure 5: The variation of the normalised MSE's of the registration estimates versus the number of measurement for track pattern (a). The solid line represents the EML results, and the line labeled 'o' denotes the LS results.

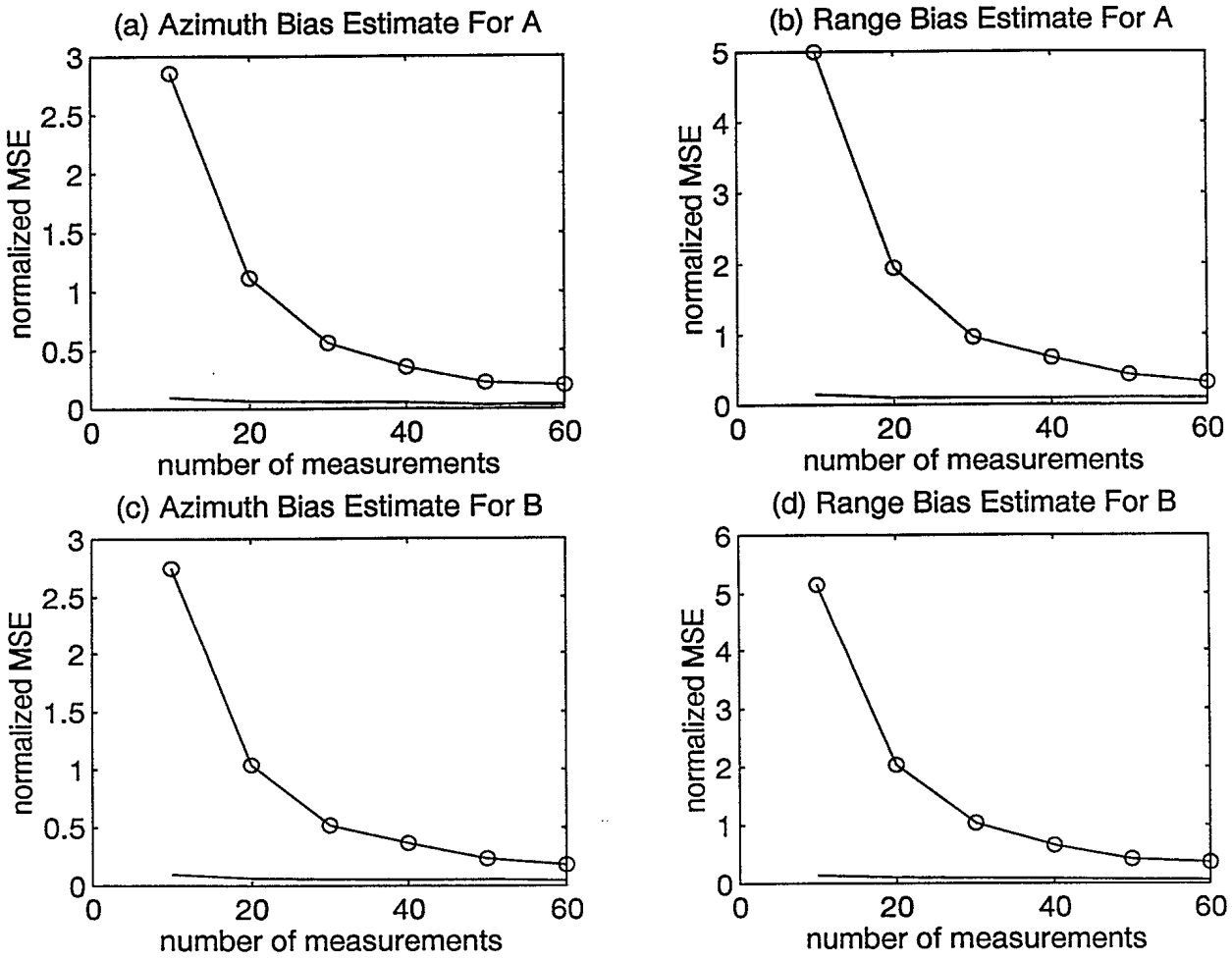


Figure 6: The variation of the normalised MSE's of the registration estimates versus the number of measurement for track pattern (b). The solid line represents the EML results, and the line labeled 'o' denotes the LS results.

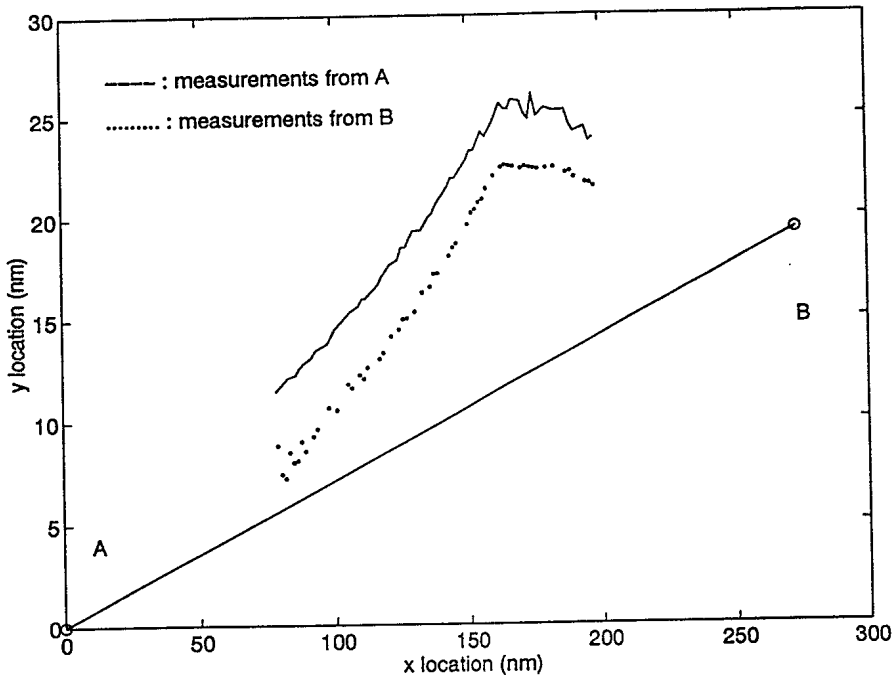


Figure 7: The target flight track measured by radar A and B in the system plane.

