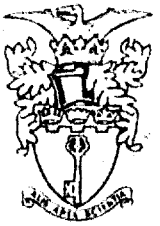
TR 88004

TR 88004



ROYAL AEROSPACE ESTABLISHMENT

Technical Report 88004

January 1988

MIXTURE REDUCTION ALGORITHMS FOR UNCERTAIN TRACKING

by

D. J. Salmond

Procurement Executive, Ministry of Defence
Farnborough, Hants

ROYAL AEROSPACE ESTABLISHMENT

Technical Report 88004

Received for printing 19 January 1988

MIXTURE REDUCTION ALGORITHMS FOR UNCERTAIN TRACKING

by

D. J. Salmond

SUMMARY

Bayesian solutions of tracking problems that involve measurement association uncertainty, give rise to Gaussian mixture distributions, which are composed of an ever increasing number of components. To implement such a tracking filter, the growth of components must be controlled by approximating the mixture distribution. A popular and economical scheme is the Probabilistic Data Association Filter (PDAF), which reduces the mixture to a single Gaussian component at each time step. However, this approximation may destroy valuable information, especially if several significant, well spaced components are present.

In this Report, two new algorithms for reducing Gaussian mixture distributions are presented. These techniques preserve the mean and covariance of the original mixture, and the final approximation is itself a Gaussian mixture. The reduction is achieved by successively merging components or groups of components. The two algorithms have been used to control the growth of components which occurs with the solution to the problem of tracking a single object, in the presence of uniformly distributed false measurements. Simulation results are presented which compare the performance of the resulting tracking filters and the PDAF.

This Report is an extended version of a joint paper with Prof D. Atherton and Prof J. Bather**, to be presented at the 1988 IFAC Symposium.*

Departmental Reference: Attack Weapons 28

* *University of Sussex, School of Engineering & Applied Sciences.*
** *University of Sussex, School of Mathematical & Physical Sciences.*

LIST OF CONTENTS

1 INTRODUCTION

A tracking filter is an algorithm for estimating the position (and possibly also the velocity or other factors) of an object from measurements of a sensor such as a radar. In general the sensor will produce measurements from the required object and also from random noise interference, clutter and other objects[11]. Usually it is not possible to distinguish with certainty between the useful measurements from the object and other unwanted measurements. In these circumstances the computational requirements of the full Bayesian solution to this problem rapidly increase as tracking proceeds. This Report is concerned with methods for containing the computational requirements within specified bounds, while minimizing the consequent performance penalty.

It is usual practice for the computational demands of the tracking filter to be controlled in two stages on every occasion that measurements are received from the sensor. The first of these is a coarse acceptance test which is applied before new measurements are processed, while the second stage is applied after processing. The acceptance test is effectively a tracking gate which rejects any measurements which are very unlikely to originate from the object of interest, and since it is applied before processing it is computationally inexpensive. This type of test is well known (see Refs 2, 9 and 11) and is widely applied to measurement association problems where ambiguities may exist. Therefore the acceptance test will not be considered further in the main text of this Report, although its application to the simulation example is described in Appendix D. After processing of the accepted measurements is complete, it may be necessary to approximate the solution to avoid an excessive computational load when subsequent measurements are incorporated. Unlike the acceptance test, this second stage of control may result in a significant modification of the complete solution. Hence careful consideration should be given to this approximation in order to minimize the effect on filter performance. The design of such reduction algorithms is the main subject of this Report.

In section 2 the Bayesian solution of the tracking problem is briefly discussed and a set of requirements for a reduction algorithm is formulated. Previous approaches to this problem are also discussed. The design of reduction algorithms is reported in sections 3 to 5, and in section 6 the performance of the algorithms is assessed by simulation for a particular tracking example. This assessment also compares the performance of the proposed algorithms with the popular Probabilistic Data Association Filter[3] (PDAF) approach.

2 MIXTURE DISTRIBUTIONS AND THE REQUIREMENTS OF A REDUCTION ALGORITHM

Bayes theory provides an ideal approach to the tracking problem. In this approach the probability density function (pdf) $p(\underline{x})$ of the state vector $\underline{x}$ at time t_k is constructed using all available information. Here $\underline{x}$ is the vector of parameters to be estimated. When further sensor measurements become available at time t_{k+1}, this information is used to update the pdf using Bayes theorem. In principle, an optimal estimate for any desired criterion, such as minimum mean square error, may be obtained from the pdf $p(\underline{x})$.

The solution of tracking problems involving measurement uncertainty leads to mixture distributions for the required state vector. A mixture distribution has a pdf of the form:

$$p(\underline{x}) = \sum_{i=1}^{N} \beta_i p_i(\underline{x})$$

where $p_i(\underline{x})$ is a component pdf and β_i is a probability associated with the ith component such that

$$\beta_i > 0 \quad \text{and} \quad \sum_{i=1}^{N} \beta_i = 1 \ .$$

For the tracking problem each component of the mixture corresponds to a possible track, and β_i is the probability that the assumed measurement history for track i is correct. If the equations of motion of the object to be tracked are linear with Gaussian disturbances, and measurements originating from the object are linearly related to the state vector but corrupted by Gaussian measurement noise, then each of the mixture components is a Gaussian distribution (see Refs 1, 2 and Appendix C, where this result is derived for a particular tracking problem):

$$p_i(\underline{x}) = \mathcal{N}(\underline{x} \ ; \ \underline{\mu}_i \ , \ P_i) \ ,$$

where $\underline{\mu}_i$ is the mean of the Gaussian distribution and
P_i is the covariance matrix.

In this case $p(\underline{x})$ is known as a Gaussian mixture (see Ref 6) and each component may be thought of as the output of a Kalman tracking filter.

While uncertainty persists, the number of components in the mixture for the full Bayesian solution will grow as tracking proceeds. The coarse acceptance test may be employed to cut down the number of feasible measurement histories to consider (and so the number of components) by rejecting very unlikely measurements. However if more than one measurement is passed by the acceptance test at each time step, the number of mixture components will still increase. Since every component must be propagated at each time step, to implement a tracking filter based on the Bayesian solution, it is essential to control the growth in the number of components. Here, it is considered that the control should be exercised by a reduction algorithm which fulfils the following requirements:

(i) The approximation should result in another Gaussian mixture. This is necessary to preserve the basic tracking filter algorithm which is a bank of Kalman filters.

(ii) The algorithm should allow the maximum number N_T of components after approximation to be chosen as desired.

(iii) Whenever possible, reduction should be achieved without modifying the 'structure' of the distribution beyond some acceptable limit. Conversely, to avoid retaining unnecessary components, reduction should continue until this limit is reached, so that the approximation may contain less than N_T components.

(iv) Intuitively the approximation should preserve the mean and covariance of the original mixture.

(v) The reduction algorithm should be computationally efficient, even when the original mixture consists of a large number of components (for example over 100), each with a different covariance matrix.

Of these requirements, number (iii) needs further comment. Ideally the reduction algorithm should attempt to maintain some level of filter performance within the limit of N_T components. A suitable performance measure would be the probability of losing track. Unfortunately the relationship between this performance measure and modification of the mixture distribution cannot be easily determined (but see section 6). However it is likely that the performance penalty will be related to the extent to which the approximation modifies the structure of the distribution. Hence requirement (iii) is written in terms of distribution structure (see below).

A number of techniques for controlling the growth of the mixture distribution have been reported. A popular and economical approach is the PDAF[3] in which the Gaussian mixture is approximated by a single Gaussian component at every time step. However if well spaced components are present, the approximation may destroy important structure in the distribution. Methods which allow for the retention of more than one component include the N-scan memory filter of Singer *et al*[1], and the direct approximation approaches of Alspach[4] and Lainiotis and Park[5]. None of these techniques ensures that the maximum number of components in the approximation is always within a specified limit, and Ref 1 does not use a direct measure of mixture structure. Ref 4 assures that all components have the same covariance and the method of Ref 5 would be very time consuming.

In the following sections two new mixture reduction algorithms which meet all of the above requirements are proposed. These algorithms operate by merging similar components together. In the first of these algorithms, the Joining Algorithm, a single pair of the 'most similar' components are merged at every iteration. In the second algorithm, the Clustering Algorithm, groups of similar components are merged at each iteration. The second method should be computationally more efficient, but with the former, the reduction process can be more finely regulated so that over-reduction is avoided. Both techniques are based on the same measure of mixture structure modification (requirement (iii)), which is derived from a decomposition of the mixture covariance matrix.

3 MIXTURE STRUCTURE: THE COVARIANCE MATRIX

The covariance matrix P of any mixture distribution with N components may be decomposed into two contributions, W and B (see Appendix A.1):

$$P = W + B$$

where $$W = \sum_{i=1}^{N} \beta_i P_i$$

$$B = \sum_{i=1}^{N} \beta_i \left(\underline{\mu}_i - \hat{\underline{x}}\right)\left(\underline{\mu}_i - \hat{\underline{x}}\right)^T$$

$$\hat{\underline{x}} \quad \sum_{i=1}^{N} \beta_i \underline{\mu}_i \quad \text{is the mean of the distribution, and}$$

$\underline{\mu}_i$, P_i and β_i are defined in section 2.

The matrix W may be interpreted as the contribution from the covariance 'within' each component of the mixture, while B may be interpreted as the between component contribution due to the separation between the mixture components. B and W are both symmetric matrices, W being positive definite and B being positive semi-definite.

Suppose that the mixture distribution is approximated by merging several components together. If $\mathbb{C}$ is the set of subscripts of components to be merged, then in order to preserve the mean and covariance of the mixture, the probability mass β', the mean $\underline{\mu}'$ and the covariance P' of the new component should be chosen (see Appendix A.2) as:

$$\beta' = \sum_{i\in\mathbb{C}} \beta_i , \qquad \underline{\mu}' = \frac{1}{\beta'} \sum_{i\in\mathbb{C}} \beta_i \underline{\mu}_i ,$$

$$P' = \frac{1}{\beta'} \sum_{i\in\mathbb{C}} \beta_i \left(P_i + \underline{\mu}_i \underline{\mu}_i^T \right) - \underline{\mu}' \underline{\mu}'^T \quad . \tag{1}$$

Although the overall covariance matrix P is unchanged, this merging of components results in a loss of between component covariance B which is balanced by an increase in W. More precisely, the difference $L = W' - W$ is a positive semidefinite matrix given by (see Appendix A.3):

$$L = \beta' P' - \sum_{i\in\mathbb{C}} \beta_i P_i = \sum_{i\in\mathbb{C}} \beta_i \left(\underline{\mu}_i - \underline{\mu}' \right) \left(\underline{\mu}_i - \underline{\mu}' \right)^T .$$

This shift of covarianve from B to W provides a useful measure of the change in the structure of a mixture distribution when components are combined. (A similar matrix decomposition has been used in Cluster Analysis, which is concerned with the grouping of data points into natural clusters - see Hand[6].)

4 THE JOINING ALGORITHM

Ideally the final partition of components into sets for merging should be such that the increase in some cost function is minimized. However, to reduce the mixture from N to M components, this could involve the evaluation of the criterion for every possible partition to identify the minimum. Such a procedure

for a number of different values of M would be far too time consuming and so a suboptimal approach has been adapted from the agglomerative methods of Cluster Analysis (see Hand[6]). In this approach, which we call the Joining Algorithm, a pair of components are merged at every iteration of the algorithm. The components for merging are chosen to minimize the increase in the chosen cost function at each stage. Clearly there is no guarantee that the final partition from such a procedure will achieve the smallest possible value of the cost function.

To implement the Joining Algorithm using a cost function based on an increase in the within component covariance, we require a suitable scalar measure. If components i and j are merged, the increase in W is given (see Appendix A.4) by:

$$L_{ij} = \frac{\beta_i \beta_j}{\beta_i + \beta_j} \left(\underline{\mu}_i - \underline{\mu}_j\right)\left(\underline{\mu}_i - \underline{\mu}_j\right)^T \quad .$$

One possible measure is the trace of L_{ij} which is the squared Euclidean distance between component means modified by the factor $\beta_i\beta_j/(\beta_i + \beta_j)$. However this has the disadvantage that it is dependent on the scaling of the elements of the state vector and so is problem dependent. This difficulty is avoided by using the Mahalanobis distance (see Ref 6) to give:

$$d_{ij}^2 = \frac{\beta_i \beta_j}{\beta_i + \beta_j} \left(\underline{\mu}_i - \underline{\mu}_j\right)^T P^{-1} \left(\underline{\mu}_i - \underline{\mu}_j\right) \quad , \qquad (2)$$

which is related to L_{ij} by (see Appendix A.5)

$$d_{ij}^2 = \mathrm{tr}\left(P^{-1} L_{ij}\right) \quad .$$

This measure is invariant under all non singular linear transformations of the state vector (see Appendix A.6). At each iteration of the Joining Algorithm, the two components which are closest in the sense of the distance measure, equation (2), are combined to form a new component defined by the relations, equation (1).

The minimum value of the distance measure at each iteration is an indicator of the change in distribution structure resulting from the merging of the two closest components. It is shown in Appendix B that this minimum distance increases monotonically as reduction proceeds and so each merging operation

increases this measure of structural modification. Thus if a threshold T defining the maximum acceptable modification to the distribution is specified, approximation should proceed until the minimum distance exceeds this threshold. In choosing a value for the threshold T, it is useful to note (see Appendix A.7) that the distance d_{ij}^2 is bounded:

$$d_{ij}^2 < \dim(\underline{x}) \quad .$$

Simulation studies indicate that a value of

$$T = 0.001 \dim(\underline{x})$$

retains sufficient components to give, on visual inspection, a good approximation to the mixture. At each iteration, the algorithm determines the number N_R of remaining components, excluding the set of smallest components with total probability mass (ie the sum of their β weights) less than B_T. If d_{ij}^2 exceeds T before N_R has been reduced below the specified maximum N_T, then approximation continues beyond the acceptable limit of modification. The purpose of B_T, which has been set to 0.01, is to avoid wasting effort on grouping insignificant components.

To implement the Joining Algorithm, a matrix (d_{ij}^2) containing the distance between every pair of components in the original mixture is evaluated using equation (2). Note the matrix (d_{ij}^2) is symmetric and $d_{ii} = 0$, so that only the upper triangular part of the matrix need be evaluated. At each iteration the smallest element of the matrix (for $i < j$) is found and the corresponding pair of components are merged using the formulae (1). Then row j and column j of the distance matrix are deleted, the new component is written into storage location i, and row i and column i of the matrix are revaluated using (2). Again all processing is confined to the upper triangle of the matrix. Note that since the merging of components preserves the covariance matrix P, only one matrix inversion suffices for all distance evaluations. Algorithm iterations continue until the stopping criteria are satisfied (see the flow diagram of the algorithm given in Fig 1). The storage requirement for the algorithm is $O(N^2)$, the number of distance evaluations is $O(N^2)$ and the number of comparisons is $O(N^3)$, where N is the number of components in the original mixture.

Figs 2 to 4 show an example of mixture reduction with the Joining Algorithm for a two dimensional distribution. Note that for $N_T = 10$ the approximation appears to be very good, although for $N_T = 4$ several important components have been combined.

5 THE CLUSTERING ALGORITHM

The second algorithm is based on the proposition that the mixture components with the largest β weightings carry the most important information. Thus starting with the largest component, this algorithm gathers in all surrounding components that are close to the principal component. Subsequently the largest component of the remainder is selected and the process is repeated until all the components have been clustered. This is called the Clustering Algorithm.

The distance measure chosen to represent the closeness of component i to the cluster centre is defined by

$$D_i^2 = \frac{\beta_i \beta_c}{\beta_i + \beta_c} \left(\underline{\mu}_i - \underline{\mu}_c\right)^T P_c^{-1} \left(\underline{\mu}_i - \underline{\mu}_c\right) ,$$

where β_c , $\underline{\mu}_c$ and P_c are the parameters of the principal component, and β_i and $\underline{\mu}_i$ are the probability mass and mean of the ith component. This is the same as the distance measure d_{ij}^2 of the previous section, except the distance is normalized to the covariance of the cluster centre rather than the complete mixture. Any component i for which $D_i^2 < T_1$ is selected as a cluster member. The threshold T_1 defines the acceptable modification to the distribution.

In chosing T_1 , it is helpful to first consider the measure $D_i'^2$ defined by:

$$D_i'^2 = \left(\underline{\mu}_i - \underline{\mu}_c\right)^T P_c^{-1} \left(\underline{\mu}_i - \underline{\mu}_c\right) .$$

If the criterion for clustering a component i were $D_i'^2 < T_1'$, then any component i whose mean were to fall within the hyperellipsoid defined by T_1' would be clustered. This hyperellipsoid is a contour of constant probability density of the principal component, and the proportion of probability mass enclosed is a measure of the selectivity of the clustering operation. If T_1' were chosen so that only a small proportion, say 1%, of the probability mass of the cluster centre were enclosed, then the structure of the distribution should be little altered by clustering. However $D_i'^2$ is independent of the probability mass (β_i) of the component, and intuitively, merging a large component would have a greater effect on the mixture than merging a small component. The modifying factor $\beta_i\beta_c/(\beta_i + \beta_c)$ biases this distance so that small components are more easily clustered while large components retain their individuality. It is suggested that the threshold for

$$D_i^2 = \frac{\beta_i \beta_c}{\beta_i + \beta_c} D_i'^2$$

should be chosen so that small components with β weights less than 0.05 are more readily clustered while components with β weights exceeding 0.05 are clustered less readily. Fig 5 shows that the contour

$$\frac{\beta_i \beta_c}{\beta_i + \beta_c} = 0.05$$

is close to the line $\beta_i = 0.05$ inside the region of interest, except when β_i is nearly equal to β_c. Thus it is suggested that to give a good mixture approximation, the threshold for D_i^2 should be set to

$$T_1 = 0.05 T_1' ,$$

where T_1' defines the hyperellipsoid containing only 1% of the probability mass. (T_1' can be found from tables of χ^2.)

Each cluster of components (some clusters may consist of a single component) is approximated by a single Gaussian defined by equation (1). Clustering proceeds until the probability mass of the unclustered components is less than B_T. As for the Joining Algorithm, the purpose of B_T, which is set to 0.01, is to avoid wasting effort on clustering insignificant components. If the number of clusters is less than or equal to N_T, the unclustered components are deleted and approximation is complete; otherwise further reduction is necessary. This is achieved by repeating the clustering procedure on the first approximation, but with the clustering threshold incremented by ΔT. This clustering operation is iterated until the necessary reduction has been effected. The choice of the increment ΔT is a compromise between the number of iterations required and the possibility of clustering more components than necessary. In this study, the value of ΔT is fixed:

$$\Delta T = 0.05 \Delta T'$$

where $T' + \Delta T'$ defines the hyperellipsoid which contains 6% of the probability mass of the principal component. (Simulation work has shown this to be a reasonable compromise.) However an override is provided which may increase the clustering threshold further to ensure that at least one component is clustered on each iteration. A flow diagram of the algorithm is given in Fig 6.

The computational cost of this algorithm depends on how many iterations are required to adequately reduce the mixture. It is most efficient when all approximation is accomplished within a single iteration, which required between N and NM distance evaluations and comparisons, and M matrix inversions to reduce the mixture from N to M components. In the worst case when only one component is clustered on each iteration, the number of distance evaluations and comparisons is $O(N^3)$ and the number of matrix inversions is $O(N^2)$. The algorithm requires minimal extra storage above that needed to hold the mixture components.

Figs 7 and 8 show the result of applying the Clustering Algorithm to the mixture shown in Fig 2. Note that for $N_T = 10$, the approximation is very similar to that produced by the Joining Algorithm (Fig 3). However for $N_T = 4$ there are clear differences between the approximations from the two algorithms (Figs 4 and 8).

6 COMPARISON OF FILTER PERFORMANCE AND THE EFFECT OF VARYING N_T

6.1 The tracking problem

The main object of this simulation is to compare the performance of tracking filters using the Joining Algorithm, the Clustering Algorithm and the PDAF approximation. The performance of these filters has been assessed for the problem of tracking an object moving in a plane, using measurements produced by a sensor (the same problem is considered in Ref 7). The simulation is in three parts:

(i) the generation of the object trajectory and the sensor measurements;

(ii) the implementation of the tracking filters;

(iii) the assessment of the filters' performance.