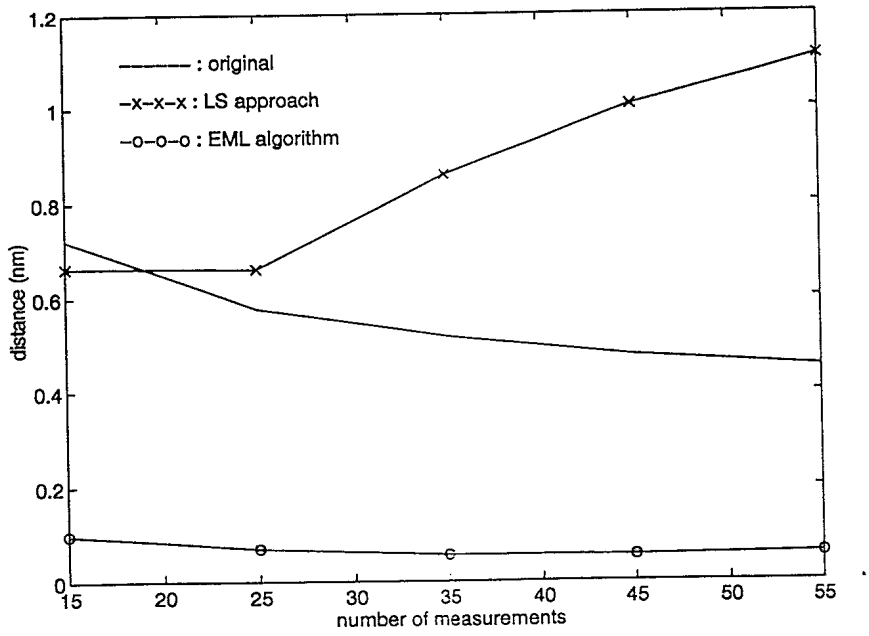


Figure 8: The distances between the updated tracks based on the bias estimates using 15 measurements.

Appendix: Matlab Program Listing

rgdata1.m

```
% This function file rgdata1.m generates the simulated target data. The
% returns are azimuth and range with respect to site A and B, respectively.
% The inputs are site B coordinate, the slope, number of target K, the step
% size, the pattern number and myv is the y functin value for pattern 3 and
% 4.

%                    Yifeng Zhou, McMaster University, Jan. 1995

function rgm=rgdata1(u,K,st)

rgm=zeros(4,K);
mta=zeros(1,K);
mra=zeros(1,K);
mtb=zeros(1,K);
mrb=zeros(1,K);
my=zeros(1,K);

mx=u*ones(1,K)/2;
mk=0:K-1;
my=(mk-(K-1)/2)*st;

thrd=10e-4;

for k=1:K
 mra(k)=sqrt(mx(k)*mx(k)+my(k)*my(k));
 mrb(k)=sqrt((mx(k)-u)*(mx(k)-u)+my(k)*my(k));

 if abs(my(k))<thrd
  mta(k)=pi/2;
  mtb(k)=-pi/2;
 else
  if my(k) >= 0
   mta(k)=atan(mx(k)/my(k));
   mtb(k)=atan((mx(k)-u)/my(k));
  else
   mta(k)=-atan(my(k)/mx(k))+pi/2;
   mtb(k)=-atan(my(k)/(mx(k)-u))-pi/2;
  end
 end
end

rgm=[mta;mra;mtb;mrb];
```

rgdata2.m

```
% This function file rgdata2.m generates the simulated target data for
% pattern 2. The returns are azimuth and range with respect to site A
% and B, respectively. The inputs are site B coordinate, the slope,
% number of target K, the step size.

%                    Yifeng Zhou, McMaster University, Jan. 1995

function rgm=rgdata2(u,kr,K,st)

rgm=zeros(4,K);
mta=zeros(1,K);
mra=zeros(1,K);
mtb=zeros(1,K);
mrb=zeros(1,K);
mx=zeros(1,K);
my=zeros(1,K);

mk=1:K;
mx=(mk-1-(K-1)/2)*st+u/2;
my=kr*(mx-u/2);

thrd=10e-4;

for k=1:K
 mra(k)=sqrt(mx(k)*mx(k)+my(k)*my(k));
 mrb(k)=sqrt((mx(k)-u)*(mx(k)-u)+my(k)*my(k));

 if abs(my(k))<thrd
  mta(k)=pi/2;
  mtb(k)=-pi/2;
 else
  if my(k) >= 0
   mta(k)=atan(mx(k)/my(k));
   mtb(k)=atan((mx(k)-u)/my(k));
  else
   mta(k)=-atan(my(k)/mx(k))+pi/2;
   mtb(k)=-atan(my(k)/(mx(k)-u))-pi/2;
  end
 end
end

rgm=[mta;mra;mtb;mrb];
```