Object trajectories have been generated from an α-β model[8]. This model has been widely used in tracking problems as it is simple, while providing an adequate trajectory representation for many practical cases. The trajectory described by the model is a variation about a constant velocity course, whose magnitude and direction are defined by initial conditions. The deviation from this mean course is controlled by the variance q of the model driving noise. The α-β model is defined by the following equation:

$$\underline{x}_{k+1} = \begin{pmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{pmatrix} \underline{x}_k + \begin{pmatrix} \Delta t^2/2 & 0 \\ \Delta t & 0 \\ 0 & \Delta t^2/2 \\ 0 & \Delta t \end{pmatrix} \underline{w}_k \qquad (3)$$

where the state vector $\underline{x}_k$ represents the position and velocity of the object at time $k\Delta t$:

$$\underline{x}_k = \left(x , \dot{x} , y , \dot{y}\right)_k^T ,$$

Δt is the time step between measurements, and

w_k is a 2×1 vector from a Gaussian random sequence with zero mean and constant covariance

$$Q = \begin{pmatrix} q & 0 \\ 0 & q \end{pmatrix} .$$

Thus, to generate a trajectory $\{\underline{x}_k\}$, Gaussian random numbers of variance q were fed through the recurrence relation (3), starting from some initial condition $\underline{x}_0$.

At each time step a set of Cartesian position measurements have been generated to simulate sensor measurements. It is assumed that the probability P_D of detecting the object is unity, so exactly one measurement in each set originates from the object. This is called the true measurement and it is a Gaussian perturbation about the position of the object. It is generated from the state vector using the equation

$$\underline{z}_k = \begin{pmatrix} x \\ y \end{pmatrix}_k + \underline{v}_k \qquad (4)$$

where $\underline{v}_k$ is a 2×1 vector of Gaussian measurement noise with zero mean and constant covariance

$$R = \begin{pmatrix} r & 0 \\ 0 & r \end{pmatrix} .$$

The other measurements are independent of the object and are called false measurements. These are uniformly distributed over the sensor surveillance region, with density ρ per unit area. At each time step, the surveillance region of the sensor is arranged to be sufficiently extensive to include the object position and the acceptance regions of the filters, while track is maintained. False measurements were simulated by generating $A_k\rho$ pairs of uniformly distributed random numbers with appropriate scaling; A_k being the area of the surveillance region.

At each time step, every simulated measurement is passed to the tracking filters which attempt to estimate the current value of the object's state vector. The following information is available to the filters:

(i) the value of the initial state vector $\underline{x}_0$;

(ii) the model of the object motion, equation (3);

(iii) the relationship between the state vector and the true measurement, equation (4);

(iv) the statistics of the false measurements, the true measurement noise and the model driving noise, including the values of ρ, r and q ;

(v) the detection probability P_D of the sensor.

The tracking filters do not know:

(a) the values of the state vector $\underline{x}_k$, or the noise vectors $\underline{v}_k$ and $\underline{w}_k$ at each time step ($k \neq 0$);

(b) the identity of the true measurement.

As indicated, three filters have been implemented: the Joining Algorithm filter (JAF), the Clustering Algorithm filter (CAF) and the PDAF. As already discussed, each of these filters is based on the Bayesian solution of the above problem (see Appendix C) and each uses the coarse acceptance test described in Appendix D. The only difference between the filters is the mixture reduction algorithm employed. However for the PDAF, which approximates the mixture by a single Gaussian component, a full propagation of all components is unnecessary and a very efficient filter algorithm may be used[3].

The performance of the filters was assessed by measuring how long each of the three filters was able to maintain track on the object, *ie* the track lifetime. Each filter was allowed to continue tracking the object until track was lost. A track was deemed to be lost if either of the following criteria were satisfied:

(a) The true measurement is rejected by the acceptance test for five consecutive time steps.

(b) $|\hat{x}_k - x_k| > 10\,\sigma_{xk}$ or $|\hat{y}_k - y_k| > 10\,\sigma_{yk}$ for five consecutive time steps,

where $\hat{x}_k$, $\hat{y}_k$ is the filter estimate (the mean of the posterior distribution) of the object position at time step k ,

x_k , y_k is the actual object position at time step k , and

σ_{xk} and σ_{yk} are the standard deviations of the position estimates of the equivalent Kalman filter (*ie* the optimal filter for the same problem but with $\rho = 0$).

6.2 Choice of problem parameters

To analyse this tracking problem it is convenient to normalize the variables, so that the unit of time is Δt and the unit of distance is $\sqrt{r}$. Then the non-dimensional form of the state vector is

$$\underline{x}'_k = \left(\frac{x}{\sqrt{r}} , \frac{\dot{x}\Delta t}{\sqrt{r}} , \frac{y}{\sqrt{r}} , \frac{\dot{y}\Delta t}{\sqrt{r}} \right)^T_k .$$

If the target model and measurement equations are written in this form, it can be shown that the statistics of the problem are completely described by three non-dimensional parameters.

(i) $\frac{q\Delta t^4}{r}$, the ratio which determines the values of the filter gains for the standard α-β filter[10], *ie* in the absence of false measurements. As this parameter increases, the α-β filter becomes more responsive to position measurements.

(ii) ρr , the expected number of false measurements falling within a square whose side is one standard deviation of the measurement error.

(iii) Γ_D , the detection probability, assumed to be unity.

Since the initial state vector is assumed to be known perfectly, the filter performance in normalized co-ordinates should only depend on these three parameters. (This is because the problem may be written as the estimation of the deviation about the nominal constant velocity course defined by the initial state vector.)

All simulation results reported here are for a single point in this parameter space:

$$\frac{q\Delta t^4}{r} = 1$$

$$\rho r = 0.012$$

$$P_D = 1 \quad .$$

These values have been chosen to illustrate the possible improvement in tracking performance of the new reduction algorithms over the PDAF. However, it is believed that the region of the parameter space where there is a significant improvement is extensive, and a full investigation of filter performance over the space will be reported separately. Another factor in the choice of the above parameter values was a requirement for modest track lifetimes, to avoid excessive computation costs.

One hundred object trajectories with associated measurements were generated so that the mean track lifetime and the distribution of lifetimes could be estimated. The initial object position was taken as the origin and the initial speed was $10\sqrt{r}/\Delta t$. The initial heading of the object was chosen randomly for each trajectory. For the chosen problem parameters, the equivalent Kalman filter rapidly reaches steady state conditions, and the standard deviation of the position error on one of the co-ordinates approaches within 1% of its final steady state value after only four time steps. Also if the track of the object is estimated by assuming a constant velocity course and extrapolating from the initial perfect data (*ie* ignoring all measurements), the number k of time steps for the standard deviation of the error on one of the co-ordinates, say x , to exceed $10\ \sigma_{xk}$ (see track loss criterion (b)) is 7. These two figures provide a useful timescale when considering the results of the simulation.

6.3 Results

6.3.1 Average number of time steps to track loss

Fig 9 shows the average number N_{AVE} of time steps until track loss as a function of N_T , for filters using the Clustering Algorithm and the Joining Algorithm with thresholds set to the values given in sections 4 and 5. $N_T = 1$ corresponds to the special case of the PDAF, and clearly the filters which retain more than one mixture component perform better than the PDAF. The Joining Algorithm filter gives slightly larger values of N_{AVE} than the Clustering Algorithm, possibly due to the settings of T and T_1 .

Also shown in Fig 9 is the filter performance for the JAF with T = 0, *ie* with the acceptable modification check switched off. Note that the original setting of T for the JAF does not significantly degrade the filter's performance, and that the performance for all three cases shown in Fig 9 is similar. For $N_T < 10$, N_{AVE} rises approximately linearly with N_T, while for $N_T > 10$, N_{AVE} is nearly constant. For the JAF with T = 0 and N_T very large, the mixture is not subject to approximation, and so this constant level is the optimal value of N_{AVE}.

Fig 10 shows the average number of mixture components before and after reduction for the three cases of Fig 9. Comparing Fig 10a&b with Fig 10c, the effect of the acceptable modification check, defined by T_1 or T, in regulating the number of components for the large values of N_T is obvious. For small values of N_T, the approximation for all three cases is principally controlled by N_T itself. For this example, T_1 and T become the main regulators of the approximation at about $N_T = 10$, so the acceptable modification check appears to select the minimum number of components for near optimal performance. Clearly this cannot be guaranteed for other tracking problems, but since the thresholds were not specially tuned for this simulation, the performance with other problems may not be far from optimal.

For an interpretation of these results, it is useful to view the generation of mixture components as the filter's way of keeping options open when the choice of the true measurement is uncertain. First consider the optimal case when the mixture is not approximated. In this case, track will still be lost (according to the criteria of section 6.1) when components corresponding to false measurements are given a high probability weighting (β) through a chance event. For instance, a manoeuvre by the object under track might coincide with the production of a false measurement on the original heading, causing the filter to give a high weighting to the false measurement. Several such occurrences could lead the mean position estimate away from the actual object position so satisfying the track loss criteria. (Note that the average track lifetime depends on the track loss criteria.) Clearly the probability of such occurrences is likely to increase with the 'difficulty' of the tracking problem, for example if the density ρ of false measurements were to be increased. Thus, in agreement with intuition, we expect the average track survival time N_{AVE} to decrease with increasing problem difficulty for given (sensible) track loss criteria.

Now consider the effect of the reduction algorithms. Even without approximation, at any time step, many components of the distribution are almost identical so that the complete mixture distribution appears to consist of only a limited number of clearly distinct, significant components. For example, the distribution shown in Fig 2 comprises 37 components, although many of these are almost identical. The reduction algorithms attempt to combine the most similar components and if this can be accomplished without merging the significant distinct components (see Figs 2 and 3) little degradation in tracking performance is to be expected. However if the number of components retained falls below some critical level, these key components will be merged and tracking performance will deteriorate progressively as the permitted number of components is reduced. In the current example $N_T = 10$ appears to be the critical level at which tracking performance begins to degrade.

6.3.2 Distribution of number of time steps to track loss

In the previous section, the average track lifetime was discussed. In this section we consider the distribution of track lifetimes about this mean. To illustrate the distribution and to compare the performance of the CAF and JAF for individual replications, the track maintainence times have been plotted in Figs 11 to 13 for $N_T = 2$, 4 and 30 respectively. In these diagrams each point corresponds to a single replication, and the X and Y co-ordinates of the point are the time steps at which the JAF and CAF (with original threshold settings) lost track respectively. So points falling on the X = Y line indicate that both filters lost track coincidently. For large values of N_T (*eg* $N_T = 30$, Fig 13), the performance of the two filters is remarkably similar for the majority of replications. The few replications biasing N_{AVE} in favour of the JAF are obvious. For small values of N_T (*eg* $N_T = 2$, Fig 11), the points are scattered further from X = Y, although N_{AVE} is almost identical for the two filters. These results bear out the observation that the mixture approximations produced by the two reduction algorithms are usually very similar for large N_T, while for small N_T there are often clear differences.