rgdata3.m

```
% This function file rgdata3.m generates the simulated target data for
% pattern 3. The returns are azimuth and range with respect to site A
% and B, respectively. The inputs are site B coordinate, the slope,
% number of target K, step size and myv is the y functin value.

%                    Yifeng Zhou, McMaster University, Jan. 1995

function rgm=rgdata3(u,K,st,myv)

rgm=zeros(4,K);
mta=zeros(1,K);
mra=zeros(1,K);
mtb=zeros(1,K);
mrb=zeros(1,K);
mx=zeros(1,K);
my=zeros(1,K);

mk=1:K;
mx=(mk-1-(K-1)/2)*st+u/2;
my=myv*ones(1,K);
thrd=10e-4;

for k=1:K
  mra(k)=sqrt(mx(k)*mx(k)+my(k)*my(k));
  mrb(k)=sqrt((mx(k)-u)*(mx(k)-u)+my(k)*my(k));

  if abs(my(k))<thrd
    mta(k)=pi/2;
    mtb(k)=-pi/2;
  else
    if my(k) >= 0
      mta(k)=atan(mx(k)/my(k));
      mtb(k)=atan((mx(k)-u)/my(k));
    else
      mta(k)=-atan(my(k)/mx(k))+pi/2;
      mtb(k)=-atan(my(k)/(mx(k)-u))-pi/2;
    end
  end
end

rgm=[mta;mra;mtb;mrb];
```

rgdata4.m

```
% This function file rgdata4.m generates the simulated target data for
% pattern 4. The returns are azimuth and range with respect to site A
% and B, respectively. % The inputs are site B coordinate, number of
% target K, the step size and the y function value myv.

%                    Yifeng Zhou, McMaster University, Jan. 1995

function rgm=rgdata4(u,K,st,myv)

rgm=zeros(4,K);
mta=zeros(1,K);
mra=zeros(1,K);
mtb=zeros(1,K);
mrb=zeros(1,K);
mx=zeros(1,K);
my=zeros(1,K);

HK=K/2;
mk=1:HK;
mx(1:HK)=(mk-1-(HK-1)/2)*st+u/2;
mx(HK+1:K)=mx(1:HK);
my(1:HK)=myv*ones(1,HK);
my(HK+1:K)=-myv*ones(1,K-HK);
thrd=10e-4;

for k=1:K
  mra(k)=sqrt(mx(k)*mx(k)+my(k)*my(k));
  mrb(k)=sqrt((mx(k)-u)*(mx(k)-u)+my(k)*my(k));

  if abs(my(k))<thrd
    mta(k)=pi/2;
    mtb(k)=-pi/2;
  else
    if my(k) >= 0
      mta(k)=atan(mx(k)/my(k));
      mtb(k)=atan((mx(k)-u)/my(k));
    else
      mta(k)=-atan(my(k)/mx(k))+pi/2;
      mtb(k)=-atan(my(k)/(mx(k)-u))-pi/2;
    end
  end
end

rgm=[mta;mra;mtb;mrb];
```