Figs 14 and 15 show histograms of the data points from Figs 11 and 13; *ie* for the track lifetimes for the JAF and CAF with $N_T = 2$ and $N_T = 30$. It can be seen that those track lifetimes exceeding 20 time steps can be well fitted by an exponential distribution of the form:

$$p(t) = \begin{cases} \frac{1}{\alpha} e^{-\left(\frac{t - t_{min}}{\alpha}\right)} & \text{for} \quad t > t_{min} \\ 0 & \text{otherwise} \end{cases},$$

where $(t_{min} + \alpha)$ is the average lifetime of tracks which survive for at least $t_{min} = 20$ time steps. This is confirmed by a χ^2 test, the exponential hypothesis is only once rejected at the 5% level of significance for any of the 24 sets of replications. This exponential distribution indicates that after 20 time steps, the probability of losing track is independent of track lifetime, *ie* after an initial transient the filters reach steady state conditions. The value $t_{min} = 20$ was chosen by examining the transient behaviour of the equivalent Kalman filter (see last paragraph of section 6.2) and by inspection of the simulation results. The distribution parameter α may be interpreted as the average number of time steps that a track will survive in steady state conditions. Estimates of α are shown in Fig 16. These values are slightly greater than T_{AVE}-20 , as tracks surviving for less than 20 time steps are excluded.

6.3.3 Computation time

Fig 17 shows the average cpu time T_{AVE} for the filters to perform a single time step. The time scale is normalized to the average cpu time for a single PDAF time step which, for the data simulated here, was 1.12 ms on a Cray 1S computer. The computational effort is divided between the propagation of mixture components or tracks (see Appendix C) and mixture reduction. For the two filters with the original threshold settings (Fig 17a&b), T_{AVE} falls rapidly to nearly constant values for $N_T > 10$. Also for low values of N_T most time is spent reducing the mixture, and as N_T increases more time is required for track propagation while the mixture reduction time decreases. This is explained by Fig 10: the initial high values of T_{AVE} are due to time spent reducing large mixtures which result from inadequate approximations at values of $N_T < 6$. Except for the case $N_T = 6$, the JAF was more time-consuming than the CAF, usually by about 50%, and as expected, the execution times for the filters were in all cases considerably greater than the PDAF. However for $N_T > 10$, the eight-fold increase in execution time for the CAF may well be an acceptable price for the performance improvement offered by this filter.

The time taken by the JAF with $T = 0$ is shown in Fig 17c. This clearly shows the value of the acceptable modification check in the reduction algorithms:

for the very small improvement for $N_T > 10$ over the filter with the original threshold settings, there is a large increase in processing time. This extra time is required for the propagation and reduction of the extra tracks generated when the full N_T components are retained for $N_T > 10$ (see Fig 10).

7 CONCLUSIONS

(1) Two new mixture reduction algorithms for uncertain tracking have been developed. These algorithms have been applied to the optimal filter for tracking an object in uniformly distributed false measurements to produce two practical tracking filters: the Joining Algorithm filter (JAF) and the Clustering Algorithm filter (CAF).

(2) For the chosen simulation example (an object moving according to an α-β model) these filters give a substantial performance improvement over the popular PDAF filter: average track survival time (from an initially perfect track) may be increased by a factor of 8.

(3) However the computation times for these more complex filters are also greater than the PDAF: a factor of 8 for the CAF and a factor of 13 for the JAF. Also computer memory requirements are increased, particularly for the JAF.

(4) The simulation indicates that the minimum computation time and near optimum performance are obtained when satisfactory mixture approximation (defined by algorithm thresholds) is achieved within the maximum number of components allowed. If the permitted number of mixture components is reduced below some critical level, tracking performance will deteriorate.

(5) Under these conditions, the track survival times for the two filters were identical on at least 85% of the replications. This suggests that filter performance is not highly sensitive to the method of mixture reduction, provided that the most important mixture components are retained.

(6) With continuing improvements in computing power, tracking filters which retain more than one mixture component, such as the JAF and CAF, are practical alternatives to the PDAF for problems involving measurement association ambiguity. Further work is necessary to assess the performance and computer requirements of such filters for a wider range of problems.

Appendix A

PROOF OF RESULTS OF SECTIONS 3 AND 4

A.1 Structure of mixture covariance

Consider any mixture distribution with pdf

$$p(\underline{x}) = \sum_{i=1}^{N} \beta_i p_i(\underline{x})$$

and let the mean of the ith component be $\underline{\mu}_i$ and the covariance of the ith component be P_i .

The mean of the mixture is defined by

$$\hat{\underline{x}} = \int \underline{x} p(\underline{x}) d\underline{x}$$

$$= \sum_{i=1}^{N} \beta_i \int \underline{x} p_i(\underline{x}) d\underline{x} = \sum_{i=1}^{N} \beta_i \underline{\mu}_i \ .$$

The covariance matrix of the mixture is defined by

$$P = \int (\underline{x} - \hat{\underline{x}})(\underline{x} - \hat{\underline{x}})^T p(\underline{x}) d\underline{x}$$

$$= \int \underline{x}\underline{x}^T p(\underline{x}) d\underline{x} - \hat{\underline{x}}\hat{\underline{x}}^T$$

$$= \sum_{i=1}^{N} \beta_i \int \underline{x}\underline{x}^T p_i(\underline{x}) d\underline{x} - \hat{\underline{x}}\hat{\underline{x}}^T \ .$$

But

$$P_i = \int \underline{x}\underline{x}^T p_i(\underline{x}) d\underline{x} - \underline{\mu}_i \underline{\mu}_i^T \ ,$$

so

$$P = \sum_{i=1}^{N} \beta_i \left(P_i + \underline{\mu}_i \underline{\mu}_i^T \right) + \hat{\underline{x}}\hat{\underline{x}}^T \qquad \text{(A-1)}$$

$$= W + B$$

where $W = \sum_{i=1}^{N} \beta_i P_i$, which depends on the spread of each individual component of the mixture,

and $B = \sum_{i=1}^{N} \beta_i \underline{\mu}_i \underline{\mu}_i^T - \hat{\underline{x}}\hat{\underline{x}}^T$

$$= \sum \beta_i \underline{\mu}_i \underline{\mu}_i^T - \hat{\underline{x}} \sum \beta_i \underline{\mu}_i^T - \sum \beta_i \underline{\mu}_i \hat{\underline{x}}^T + \hat{\underline{x}}\hat{\underline{x}}^T$$

$$= \sum_{i=1}^{N} \beta_i \left(\underline{\mu}_i - \hat{\underline{x}} \right)\left(\underline{\mu}_i - \hat{\underline{x}} \right)^T$$, which depends on the separation between components.

A.2 Merging components

Suppose the reduced mixture $p_A(\underline{x})$ is formed by merging several components of the original mixture $p(\underline{x})$, *ie*

$$p_A(\underline{x}) = \beta' p'(\underline{x}) + \sum_{i \notin \mathbb{C}} \beta_i p_i(\underline{x})$$

where $p'(\underline{x})$ is the new component formed by merging those components with subscrips from the set $\mathbb{C}$. To ensure that $p_A(\underline{x})$ is a proper pdf, the probability mass of the new component must be given by

$$\beta' = \sum_{i \in \mathbb{C}} \beta_i \ .$$

If the means of $p(\underline{x})$ and $p_A(\underline{x})$ are to be equal,

$$\sum_i \beta_i \underline{\mu}_i = \beta' \underline{\mu}' + \sum_{i \notin \mathbb{C}} \beta_i \underline{\mu}_i \quad .$$

Thus the mean of the new component is given by

$$\underline{\mu}' = \frac{1}{\beta'} \sum_{i \in \mathbb{C}} \beta_i \underline{\mu}_i \quad .$$

If the covariances of $p_A(\underline{x})$ and $p(\underline{x})$ are to be equal, from (A-1)

$$\sum_i \beta_i \left(P_i + \underline{\mu}_i \underline{\mu}_i^T \right) - \hat{\underline{x}} \hat{\underline{x}}^T = \beta' \left(P' + \underline{\mu}' \underline{\mu}'^T \right) + \sum_{i \notin \mathbb{C}} \beta_i \left(P_i + \underline{\mu}_i \underline{\mu}_i^T \right) - \hat{\underline{x}} \hat{\underline{x}}^T \quad .$$

Thus the covariance of the new component is given by

$$P' = \frac{1}{\beta'} \sum_{i \in \mathbb{C}} \beta_i \left(P_i + \underline{\mu}_i \underline{\mu}_i^T \right) - \underline{\mu}' \underline{\mu}'^T \quad .$$

A.3 Merging components result in a loss of between component covariance

Let W and W' be the within component covariance of $p(\underline{x})$ and $p_A(\underline{x})$ respectively, and let B and B' be the between component covariance of $p(\underline{x})$ and $p_A(\underline{x})$ respectively (see section A.1). Then since overall covariance P is preserved,

$$P = W + B = W' + B' \quad .$$

Define the matrix L as

$$L = B - B' = W' - W$$

from above.

From sections A.1 and A.2,

$$W' - W = \beta'P' - \sum_{i\in\mathbb{C}} \beta_i P_i$$

$$= \sum_{i\in\mathbb{C}} \beta_i\left(P_i + \underline{\mu}_i\underline{\mu}_i^T\right) - \beta'\underline{\mu}'\underline{\mu}'^T - \sum_{i\in\mathbb{C}} \beta_i P_i$$

$$= \sum_{i\in\mathbb{C}} \beta_i\underline{\mu}_i\underline{\mu}_i^T - \beta'\underline{\mu}'\underline{\mu}'^T \quad .$$

However

$$\beta'\underline{\mu}'\underline{\mu}'^T = \sum_{i\in\mathbb{C}} \beta_i\underline{\mu}_i\underline{\mu}'^T = \sum_{i\in\mathbb{C}} \beta_i\underline{\mu}'\underline{\mu}_i^T = \sum_{i\in\mathbb{C}} \beta_i\underline{\mu}'\underline{\mu}'^T \quad ,$$

therefore $$L = W' - W = \sum_{i\in\mathbb{C}} \beta_i\left\{\underline{\mu}_i\underline{\mu}_i^T - \underline{\mu}_i\underline{\mu}'^T - \underline{\mu}'\underline{\mu}_i^T + \underline{\mu}'\underline{\mu}'^T\right\}$$

$$= \sum_{i\in\mathbb{C}} \left(\underline{\mu}_i - \underline{\mu}'\right)\left(\underline{\mu}_i - \underline{\mu}'\right)^T \quad .$$

Thus L is a positive semidefinite matrix and in this sense the merging of components results in a loss of between component covariance.

A.4 The loss of between component covariance resulting from merging two components

Suppose that only two components, i and j, are merged. Then from (A-2), the probability mass of the new component is

$$\beta' = \beta_i + \beta_j$$

the mean of the new component is

$$\underline{\mu}' = \frac{\beta_i\underline{\mu}_i + \beta_j\underline{\mu}_j}{\beta_i + \beta_j}$$

and the covariance of the new component is

$$P' = \frac{1}{\beta'}\left(\beta_i P_i + \beta_j P_j\right) + \frac{1}{\beta'}\left\{\beta_i \underline{\mu}_i \underline{\mu}_i^T + \beta_j \underline{\mu}_i \underline{\mu}_j^T - \frac{1}{\beta'}\left(\beta_i \underline{\mu}_i + \beta_j \underline{\mu}_j\right)\left(\beta_i \underline{\mu}_i + \beta_j \underline{\mu}_j\right)^T\right\}$$

$$= \frac{1}{\beta'}\left(\beta_i P_i + \beta_j P_j\right) + \frac{\beta_i \beta_j}{\beta'^2}\left(\underline{\mu}_i - \underline{\mu}_j\right)\left(\underline{\mu}_i - \underline{\mu}_j\right)^T .$$

From (A-2), the loss of between component covariance resulting from joining i and j is

$$L_{ij} = \beta' P' - \left(\beta_i P_i + \beta_j P_j\right)$$

$$= \frac{\beta_i \beta_j}{\beta_i + \beta_j}\left(\underline{\mu}_i - \underline{\mu}_j\right)\left(\underline{\mu}_i - \underline{\mu}_j\right)^T .$$

A.5 The relationship between d_{ij} and L_{ij}

Consider

$$\mathrm{tr}\left(P^{-1} L_{ij}\right) = \frac{\beta_i \beta_j}{\beta_i + \beta_j}\,\mathrm{tr}\left[P^{-1}\left(\underline{\mu}_i - \underline{\mu}_j\right)\left(\underline{\mu}_i - \underline{\mu}_j\right)^T\right]$$

$$= \frac{\beta_i \beta_j}{\beta_i + \beta_j}\left(\underline{\mu}_i - \underline{\mu}_j\right)^T P^{-1}\left(\underline{\mu}_i - \underline{\mu}_j\right)$$

$$= d_{ij}^2 ,$$

from (2) of section 4.

A.6 d_{ij}^2 is invariant under non-singular linear transformations of $\underline{x}$

Consider the transformation

$$\underline{y} = A\underline{x} + \underline{b} , \qquad \text{(A-3)}$$

where the inverse of A exists. If

$$p(\underline{x}) = \sum_{i=1}^{N} \beta_i \mathcal{N}\left(\underline{x};\underline{\mu}_i,P_i\right)$$

then under the above transformation

$$p(\underline{y}) = \sum_{i=1}^{N} \beta_i \mathcal{N}\left(\underline{y};\underline{\xi}_i,Q_i\right)$$

where $\underline{\xi}_i = A\underline{\mu}_i + \underline{b}$

and $Q_i = AP_iA^T$.

The distance between components i and j of $p(\underline{y})$ is given by

$$d_{ij}^2 = \frac{\beta_i\beta_j}{\beta_i + \beta_j}\left(\underline{\xi}_i - \underline{\xi}_j\right)^T Q^{-1}\left(\underline{\xi}_i - \underline{\xi}_j\right) \qquad \text{(A-4)}$$

where Q is the covariance of the mixture $p(\underline{y})$. From the linearity of the expectation operator $Q = APA^T$. Also

$$\underline{\xi}_i - \underline{\xi}_j = A\left(\underline{\mu}_i - \underline{\mu}_j\right) ,$$

so on substituting into (A-4),

$$d_{ij}^2 = \frac{\beta_i\beta_j}{\beta_i + \beta_j}\left(\underline{\mu}_i - \underline{\mu}_j\right)^T A^T\left(APA^T\right)^{-1}A\left(\underline{\mu}_i - \underline{\mu}_j\right) .$$

Hence, since $A^T\left(APA^T\right)^{-1}A = P^{-1}$, the distance measure is invariant under the transformation (A-3).

A.7 The distance d_{ij} is bounded

From section A.3,

$$P = W + B = W + B' + B - B'$$

$$= (W + B') + L_{ij}$$

where P and W are positive definite matrices, and B' and L_{ij} are positive semidefinite matrices. Multiply through by P^{-1} to give,

$$I = P^{-1}P = P^{-1}(W + B') + P^{-1}L_{ij} \quad .$$

Taking the trace gives

$$n = tr\left[P^{-1}(W + B')\right] + tr\left(P^{-1}L_{ij}\right)$$

where n is the dimension of the state space.

Hence since P^{-1} and $(W + B')$ are both positive definite,

$$tr\left[P^{-1}(W + B')\right] > 0$$

and so

$$d_{ij}^2 < n \quad .$$

Appendix B

THE MINIMUM DISTANCE BETWEEN COMPONENTS INCREASES MONOTONICALLY AS REDUCTION BY THE JOINING ALGORITHM PROCEEDS

Suppose that at some stage during mixture reduction, the closest components have means $\underline{x}$ and $\underline{y}$ and weights β_x and β_y. The distance between these components is d_{min}, where

$$d^2_{min} = f(\beta_x, \beta_y)||\underline{x} - \underline{y}||^2$$

where $||\underline{x} - \underline{y}||^2 = (\underline{x} - \underline{y})^T P^{-1}(\underline{x} - \underline{y})$

and $f(\beta_x, \beta_y) = \beta_x \beta_y / (\beta_x + \beta_y)$.

As they are closest, these components are merged to produce a new component with mean

$$\underline{w} = \frac{\beta_x \underline{x} + \beta_y \underline{y}}{\beta_x + \beta_y}$$

and weight

$$\beta_w = \beta_x + \beta_y \ .$$

Now consider any other component with mean $\underline{z}$ and weight β_z. The distances d_{xz} and d_{yz} between this component and either of the two which have been merged must be greater than or equal to d_{min}, so

$$d^2_{min} \leq d^2_{xz} = f(\beta_x, \beta_z)||\underline{x} - \underline{z}||^2 \qquad \text{(B-1)}$$

and

$$d^2_{min} \leq d^2_{yz} = f(\beta_y, \beta_z)||\underline{y} - \underline{z}||^2 \ . \qquad \text{(B-2)}$$

To confirm that the minimum distance increases monotonically as reduction proceeds, we must prove that

$$d^2_{zw} \geq d^2_{min} \ .$$

Now

$$d_{zw}^2 = f(\beta_w,\beta_z)||\underline{z} - \underline{w}||^2$$

$$= f(\beta_w,\beta_z)\left|\left|\underline{z} - \frac{\beta_x\underline{x} + \beta_y\underline{y}}{\beta_w}\right|\right|^2$$

$$= f(\beta_w,\beta_z)\left|\left|(\underline{z} - \underline{y}) - \frac{\beta_x}{\beta_w}(\underline{x} - \underline{y})\right|\right|^2$$

$$= f(\beta_w,\beta_z)\left\{||\underline{z} - \underline{y}||^2 + \frac{\beta_x^2}{\beta_w^2}||\underline{x} - \underline{y}||^2 + \frac{\beta_x}{\beta_w}\left[||\underline{z} - \underline{x}||^2 - ||\underline{z} - \underline{y}||^2 - ||\underline{x} - \underline{y}||^2\right]\right\}$$

$$= \frac{f(\beta_w,\beta_z)}{\beta_w}\left\{\beta_y||\underline{z} - \underline{y}||^2 + \beta_x||\underline{z} - \underline{x}||^2 - \frac{\beta_x\beta_y}{\beta_w}||\underline{x} - \underline{y}||^2\right\}.$$

Since

$$\frac{f(\beta_w,\beta_z)}{\beta_w} = \frac{\beta_w\beta_z}{\beta_w(\beta_w + \beta_z)} = \frac{\beta_z}{\beta_w + \beta_z}$$

and using the definition of the distance measure,

$$d_{zw}^2 = \frac{1}{\beta_w + \beta_z}\left\{(\beta_y + \beta_z)d_{yz}^2 + (\beta_x + \beta_z)d_{xz}^2 - \beta_z d_{min}^2\right\}.$$

Hence from (B-1) and (B-2)

$$d_{zw}^2 \leq \frac{1}{\beta_w + \beta_z}\left\{\left(\beta_y + \beta_z\right)d_{min}^2 + \left(\beta_x + \beta_z\right)d_{min}^2 - \beta_z d_{min}^2\right\}$$

$$= \frac{1}{\beta_w + \beta_z}\left(\beta_y + \beta_z + \beta_x\right)d_{min}^2 = d_{min}^2 \quad .$$

This completes the proof.

Appendix C

BAYESIAN SOLUTION OF AN UNCERTAIN TRACKING PROBLEM

C.1 Introduction

This Appendix contains a formal statement of a tracking problem of which an example is given in section 6. This tracking problem, which is taken from Refs 1 and 3, illustrates many of the difficulties of uncertain tracking. The purpose of this Appendix is to show that the optimal solution of the tracking problem generates Gaussian mixture distributions and to specify the optimal tracking filter. The recurrence relations of the JAF and CAF (see section 6.1) are the same as the optimal filter, except that received measurements are subjected to a coarse acceptance test and the Gaussian mixture (C-21) is approximated at each time step.

The solution of the tracking problem is approached from a Bayesian point of view (see Refs 1 to 4). We consider the conditional pdf of the state vector of the object at time t_k, conditioned by all the information available up to that time. This conditional pdf is a complete solution of the tracking problem. In section C.3 it is shown that the conditional pdf is a Gaussian mixture. Assuming the prior pdf of the state at time step k is a Gaussian mixture and given the problem statement of section C.2, the posterior pdf, after updating with measurements received at this time step is shown to be another Gaussian mixture, with an increased number of components. This posterior pdf is projected forwards to show that the prior pdf at the following time step $k + 1$ is also a Gaussian mixture. Thus the solution is established by induction.