rgsam1.m

```
% This file rgsam1.m is is to compute the MSE of system error via the changes
% of number of measurements for the EML and the LS approaches.

%                    Yifeng Zhou,  McMaster University 1995

clear;

kn=10:10:60;
lkn=length(kn);

% dta, dra, dtb,drb denotes the system bias of site A and B
dta=0.5*pi/180;
dtb=0.5*pi/180;
dra=1.0;
drb=1.0;
eta0=[dta,dra,dtb,drb]';

% the unit of range measurement is nautical miles
u=300;
v=0;

cmeta=zeros(4,lkn);
lmeta=zeros(4,lkn);
emeta=zeros(4,lkn);

laeta=zeros(4,lkn);
eaeta=zeros(4,lkn);

eiter=zeros(1,lkn);

ldse=zeros(1,lkn);
edse=zeros(1,lkn);

TM=100;

thrd=1.0e-3;
mu0=0.8;
epsilone=0.02;
epsilonx=0.01;
tumax=21;
tu=zeros(1,tumax);
tmpxi=zeros(2,tumax);
mu=[mu0.^(1:tumax-1),0];
```

```
st=4;
mgv=120;
kr=1;

snr=0.5*dra;

%*************** compute the asymptotic covariance matrix ********
Rk=zeros(4,2);
tceta=zeros(4,4);
ceta=zeros(4,4);

for tn=1:lkn

  K=kn(tn);
  mta=zeros(1,K);
  mtb=zeros(1,K);
  mra=zeros(1,K);
  mrb=zeros(1,K);

  tm=rgdata1(u,K,st);
  %tm=rgdata2(u,kr,K,st);
  %tm=rgdata3(u,K,st,mgv);
  %tm=rgdata4(u,K,st,mgv);

  mta=tm(1,:);
  mra=tm(2,:);
  mtb=tm(3,:);
  mrb=tm(4,:);

  % simulate the biased xy coordinate with respect to site A and B
  txa=zeros(1,K);
  tya=zeros(1,K);
  txb=zeros(1,K);
  tyb=zeros(1,K);

  txa=(mra+dra).*sin(mta+dta);
  tya=(mra+dra).*cos(mta+dta);
  txb=(mrb+drb).*sin(mtb+dtb);
  tyb=(mrb+drb).*cos(mtb+dtb);

  for k=1:K
    alpha=mra(k)*sin(mta(k));
    beta=mra(k)*cos(mta(k));
    rda=sqrt(alpha^2+beta^2);
    rdb=sqrt((alpha-u)^2+(beta-v)^2);
```

```
 Rk(1,1)=-1-eta0(2)*beta^2/rda^3;
 Rk(2,1)=eta0(1)+eta0(2)*alpha*beta/rda^3;
 Rk(3,1)=-1-eta0(4)*(beta-v)^2/rdb^3;
 Rk(4,1)=eta0(3)+eta0(4)*(alpha-u)*(beta-v)/rdb^3;

 Rk(1,2)=-eta0(1)+eta0(2)*alpha*beta/rda^3;
 Rk(2,2)=-1-eta0(2)*alpha^2/rda^3;
 Rk(3,2)=-eta0(3)+eta0(4)*(alpha-u)*(beta-v)/rdb^3;
 Rk(4,2)=-1-eta0(4)*(alpha-u)^2/rdb^3;

 Ppk=eye(4,4)-Rk*inv(Rk'*Rk)*Rk';
 AA=zeros(4,4);
 AA(1:2,1:2)=[beta,alpha/rda;-alpha,beta/rda];
 AA(3:4,3:4)=[beta-v,(alpha-u)/rdb;-(alpha-u),(beta-v)/rdb];

 tceta=tceta+AA'*Ppk*AA;
end

ceta=inv(tceta);
cmeta(:,tn)=snr*snr*diag(ceta);

% ********* obtain the initial estimate on xi and eta ***********
txit=zeros(2,K);
xi=zeros(2,K);
xi(1,:)=mra.*sin(mta);
xi(2,:)=mra.*cos(mta);
xi0=xi;
tda=sqrt(xi(1,:).^2+xi(2,:).^2);
tdb=sqrt((xi(1,:)-u).^2+(xi(2,:)-v).^2);

for tms=1:TM

 % generate the simulated target data for different SNR
 xa=txa+snr*randn(1,K);
 ya=tya+snr*randn(1,K);
 xb=txb+snr*randn(1,K)+u;
 yb=tyb+snr*randn(1,K)+v;

 X=[xa;ya;xb;yb];
 XA=[xa;ya];
 XB=[xb;yb];

 % obtain the noise azimuth and range measurement
 for k=1:K
  ra(k)=sqrt(xa(k)*xa(k)+ya(k)*ya(k));
  rb(k)=sqrt((xb(k)-u)*(xb(k)-u)+(yb(k)-v)*(yb(k)-v));
```

```
 if abs(ya(k))<thrd
   ta(k)=pi/2;
 elseif ya(k)>=0
   ta(k)=atan(xa(k)/ya(k));
 else
   ta(k)=-atan(ya(k)/xa(k))+pi/2;
 end

 if abs(yb(k))<thrd
   tb(k)=-pi/2;
 elseif yb(k)>=0
   tb(k)=atan((xb(k)-u)/yb(k));
 else
   tb(k)=-atan(yb(k)/(xb(k)-u))-pi/2;
 end

end

% ***************** the least squares algorithm **********************