C.2 Problem formulations

It is assumed that the state vector $\underline{x}$ of the object of interest evolves according to a linear equation

$$\underline{x}_{k+1} = \Phi \underline{x}_k + \Gamma \underline{w}_k \qquad \text{(C-1)}$$

where $\underline{x}_k$ is the n-component state vector at time t_k,

Φ is the $n \times n$ state transition matrix,

Γ is an $n \times r$ matrix

and $\underline{w}_k$ is an r-component vector of system driving noise which has a Gaussian distribution with zero mean and covariance

$$E\left[\underline{w}_i \underline{w}_k^T\right] = Q\delta_{ik} \quad . \tag{C-2}$$

Here Q is a positive definite $r \times r$ matrix and δ_{ik} is the Kronecker delta. The state vector contains the object position, and usually the velocity and possibly other attributes of the object. Also it is assumed that at time t_1, the state vector $\underline{x}$, is known to have a Gaussian distribution with mean $\underline{\bar{x}}_1$ and covariance M_1.

At every time step k (*ie* at each scan), a number of measurements are received from the sensor. If Z_k denotes the set of m_k measurements received at time t_k, then

$$Z_k = \left\{ \underline{z}_{kj} : j = 1, \ldots, m_k \right\} \quad .$$

Each measurement $\underline{z}_{kj}$ is a u-component vector. It is assumed that the object is well inside the surveillance region of the sensor, but that the (known) probability P_D of detecting the object may be less than unity. It is also assumed that at most one of the measurements may originate from the object. If measurement $\underline{z}_{kj}$ does originate from the object then it is related to the state vector by the linear relationship

$$\underline{z}_{kj} = H\underline{x}_k + \underline{v}_k \quad , \tag{C-3}$$

where H is the $u \times n$ measurement matrix
and $\underline{v}_k$ is a u-component vector of measurement noise which has a Gaussian distribution with zero mean and covariance

$$E\left[\underline{v}_i \underline{v}_k^T\right] = R\delta_{ik} \quad . \tag{C-4}$$

Here R is a positive definite $u \times u$ matrix and δ_{ik} is the Kronecker delta. A measurement which originates from the object is said to be true, while all other measurements are false. A false measurement is assumed to be independent of the state vector, to have a uniform distribution over the surveillance region of the sensor and to be independent of all present and past measurements. False measurements are assumed to occur at an average density of ρ per unit area. Further it is assumed that before examining the values of the measurements in the set Z_k, there is no information on which, if any, of the measurements are associated with the object. Note that if the identity of the true measurement were known, the problem would reduce to that of the standard Kalman filter.

The tracking problem is to estimate the state vector $\underline{x}_k$ at each time step, based on the available information up to and including time t_k. It is assumed that Φ, Γ, Q, H, R, P_D and ρ are given, together with all the measurements.

C.3 The optimal solution

C.3.1 The prior distribution of the state vector at time t_k

The prior pdf of the state vector at time t_k is the pdf of $\underline{x}_k$ given all available information up to time t_k but excluding the set of measurements received at time t_k. This available prior information at time t_k is denoted $\mathscr{P}_k$, and this includes all measurements received at the previous time steps:

$$Z_1, Z_2, \ldots, Z_{k-1} .$$

Since any one or none of the measurements of Z_i could be true, there are exactly $m_i + 1$ exclusive hypotheses concerning the truth or falsehood of the members of Z_i. Thus the total number of possible hypotheses under $\mathscr{P}_k$ is

$$n_{k-1} = \prod_{i=1}^{k-1} (m_i + 1) . \qquad \text{(C-5)}$$

Therefore, given n_{k-1} possible hypotheses, the pdf of the state vector $\underline{x}_k$ may be written

$$p\left(\underline{x}_k \middle| \mathscr{P}_k\right) = \sum_{i=1}^{n_{k-1}} p\left(\underline{x}_k \middle| \mathscr{H}_{k-1\ i}, \mathscr{P}_k\right) \Pr\left\{\mathscr{H}_{k-1\ i} \middle| \mathscr{P}_k\right\} . \qquad \text{(C-6)}$$

Here $\mathscr{H}_{k-1\ i}$ denotes one of the possible hypotheses on the measurements available under $\mathscr{P}_k$, $p\left(\underline{x}_k \middle| \mathscr{H}_{k-1\ i}, \mathscr{P}_k\right)$ is the pdf of $\underline{x}_k$ assuming $\mathscr{H}_{k-1\ i}$ is correct and $\mathscr{P}_k$ is given, and $\Pr\left\{\mathscr{H}_{k-1\ i} \middle| \mathscr{P}_k\right\}$ is the probability that $\mathscr{H}_{k-1\ i}$ is correct given the information $\mathscr{P}_k$.

Now suppose that the conditional pdfs in the RHS of (C-6) are known to be Gaussian, *ie*

$$p\left(\underline{x}_k \middle| \mathscr{H}_{k-1\ i}, \mathscr{P}_k\right) = \mathcal{N}\left(\underline{x}_k ; \bar{\underline{x}}_{ki}, M_{ki}\right) . \qquad \text{(C-7)}$$

Also suppose that the probabilities of the hypotheses are known and are denoted

$$\Pr\left\{ \mathscr{H}_{k-1\ i} \middle| \mathscr{D}_k \right\} = \beta_{k-1\ i} \quad . \tag{C-8}$$

In this case (C-6) is a fully specified Gaussian mixture pdf. Note that the above suppositions are true for $k = 1$.

C.3.2 The posterior pdf of the state vector

The set Z_k of m_k measurements received at time t_k is to be used to update the prior pdf of $\underline{x}_k$ specified by (C-6) to (C-8). The resulting posterior pdf is denoted

$$p\left(\underline{x}_k \middle| Z_k, \mathscr{D}_k \right) \quad .$$

In the following working we shall omit $\mathscr{D}_k$ for ease of notation, although the dependency should be understood for all conditional probabilities and pdfs. Thus the posterior pdf of $\underline{x}_k$ will be written

$$p\left(\underline{x}_k \middle| Z_k \right) \quad .$$

After updating with the latest set of measurements, the total number of possible hypotheses is increased to

$$n_{k-1}\left(m_k + 1 \right) \quad .$$

This increase may be viewed as a branching process where each of the $\mathscr{H}_{k-1\ i}$ prior hypotheses of (C-6) may be seen as a potential track and each of these tracks then splits into a further $m_k + 1$ tracks resulting from the new set of measurements. Thus a posterior hypothesis including the latest set of measurements Z_k may be written as a joint hypothesis

$$\mathscr{H}'_{kij} = \left(\mathscr{H}_{k-1\ i}, \Psi_{kj} \right) \quad .$$

where Ψ_{kj} is independent of $\mathscr{H}_{k-1\ i}$ and indicates that the jth measurement of set Z_k is true (or that they are all false if $k = 0$). The complete set of posterior hypotheses is

$$\left\{ \mathscr{H}'_{kij} : i = 1, \ldots, n_{k-1} ; j = 0, \ldots, m_k \right\} \quad .$$

Hence the posterior of pdf of $\underline{x}_k$ may be written in the form

$$p\left(\underline{x}_k \middle| Z_k\right) = \sum_{i=1}^{n_{k-1}} \sum_{j=0}^{m_k} p\left(\underline{x}_k \middle| \mathscr{H}'_{kij}, Z_k\right) \Pr\left\{\mathscr{H}'_{kij} \middle| Z_k\right\} \quad . \tag{C-9}$$

First consider the posterior pdf of $\underline{x}_k$ conditioned by $\mathscr{H}'_{kij}$:

$$p\left(\underline{x}_k \middle| \mathscr{H}'_{kij}, Z_k\right)$$

is the probability density resulting from updating $p\left(\underline{x} \middle| \mathscr{H}_{k-1\ i}\right)$ on the assumption that the jth measurement from Z_k is true (for $j \neq 0$). In this case $\underline{z}_{kj}$ is the only useful measurement from Z_k and the other members of Z_k can be discarded since they contain no relevant information. A true measurement $\underline{z}_T$ has a Gaussian distribution:

$$\mathscr{N}\left(\underline{z}_T\ ;\ H\underline{x}_k,\ R\right)$$

and the prior density of $\underline{x}_k$ under $\mathscr{H}_{ki}$ is also Gaussian given by (C-7). Hence the required posterior density is also Gaussian and is given by the standard Kalman filter. So for $j \neq 0$,

$$\left.\begin{aligned} p\left(\underline{x}_k \middle| \mathscr{H}'_{kij}, Z_k\right) &= \mathscr{N}\left(\underline{x}_k\ ;\ \hat{\underline{x}}'_{kij},\ P'_{kij}\right) \\ \text{where}\quad \hat{\underline{x}}'_{kij} &= \bar{\underline{x}}_{ki} + K_{ki}\left(\underline{z}_{kj} - H\bar{\underline{x}}_{ki}\right), \\ K_{ki} &= P'_{kij}H^T R^{-1}\ , \\ P'_{kij} &= M_{ki} - M_{ki}H^T S_{ki}^{-1} H M_{ki} \\ \text{and}\quad S_{ki} &= H M_{ki} H^T + R\ . \end{aligned}\right\} \tag{C-10}$$

If $j = 0$, none of the members of Z_k are true and so the prior pdf is not modified:

$$\left.\begin{aligned} \hat{\underline{x}}'_{ki0} &= \bar{\underline{x}}_{ki} \\ \text{and} \\ P'_{ki0} &= M_{ki}\ . \end{aligned}\right\} \tag{C-11}$$

Now turning to the second term in the summation of equation (C-9), the posterior probability that $\mathscr{H}'_{kij}$ is correct may be evaluated using Bayes theorem:

$$\Pr\left\{\mathscr{H}'_{kij}\middle|Z_k\right\} = \frac{p\left(Z_k\middle|\mathscr{H}'_{kij}\right)\Pr\left\{\psi_{kj}\middle|\mathscr{H}'_{k-1\ i}\right\}\Pr\left\{\mathscr{H}_{k-1\ i}\right\}}{p\left(Z_k\right)} \quad . \quad \text{(C-12)}$$

The equation (C-12) indicates how the prior probability $\Pr\left\{\mathscr{H}_{k-1\ i}\right\}$ is modified by the observations at time t_k. The posterior probability can be found by evaluating the three factors in the numerator of the RHS of (C-12).

First consider $p\left(Z_k\middle|\mathscr{H}'_{kij}\right)$. This may be written

$$p\left(Z_k\middle|\mathscr{H}'_{kij}\right) = \int p\left(Z_k,\underline{x}_k\middle|\mathscr{H}'_{kij}\right)\underline{dx}_k = \int p\left(Z_k\middle|\underline{x}_k,\mathscr{H}'_{kij}\right)p\left(\underline{x}_k\middle|\mathscr{H}'_{kij}\right)\underline{dx}_k$$

..... (C-13)

Since the elements of Z_k are independent

$$p\left(Z_k\middle|\underline{x}_k,\mathscr{H}'_{kij}\right) = \prod_{\ell=1}^{m_k} p\left(\underline{z}_{k\ell}\middle|\underline{x}_k,\psi_{kj}\right) \quad .$$

A measurement $\underline{z}_{k\ell}$ is false under ψ_{kj} if $j \neq \ell$. False measurements are uniformly distributed over the surveillance region of the sensor, and so the pdf of a false measurement is V_k^{-1}, where V_k is the volume of the surveillance region. If $j = \ell$, the measurement $\underline{z}_{k\ell}$ is true and so is a sample from the Gaussian distribution defined by (C-3). The prior pdf of $\underline{x}_k$,

$$p\left(\underline{x}_k\middle|\mathscr{H}'_{kij}\right) = p\left(\underline{x}_k\middle|\mathscr{H}_{k-1\ i}\right)$$

which is the Gaussian pdf (C-7). Hence on substituting into (C-13) we obtain, for $j \neq 0$,

$$p\left(z_k \middle| \mathscr{H}'_{kij}\right) = V_k^{-m_k+1} \int \mathcal{N}\left(\underline{z}_{kj} ; H\underline{x}_k, R\right) \mathcal{N}\left(\underline{x}_k ; \bar{\underline{x}}_{ki}, M_{ki}\right) d\underline{x}_k$$

$$= V_k^{-m_k+1} \mathcal{N}\left(\underline{z}_{kj} ; H\bar{\underline{x}}_{ki}, S_{ki}\right) \qquad \text{(C-14)}$$

where S_{ki} is defined in the relations (C-10). Expression (C-14) is strictly correct only for a surveillance region of inifinite extent. However, the truncation effect is negligible provided that, for each component of $\underline{z}_{kj}$, the distance from $H\bar{\underline{x}}_{ki}$ to the boundary of the surveillance region is large compared with the standard deviation of that component. If $j = 0$ so all the measurements are false,

$$p\left(z_k \middle| \mathscr{H}'_{ki0}\right) = V_k^{-m_k} \quad . \qquad \text{(C-15)}$$

The second factor in the numerator of (C-12) is the prior probability of Ψ_{kj}:

$$\Pr\left\{\Psi_{kj} \middle| \mathscr{H}_{k-1\ i}\right\} = \Pr\left\{\Psi_{kj}\right\}$$

since the hypothesis on the current set of measurements is independent of hypotheses on measurements from previous time steps. The only prior information available is the probability P_D of detecting the target and the probability of the sensor receiving m false measurements. If false measurements are uniformly distributed over the measurement space with density ρ, then it can be shown that the probability of m false measurements falling within the surveillance region of the sensor is given by a Poisson distribution. If the volume of the surveillance region is V_k, the probability of receiving m false measurements is given by

$$g(m) = e^{-\rho V_k} \left(\rho V_k\right)^m / m! \quad . \qquad \text{(C-16)}$$

The hypothesis Ψ_{k0} corresponds to the event of failing to detect the target and receiving m_k false measurements. The prior probability of this occurrence is

$$\Pr\left\{\Psi_{k0}\right\} = \left(1 - P_D\right) g\left(m_k\right) \quad . \qquad \text{(C-17)}$$

Any of the hypotheses Ψ_{kj}, $j \neq 0$, could correspond to the situation of detecting the target and receiving $m_k - 1$ false measurements. *A priori*, each of these hypotheses is equally probable, and since there are m_k of them (for $j \neq 0$)

$$\Pr\left\{\Psi_{kj}\right\} = P_D g \left(m_k - 1\right)/m_k \quad . \tag{C-18}$$

The third factor in the numerator of (C-12) is given directly by (C-8):

$$\Pr\left\{\mathscr{H}_{k-1\ i}\right\} = \beta_{k-1\ i} \quad . \tag{C-19}$$

Substituting (C-14) to (C-19) into (C-12) we obtain

$$\Pr\left\{\mathscr{H}'_{kij} \middle| Z_k\right\} = \begin{cases} \dfrac{\beta_{k-1\ i}\, \mathcal{N}\left(\underline{z}_{kj}\ ;\ H\bar{\underline{x}}_{ki},\ S_{ki}\right)}{D} & \text{for } j \neq 0 \\ \dfrac{\beta_{k-1\ i}\left(1 - P_D\right)\rho}{P_D D} & \text{for } j = 0 \end{cases} \qquad \ldots\ldots \text{(C-20)}$$

where $$D = \frac{\left(1 - P_D\right)\rho}{P_D} + \sum_{r=1}^{n_{k-1}} \beta_{k-1\,r} \sum_{\ell=0}^{m_k} \mathcal{N}\left(\underline{z}_{k\ell}\ ;\ H\bar{\underline{x}}_{kr},\ S_{kr}\right)$$

is the normalizing denominator. This equation is of key importance because it defines the weightings of the mixture distribution (C-9). Note that if $P_D = 1$, as in the example of section 6, knowledge of the density ρ of false measurements does not contribute to the posterior pdf.

Thus the posterior pdf of $\underline{x}_k$ given by (C-9) is a fully specified Gaussian mixture. (C-9) can be rewritten as a single sum by defining

$$\mathscr{H}_{k\ell} = \mathscr{H}'_{kij}$$

$$\underline{\hat{x}}_{k\ell} = \underline{\hat{x}}'_{kij}$$

$$P_{k\ell} = P'_{kij}$$

and

$$\beta_{k\ell} = \Pr\left\{\mathscr{H}'_{kij} \middle| Z_k\right\}$$

where $\ell = (i-1)\left(m_k + 1\right) + j + 1$, for $i = 1, \ldots, n_{k-1}$ and $j = 0, \ldots, m_k$.

Thus

$$p\left(\underline{x}_k \middle| Z_k\right) = \sum_{\ell=1}^{n_k} p\left(\underline{x}_k \middle| \mathscr{H}_{k\ell}, Z_k\right) \Pr\left\{\mathscr{H}_{k\ell} \middle| Z_k\right\} \qquad \text{(C-21)}$$

where $n_k = n_{k-1}\left(m_k + 1\right) = \prod_{i=1}^{k}\left(m_i + 1\right)$,

$$p\left(\underline{x}_k \middle| \mathscr{H}_{k\ell}, Z_k\right) = \mathcal{N}\left(\underline{x}_k ; \hat{\underline{x}}_{k\ell}, P_{k\ell}\right)$$

and $\Pr\left\{\mathscr{H}_{k\ell} \middle| Z_k\right\} = \beta_{k\ell}$.

The Gaussian mixture (C-21) contains all the available information on the state vector $\underline{x}_k$ after taking account of the latest set of measurements Z_k. Thus in principle, the optimal estimate based on any desired criterion may be obtained. In particular the minimum mean square error estimate is the mean of the distribution:

$$\hat{\underline{x}}_k = \sum_{\ell=1}^{n_k} \beta_{k\ell} \hat{\underline{x}}_{k\ell} \ .$$

However a single value of $\underline{x}_k$ is a somewhat inadequate summary of a mixture distribution, especially if there are significant well spaced components.

C.3.3 The prior pdf of the state vector at time t_{k+1}

To establish, by induction, the general property that the prior pdf of $\underline{x}_k$ (equation (C-6)) is a fully specified Gaussian mixture, it is necessary to derive the pdf of $\underline{x}_{k+1}$ from the result (C-21). This pdf may be derived from $p\left(\underline{x}_k \middle| Z_k, \mathscr{D}_k\right)$ (note $\mathscr{D}_k$ is reinstated here) via the propagation equation (C-1). This information, together with Z_k and $\mathscr{D}_k$ is denoted

$\mathscr{P}_{k+1}$, which is all the prior information available at time t_{k+1}. The prior pdf of $\underline{x}_{k+1}$ may be written

$$p\left(\underline{x}_{k+1}\middle|\mathscr{P}_{k+1}\right) = \int p\left(\underline{x}_{k+1}\middle|\underline{x}_k\right) p\left(\underline{x}_k\middle|\mathscr{P}_{k+1}\right) d\underline{x}_k \quad . \tag{C-22}$$

$p\left(\underline{x}_{k+1}\middle|\underline{x}_k\right)$ is defined by the state propagation equations, and the second term

$$p\left(\underline{x}_k\middle|\mathscr{P}_{k+1}\right) = p\left(\underline{x}_k\middle|Z_k, \mathscr{P}_k\right)$$

since the extra information on state propagation from t_k to t_{k+1} does not contribute to the pdf of state at t_k. Substituting (C-21) into (C-22) and performing the integrations gives

$$p\left(\underline{x}_{k+1}\middle|\mathscr{P}_{k+1}\right) = \sum_{\ell=1}^{n_k} \Pr\left\{\mathscr{H}_{k\ell}\middle|\mathscr{P}_{k+1}\right\} p\left(\underline{x}_{k+1}\middle|\mathscr{H}_{k\ell}, \mathscr{P}_{k+1}\right) \tag{C-23}$$

where $\Pr\left\{\mathscr{H}_{k\ell}\middle|\mathscr{P}_{k+1}\right\} = \beta_{k\ell}$

and $p\left(\underline{x}_{k+1}\middle|\mathscr{H}_{k\ell}, \mathscr{P}_{k+1}\right) = \mathcal{N}\left(\underline{x}_{k+1}\,;\, \bar{\underline{x}}_{k+1\ \ell}, M_{k+1\ \ell}\right)$

with

$$\bar{\underline{x}}_{k+1\ \ell} = \Phi\hat{\underline{x}}_{k\ell}$$

and

$$M_{k+1\ \ell} = \Phi P_{k\ell}\Phi^T + \Gamma Q\Gamma^T \quad .$$

The pdf (C-23) is of the same form as (C-6): it is a fully specified Gaussian mixture. Hence the initial supposition of section C.3.1 is proved by induction.

C.4 Discussion

It has been shown that the posterior pdf of the state vector, just after incorporating the latest set of measurements, is a Gaussian mixture given by equation (C-21). This equation is a complete description of the filter's knowledge of the state vector. Each component of the mixture represents a potential track and is a Kalman filter estimate of the state vector based on a possible history of true and false measurements. At time t_k, the n_k components

represent all feasible track histories. The weighting $\beta_{k\ell}$ is the probability that track history ℓ is the correct one.

For most interesting cases, the number of components n_k becomes very large with increasing k (see (C-21)). Since every component must be propagated at each time step, implementation of the optimal solution is impractical, hence the need for the reduction algorithms which are the subject of this Report.