LA=zeros(2*K,4);
bl=zeros(2*K,1);

for k=1:K
  m1=2*(k-1)+1;
  m2=2*k;
  LA(m1,:)=[ra(k)*cos(ta(k)),sin(ta(k)),-rb(k)*cos(tb(k)),-sin(tb(k))];
  LA(m2,:)=[-ra(k)*sin(ta(k)),cos(ta(k)),rb(k)*sin(tb(k)),-cos(tb(k))];
  bl(m1,1)=xa(k)-xb(k);
  bl(m2,1)=ya(k)-yb(k);
end

leta=inv(LA'*LA)*LA'*bl;
txit(1,:)=(ra-leta(2)).*sin(ta-leta(1));
txit(2,:)=(ra-leta(2)).*cos(ta-leta(1));
tdse1=(txit(1,:)-xi0(1,:))*(txit(1,:)-xi0(1,:))';
tdse2=(txit(2,:)-xi0(2,:))*(txit(2,:)-xi0(2,:))';
tdse=sqrt(tdse1+tdse2)/K;
ldse(tn)=ldse(tn)+tdse/TM;

% ********** the exact maximum likelihood algorithm ******************
TA1=zeros(2,2);
TA2=zeros(2,2);
ta1=zeros(2,1);
ta2=zeros(2,1);
```

```
% evaluation of the system error eta
for k=1:K
 alpha=xi(1,k);
 beta=xi(2,k);
 rda=tda(k);
 rdb=tdb(k);

 A1=[beta,alpha/rda;-alpha,beta/rda];
 A2=[beta-v,(alpha-u)/rdb;-(alpha-u),(beta-v)/rdb];
 TA1=TA1+A1'*A1;
 ta1=ta1+A1'*(XA(:,k)-xi(:,k));
 TA2=TA2+A2'*A2;
 ta2=ta2+A2'*(XB(:,k)-xi(:,k));
end

eta=[inv(TA1)*ta1;inv(TA2)*ta2];

% estimate the true target position xi, Gr is the gradient
Dm=zeros(4,1);
Gr=zeros(2,1);
Ja=zeros(2,4);
Hes=zeros(2,2);

for k=1:K
 alpha=xi(1,k);
 beta=xi(2,k);
 rda=tda(k);
 rdb=tdb(k);

 Ja(1,1)=-1-eta(2)*beta^2/rda^3;
 Ja(1,2)=eta(1)+eta(2)*alpha*beta/rda^3;
 Ja(1,3)=-1-eta(4)*(beta-v)^2/rdb^3;
 Ja(1,4)=eta(3)+eta(4)*(alpha-u)*(beta-v)/rdb^3;
 Ja(2,1)=-eta(1)+eta(2)*alpha*beta/rda^3;
 Ja(2,2)=-1-eta(2)*alpha^2/rda^3;
 Ja(2,3)=-eta(3)+eta(4)*(alpha-u)*(beta-v)/rdb^3;
 Ja(2,4)=-1-eta(4)*(alpha-u)^2/rdb^3;

 Dm(1)=alpha+alpha*eta(2)/rda+beta*eta(1);
 Dm(2)=beta+beta*eta(2)/rda-alpha*eta(1);
 Dm(3)=alpha+(alpha-u)*eta(4)/rdb+(beta-v)*eta(3);
 Dm(4)=beta+(beta-v)*eta(4)/rdb-(alpha-u)*eta(3);

 Gr=2*Ja*(X(:,k)-Dm);
 Hes=2*Ja*Ja';
 Mdn=inv(Hes)*Gr;
```

```
hmdn=sqrt(Mdn'*Mdn);

while hmdn>epsilonx

 % determine the step in the gradient
 for mx=1:tumax
  tmpxi(:,mx)=xi(:,k)-mu(mx)*Mdn;
  alpha=tmpxi(1,mx);
  beta=tmpxi(2,mx);
  rda=sqrt(alpha^2+beta^2);
  rdb=sqrt((alpha-u)^2+(beta-v)^2);

  Dm(1)=alpha+alpha*eta(2)/rda+beta*eta(1);
  Dm(2)=beta+beta*eta(2)/rda-alpha*eta(1);
  Dm(3)=alpha+(alpha-u)*eta(4)/rdb+(beta-v)*eta(3);
  Dm(4)=beta+(beta-v)*eta(4)/rdb-(alpha-u)*eta(3);

  tu(mx)=(X(:,k)-Dm)'*(X(:,k)-Dm);
 end

 [tumx,tumd]=min(tu);
 xi(:,k)=tmpxi(:,tumd);
 alpha=xi(1,k);
 beta=xi(2,k);
 rda=sqrt(alpha^2+beta^2);
 rdb=sqrt((alpha-u)^2+(beta-v)^2);
 tda(k)=rda;
 tdb(k)=rdb;

 Ja(1,1)=-1-eta(2)*beta^2/rda^3;
 Ja(1,2)=eta(1)+eta(2)*alpha*beta/rda^3;
 Ja(1,3)=-1-eta(4)*(beta-v)^2/rdb^3;
 Ja(1,4)=eta(3)+eta(4)*(alpha-u)*(beta-v)/rdb^3;
 Ja(2,1)=-eta(1)+eta(2)*alpha*beta/rda^3;
 Ja(2,2)=-1-eta(2)*alpha^2/rda^3;
 Ja(2,3)=-eta(3)+eta(4)*(alpha-u)*(beta-v)/rdb^3;
 Ja(2,4)=-1-eta(4)*(alpha-u)^2/rdb^3;

 Dm(1)=alpha+alpha*eta(2)/rda+beta*eta(1);
 Dm(2)=beta+beta*eta(2)/rda-alpha*eta(1);
 Dm(3)=alpha+(alpha-u)*eta(4)/rdb+(beta-v)*eta(3);
 Dm(4)=beta+(beta-v)*eta(4)/rdb-(alpha-u)*eta(3);

 Gr=2*Ja*(X(:,k)-Dm);
 Hes=2*Ja*Ja';
 Mdn=inv(Hes)*Gr;
```

```
  hmdn=sqrt(Mdn'*Mdn);

 end

end

TA1=zeros(2,2);
TA2=zeros(2,2);
ta1=zeros(2,1);
ta2=zeros(2,1);

for k=1:K
 alpha=xi(1,k);
 beta=xi(2,k);
 rda=tda(k);
 rdb=tdb(k);
 A1=[beta,alpha/rda;-alpha,beta/rda];
 A2=[beta,(alpha-u)/rdb;-(alpha-u),(beta-v)/rdb];
 TA1=TA1+A1'*A1;
 ta1=ta1+A1'*(XA(:,k)-xi(:,k));
 TA2=TA2+A2'*A2;
 ta2=ta2+A2'*(XB(:,k)-xi(:,k));
end

teta=[inv(TA1)*ta1;inv(TA2)*ta2];
adis=sqrt((eta-teta)'*(eta-teta));
eta=teta;

emiter=1;

% begin the iteration process
while adis>epsilone

 emiter=emiter+1;

 for k=1:K
  alpha=xi(1,k);
  beta=xi(2,k);
  rda=tda(k);
  rdb=tdb(k);

  Ja(1,1)=-1-eta(2)*beta^2/rda^3;
  Ja(1,2)=eta(1)+eta(2)*alpha*beta/rda^3;
  Ja(1,3)=-1-eta(4)*(beta-v)^2/rdb^3;
  Ja(1,4)=eta(3)+eta(4)*(alpha-u)*(beta-v)/rdb^3;
  Ja(2,1)=-eta(1)+eta(2)*alpha*beta/rda^3;
```

```
Ja(2,2)=-1-eta(2)*alpha^2/rda^3;
Ja(2,3)=-eta(3)+eta(4)*(alpha-u)*(beta-v)/rdb^3;
Ja(2,4)=-1-eta(4)*(alpha-u)^2/rdb^3;

Dm(1)=alpha+alpha*eta(2)/rda+beta*eta(1);
Dm(2)=beta+beta*eta(2)/rda-alpha*eta(1);
Dm(3)=alpha+(alpha-u)*eta(4)/rdb+(beta-v)*eta(3);
Dm(4)=beta+(beta-v)*eta(4)/rdb-(alpha-u)*eta(3);

Gr=2*Ja*(X(:,k)-Dm);
Hes=2*Ja*Ja';
Mdn=inv(Hes)*Gr;
hmdn=sqrt(Mdn'*Mdn);

while hmdn>epsilonx
 % determine the step in the gradient
 for mx=1:tumax
  tmpxi(:,mx)=xi(:,k)-mu(mx)*Mdn;
  alpha=tmpxi(1,mx);
  beta=tmpxi(2,mx);
  rda=sqrt(alpha^2+beta^2);
  rdb=sqrt((alpha-u)^2+(beta-v)^2);

  Dm(1)=alpha+alpha*eta(2)/rda+beta*eta(1);
  Dm(2)=beta+beta*eta(2)/rda-alpha*eta(1);
  Dm(3)=alpha+(alpha-u)*eta(4)/rdb+(beta-v)*eta(3);
  Dm(4)=beta+(beta-v)*eta(4)/rdb-(alpha-u)*eta(3);

  tu(mx)=(X(:,k)-Dm)'*(X(:,k)-Dm);
 end

 [tumx,tumd]=min(tu);
 xi(:,k)=tmpxi(:,tumd);
 alpha=xi(1,k);
 beta=xi(2,k);
 rda=sqrt(alpha^2+beta^2);
 rdb=sqrt((alpha-u)^2+(beta-v)^2);
 tda(k)=rda;
 tdb(k)=rdb;

 Ja(1,1)=-1-eta(2)*beta^2/rda^3;
 Ja(1,2)=eta(1)+eta(2)*alpha*beta/rda^3;
 Ja(1,3)=-1-eta(4)*(beta-v)^2/rdb^3;
 Ja(1,4)=eta(3)+eta(4)*(alpha-u)*(beta-v)/rdb^3;
 Ja(2,1)=-eta(1)+eta(2)*alpha*beta/rda^3;
 Ja(2,2)=-1-eta(2)*alpha^2/rda^3;
```

```
  Ja(2,3)=-eta(3)+eta(4)*(alpha-u)*(beta-v)/rdb^3;
  Ja(2,4)=-1-eta(4)*(alpha-u)^2/rdb^3;

  Dm(1)=alpha+alpha*eta(2)/rda+beta*eta(1);
  Dm(2)=beta+beta*eta(2)/rda-alpha*eta(1);
  Dm(3)=alpha+(alpha-u)*eta(4)/rdb+(beta-v)*eta(3);
  Dm(4)=beta+(beta-v)*eta(4)/rdb-(alpha-u)*eta(3);

  Gr=2*Ja*(X(:,k)-Dm);
  Hes=2*Ja*Ja';
  Mdn=inv(Hes)*Gr;
  hmdn=sqrt(Mdn'*Mdn);
 end
end

TA1=zeros(2,2);
TA2=zeros(2,2);
ta1=zeros(2,1);
ta2=zeros(2,1);

for k=1:K
 alpha=xi(1,k);
 beta=xi(2,k);
 rda=tda(k);
 rdb=tdb(k);

 A1=[beta,alpha/rda;-alpha,beta/rda];
 A2=[beta,(alpha-u)/rdb;-(alpha-u),(beta-v)/rdb];

 TA1=TA1+A1'*A1;
 ta1=ta1+A1'*(XA(:,k)-xi(:,k));
 TA2=TA2+A2'*A2;
 ta2=ta2+A2'*(XB(:,k)-xi(:,k));

end

teta=[inv(TA1)*ta1;inv(TA2)*ta2];
adis=sqrt((eta-teta)'*(eta-teta));
eta=teta;
end

eiter(tn)=eiter(tn)+emiter/TM;

lmeta(:,tn)=lmeta(:,tn)+(leta-eta0).^2/TM;
emeta(:,tn)=emeta(:,tn)+(eta-eta0).^2/TM;
```

```
    laeta(:,tn)=laeta(:,tn)+leta/TM;
    eaeta(:,tn)=eaeta(:,tn)+eta/TM;

    tdse1=(xi(1,:)-xi0(1,:))*(xi(1,:)-xi0(1,:))';
    tdse2=(xi(2,:)-xi0(2,:))*(xi(2,:)-xi0(2,:))';
    tdse=sqrt(tdse1+tdse2)/K;
    edse(tn)=edse(tn)+tdse/TM;

  end

  save rgsam1.mat;

end
```

drgsnr1.m

```
% This file drgsnr1.m is is to compute the MSE and mean of the system biases
% via the changes of SNR for 1) the exact maximum likelihood registration
% algorithm 2) the least squars algorithm

%                    McMaster University 1995

clear;
global u v geta globalx;

% K is the number of targets used in the registration algorithm
K=20;

% dta, dra, dtb,drb denotes the system bias of site A and B
dta=0.5*pi/180;
dtb=0.5*pi/180;
dra=1;
drb=1;
eta0=[dta,dra,dtb,drb]';

% the unit of range measurement is nautical miles
u=300;
v=0;

% obtain the true azimuth and range information with respect to A and B
% kr is the slope, st is the step size, mgv is the function for pattern
% "3" and "4", pd denotes the pattern number
mta=zeros(1,K);
mtb=zeros(1,K);
mra=zeros(1,K);
mrb=zeros(1,K);

st=4;
mgv=120;
kr=1;

tm=rgdata1(u,K,st);
%tm=rgdata2(u,kr,K,st);
%tm=rgdata3(u,K,st,mgv);
%tm=rgdata4(u,K,st,mgv);

mta=tm(1,:);
mra=tm(2,:);
mtb=tm(3,:);
mrb=tm(4,:);
```

```
% simulate the biased xy coordinate with respect to site A and B
txa=zeros(1,K);
tya=zeros(1,K);
txb=zeros(1,K);
tyb=zeros(1,K);

txa=(mra+dra).*sin(mta+dta);
tya=(mra+dra).*cos(mta+dta);
txb=(mrb+drb).*sin(mtb+dtb);
tyb=(mrb+drb).*cos(mtb+dtb);

snr0=0*dra;
snr1=2.0*dra;
dsnr=0.2*dra;

snr=snr1:-dsnr:snr0;
lsnr=length(snr);

cmeta=zeros(4,lsnr);
lmeta=zeros(4,lsnr);
emeta=zeros(4,lsnr);

dmeta=zeros(1,lsnr);
dleta=zeros(1,lsnr);

aeeta=zeros(4,lsnr);
aleta=zeros(4,lsnr);

eiter=zeros(1,lsnr);

thrd=1.0e-4;
epsilone=0.001;

%*************** compute the asymptotic covariance matrix ********
Rk=zeros(4,2);
tceta=zeros(4,4);
ceta=zeros(4,4);

for k=1:K

  alpha=mra(k)*sin(mta(k));
  beta=mra(k)*cos(mta(k));
  rda=sqrt(alpha^2+beta^2);
  rdb=sqrt((alpha-u)^2+(beta-v)^2);
```

```
Rk(1,1)=-1-eta0(2)*beta^2/rda^3;
Rk(2,1)=eta0(1)+eta0(2)*alpha*beta/rda^3;
Rk(3,1)=-1-eta0(4)*(beta-v)^2/rdb^3;
Rk(4,1)=eta0(3)+eta0(4)*(alpha-u)*(beta-v)/rdb^3;

Rk(1,2)=-eta0(1)+eta0(2)*alpha*beta/rda^3;
Rk(2,2)=-1-eta0(2)*alpha^2/rda^3;
Rk(3,2)=-eta0(3)+eta0(4)*(alpha-u)*(beta-v)/rdb^3;
Rk(4,2)=-1-eta0(4)*(alpha-u)^2/rdb^3;

Ppk=eye(4,4)-Rk*inv(Rk'*Rk)*Rk';
AA=zeros(4,4);
AA(1:2,1:2)=[beta,alpha/rda;-alpha,beta/rda];
AA(3:4,3:4)=[beta-v,(alpha-u)/rdb;-(alpha-u),(beta-v)/rdb];

tceta=tceta+AA'*Ppk*AA;
end

ceta=inv(tceta);

for tn=1:lsnr
 cmeta(:,tn)=snr(tn)*snr(tn)*diag(ceta);
end


% ********* obtain the initial estimate on xi and eta ***********
xi0(1,:)=mra.*sin(mta);
xi0(2,:)=mra.*cos(mta);

TM=100;

for tn=1:lsnr

 for tms=1:TM

  % generate the simulated target data for different SNR
  xa=txa+snr(tn)*randn(1,K);
  ya=tya+snr(tn)*randn(1,K);
  xb=txb+snr(tn)*randn(1,K)+u;
  yb=tyb+snr(tn)*randn(1,K)+v;

  X=[xa;ya;xb;yb];
  XA=[xa;ya];
  XB=[xb;yb];

  % obtain the noise azimuth and range measurement
```

```
for k=1:K
 ra(k)=sqrt(xa(k)*xa(k)+ya(k)*ya(k));
 rb(k)=sqrt((xb(k)-u)*(xb(k)-u)+(yb(k)-v)*(yb(k)-v));

 if abs(ya(k))<thrd
  ta(k)=pi/2;
  tb(k)=-pi/2;
 elseif ya(k)>=0
  ta(k)=atan(xa(k)/ya(k));
  tb(k)=atan((xb(k)-u)/yb(k));
 else
  ta(k)=-atan(ya(k)/xa(k))+pi/2;
  tb(k)=-atan(yb(k)/(xb(k)-u))-pi/2;
 end
end

% ***************** the least squares algorithm **********************
LA=zeros(2*K,4);
bl=zeros(2*K,1);

for k=1:K
 m1=2*(k-1)+1;
 m2=2*k;
 LA(m1,:)=[ra(k)*cos(ta(k)),sin(ta(k)),-rb(k)*cos(tb(k)),-sin(tb(k))];
 LA(m2,:)=[-ra(k)*sin(ta(k)),cos(ta(k)),rb(k)*sin(tb(k)),-cos(tb(k))];
 bl(m1,1)=xa(k)-xb(k);
 bl(m2,1)=ya(k)-yb(k);
end

leta=inv(LA'*LA)*LA'*bl;

% ********** the exact maximum likelihood algorithm ******************
eta=eta0;

%for k=1:K
%  tmpax=(ra(k)-eta(2))*sin(ta(k)-eta(1));
%  tmpay=(ra(k)-eta(2))*cos(ta(k)-eta(1));
%  tmpbx=(rb(k)-eta(4))*sin(tb(k)-eta(3))+u;
%  tmpby=(rb(k)-eta(4))*cos(tb(k)-eta(3))+v;
%  xi(1,k)=(tmpax+tmpbx)/2;
%  xi(2,k)=(tmpay+tmpby)/2;
%end

for k=1:K
 geta=eta;
 globalx=X(:,k);
```

```
  ta2=zeros(2,1);

  for k=1:K
    alpha=xi(1,k);
    beta=xi(2,k);
    rda=sqrt(alpha^2+beta^2);
    rdb=sqrt((alpha-u)^2+(beta-v)^2);

    A1=[beta,alpha/rda;-alpha,beta/rda];
    A2=[beta-v,(alpha-u)/rdb;-(alpha-u),(beta-v)/rdb];

    TA1=TA1+A1'*A1;
    ta1=ta1+A1'*(XA(:,k)-xi(:,k));
    TA2=TA2+A2'*A2;
    ta2=ta2+A2'*(XB(:,k)-xi(:,k));
  end

  teta=[inv(TA1)*ta1;inv(TA2)*ta2];
  adis=sqrt((eta-teta)'*(eta-teta));
  eta=teta;
 end

 lmeta(:,tn)=lmeta(:,tn)+(leta-eta0).^2/TM;
 emeta(:,tn)=emeta(:,tn)+(eta-eta0).^2/TM;

 dtx=(xi(1,:)-xi0(1,:))*(xi(1,:)-xi0(1,:))';
 dty=(xi(2,:)-xi0(2,:))*(xi(2,:)-xi0(2,:))';

 ltxa=(ra-leta(2)).*sin(ta-leta(1));
 ltya=(ra-leta(2)).*cos(ta-leta(1));
 ltxb=(rb-leta(4)).*sin(tb-leta(3))+u;
 ltyb=(rb-leta(4)).*cos(tb-leta(3))+v;

 dlxa=(ltxa-xi0(1,:))*(ltxa-xi0(1,:))';
 dlya=(ltya-xi0(2,:))*(ltya-xi0(2,:))';
 dlxb=(ltxb-xi0(1,:))*(ltxb-xi0(1,:))';
 dlyb=(ltyb-xi0(2,:))*(ltyb-xi0(2,:))';

 dleta(tn)=dleta(tn)+sqrt((dlxa+dlxb+dlya+dlyb)/(2*TM));
 dmeta(tn)=dmeta(tn)+sqrt((dtx+dty)/TM);

 aleta(:,tn)=aleta(:,tn)+leta/TM;
 aeeta(:,tn)=aeeta(:,tn)+eta/TM;
end
save drgsnr1.mat;
end
```

UNCLASSIFIED

SECURITY CLASSIFICATION OF FORM
(highest classification of Title, Abstract, Keywords)

DOCUMENT CONTROL DATA

(Security classification of title, body of abstract and indexing annotation must be entered when the overall document is classified)

1. ORIGINATOR (the name and address of the organization preparing the document. Organizations for whom the document was prepared, e.g. Establishment sponsoring a contractor's report, or tasking agency, are entered in section 8.)

Communications Research Laboratory
McMaster University
Hamilton, Ontario, L8S 4K1

2. SECURITY CLASSIFICATION (overall security classification of the document. including special warning terms if applicable)

UNCLASSIFIED

3. TITLE (the complete document title as indicated on the title page. Its classification should be indicated by the appropriate abbreviation (S,C,R or U) in parentheses after the title.)

Registration for Data Fusion by Exact Maximum Likelihood Method. (U)

4. AUTHORS (Last name, first name, middle initial)

Yifeng Zhou and P. Yip

5. DATE OF PUBLICATION (month and year of publication of document)

March 1996

6a. NO. OF PAGES (total containing information. Include Annexes, Appendices, etc.)

43

6b. NO. OF REFS (total cited in document)

15

7. DESCRIPTIVE NOTES (the category of the document, e.g. technical report, technical note or memorandum. If appropriate, enter the type of report, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered.)

Contract Final Report

8. SPONSORING ACTIVITY (the name of the department project office or laboratory sponsoring the research and development. Include the address.)

Defence Research Establishment Ottawa
3701, Carling Ave, Ottawa, Ontario, K1A 0K2

9a. PROJECT OR GRANT NO. (if appropriate, the applicable research and development project or grant number under which the document was written. Please specify whether project or grant)

5ab14

9b. CONTRACT NO. (if appropriate, the applicable number under which the document was written)

W7714-4-9827

10a. ORIGINATOR'S DOCUMENT NUMBER (the official document number by which the document is identified by the originating activity. This number must be unique to this document.)

CRL Report No. 323

10b. OTHER DOCUMENT NOS. (Any other numbers which may be assigned this document either by the originator or by the sponsor)

DRBO CONTRACTOR REPORT 96-745

11. DOCUMENT AVAILABILITY (any limitations on further dissemination of the document, other than those imposed by security classification)

(X) Unlimited distribution
( ) Distribution limited to defence departments and defence contractors; further distribution only as approved
( ) Distribution limited to defence departments and Canadian defence contractors; further distribution only as approved
( ) Distribution limited to government departments and agencies; further distribution only as approved
( ) Distribution limited to defence departments; further distribution only as approved
( ) Other (please specify):

12. DOCUMENT ANNOUNCEMENT (any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availabilty (11). However, where further distribution (beyond the audience specified in 11) is possible, a wider announcement audience may be selected.)

Unlimited announcement

13. ABSTRACT (a brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C), (R), or (U). It is not necessary to include here abstracts in both offical languages unless the text is bilingual).

The combination of data and information from multiple sources or sensors constitutes one of the most important problems in signal processing to-day. This process of data fusion has been defined as one of dealing with the association, correlation and combination of data. In this report, an exact maximum likelihood method is used to solve the problem of registration errors in data fusion. Registration is defined as the co-ordinate conversion of multiple source data. Error-free registration is a pre-requisite in the process of data fusion. In the exact maximum likelihood method (EML), a likelihood function is constructed based on the errors of co-ordinate transformation of the local sensor locations to a common system plane. Optimization is then carried out in a recursive two-step Gauss-Newton type procedure, which produces the correct system biases for proper registration. Simulation results show taht EML is more efficient when compared with conventional methods of registration.

14. KEYWORDS, DESCRIPTORS or IDENTIFIERS (technically meaningful terms or short phrases that characterize a document and could be helpful in cataloguing the document. They should be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location may also be included. If possible keywords should be selected from a published thesaurus. e.g. Thesaurus of Engineering and Scientific Terms (TEST) and that thesaurus-identified. If it is not possible to select indexing terms which are Unclassified, the classification of each should be indicated as with the title.)

Registration
Data Fusion
Maximum Likelihood
Real Time Quality Control (RTQC)
Least Squares
Bias Estimation
Gauss-Newton

#500711