Appendix D

THE COARSE ACCEPTANCE CASE

A coarse acceptance test is applied to the sensor measurements to reject any hypothesis that appears to be very unlikely on the basis of prior information. This test is computationally inexpensive as the unlikely hypotheses are rejected before their corresponding posterior mixture components need be evaluated. Hopefully the effect of this acceptance test on the posterior distribution will be insignificant. The mixture reduction algorithm is applied after the posterior mixture distribution has been compiled.

Each component of the posterior pdf of the state vector is generated by updating a feasible track from the prior pdf with either a received measurement, or by prediction on the assumption that all received measurements are false (see Appendix C, section C.3.2). Consider the prior track, or component i of (C-6), that corresponds to hypothesis $\mathscr{H}_{k-1\ i}$. Under hypothesis $\mathscr{H}'_{kij}$ $(j \neq 0)$, measurement $\underline{z}_{kj}$ is true and is used to update prior component i. From (C-14), the prior pdf of $\underline{z}_{kj}$ under $\mathscr{H}'_{kij}$ $(j \neq 0)$ is given by

$$\mathcal{N}\left(\underline{z}_{kj} ; H\bar{\underline{x}}_{ki}, S_{ki}\right) .$$

From knowledge of this distribution, an acceptance or validation region in the measurement space may be defined, such that under hypothesis $\mathscr{H}_{k-1\ i}$, the probability of the true measurement falling outside the region is very small. (This type of acceptance test is commonly applied to measurement-track association problems where ambiguities may exist - see Refs 2, 9 and 11.) If the validation region is chosen so that the probability density of the true measurement at any point within the region exceeds that at all points outside the region, then the acceptance region is bounded by a hyperellipsoid. Thus a measurement $\underline{z}_{kj}$ is accepted for updating hypothesis $\mathscr{H}_{k-1\ i}$ if and only if

$$\left(\underline{z}_{kj} - H\bar{\underline{x}}_{ki}\right)^T S_{ki}^{-1}\left(\underline{z}_{kj} - H\bar{\underline{x}}_{ki}\right) < T_A \quad . \qquad \text{(D-1)}$$

Note that since the false measurements have a uniform distribution, this is equivalent to subjecting each measurement to a likelihood ratio test. For a true measurement $\underline{z}_{kj}$, under hypothesis $\mathscr{H}_{k-1\ i}$, the LHS of (D-1) is a sample from a χ^2 distribution with degrees-of-freedom equal to the dimension of $\underline{z}_{kj}$. Thus the value of T_A corresponding to a probability α of missing the true

measurement (if the object is detected and $\mathscr{H}_{k-1\ i}$ is correct) may be obtained from tables of χ^2. In the simulation of section 6, α is set to 0.001, which corresponds to $T_A = 13.82$ for two-dimensional measurement space. Note that a different acceptance region must be defined for each component of (C-6). To take account of the possibility of rejecting the true measurement, the detection probability P_D should be replaced by $P_D(1 - \alpha)$. Thus even if $P_D = 1$, a component is generated for the finite probability of missing the true measurement.

REFERENCES

No.	Author	Title, etc
1	R.A. Singer R.G. Sea K.B. Housewright	Derivation and evaluation of improved tracking filters for use in dense multi-target environments. IEEE Trans. on Information Theory, Vol IT-20, No.4, July 1974
2	Y. Bar-Shalom	Tracking methods in a multi-target environment. IEEE Trans. on Automatic Control, Vol AC-23, August 1978
3	Y Bar-Shalom E. Tse	Tracking in a cluttered environment with probabilistic data association. Automatica, Vol 11, September 1975
4	D.L. Alspach	A Gaussian sum approach to the multi-target identification tracking problem. Automatica, Vol 11, May 1975
5	D.G. Lainiotis S.K. Park	On joint detection, estimation and system identification: discrete data case. International Journal of Control, Vol 17, No.3 (1973)
6	D.J. Hand	Discrimination and classification. John Wiley (1981)
7	Y. Bar-Shalom K. Birmiwal	Consistency and robustness of PDAF for target tracking in cluttered environments. Automatica, Vol 19, No.4, July 1983
8	D.J. Salmond	The characteristics of the second-order target model. RAE Technical Report 85071 (1985)
9	A. Farina S. Pardini	Survey of radar data-processing techniques in air-traffic-control and surveillance systems. IEEE Proc., Vol 127, Part F, No.3, June 1980
10	D.J. Salmond	The Kalman filter, the α-β filter and smoothing filters. RAE Technical Memorandum AW 48 (1981)
11	S.S. Blackman	Multiple-target tracking with radar applications. Artech House (1986)

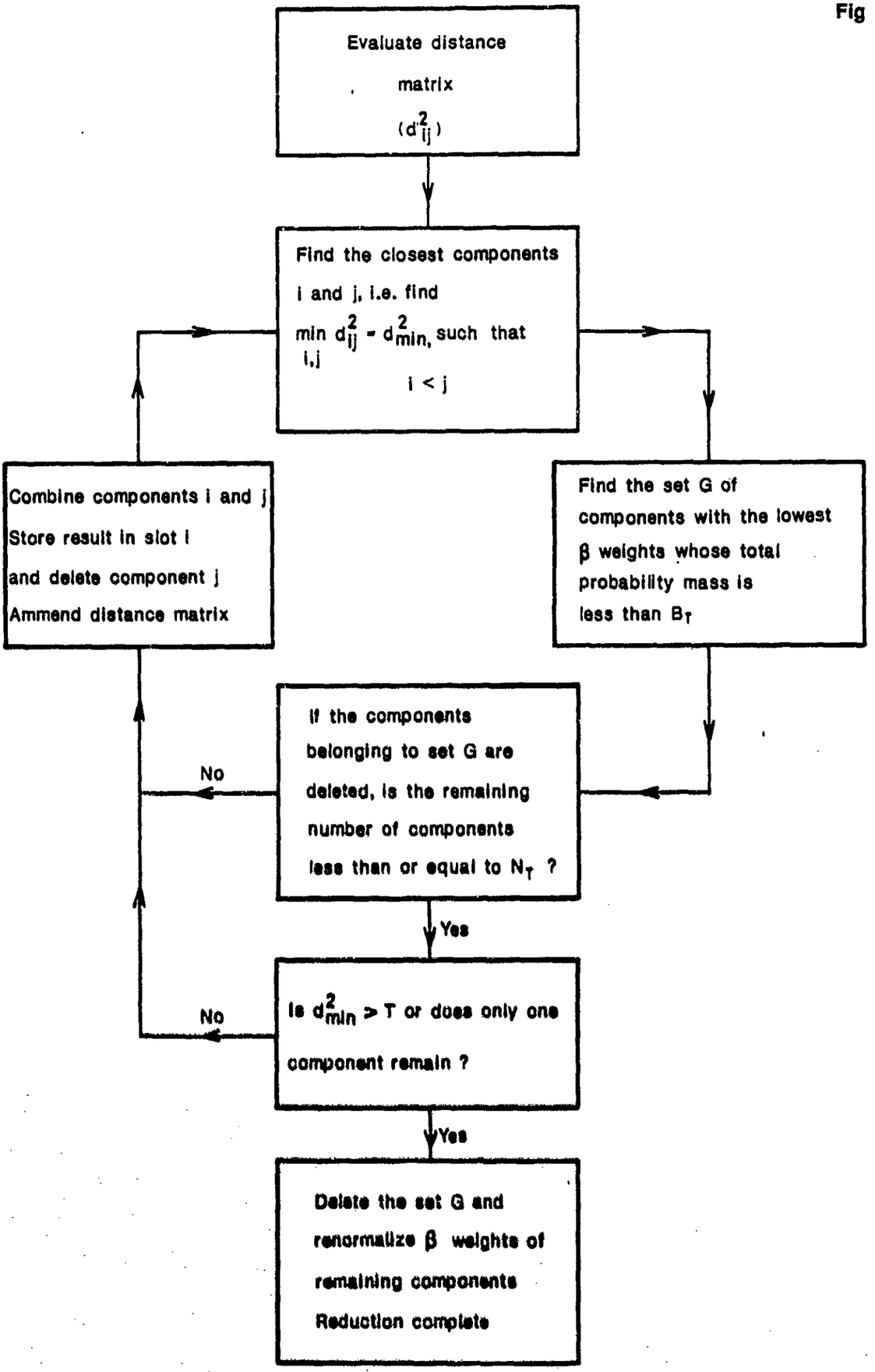


Fig 1 Flow diagram of the Joining Algorithm

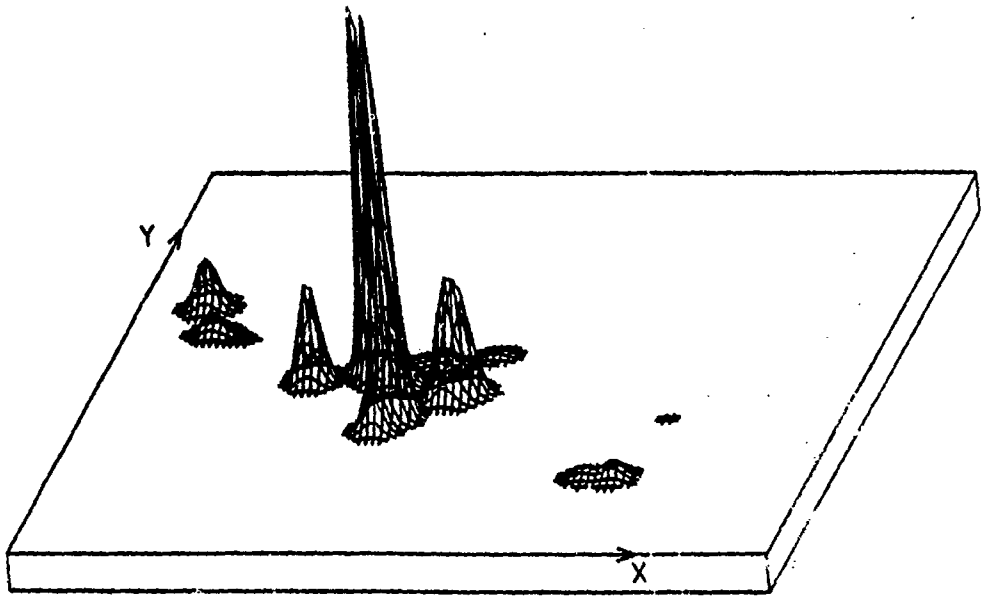


Fig 2 Mixture pdf before approximation (37 components)

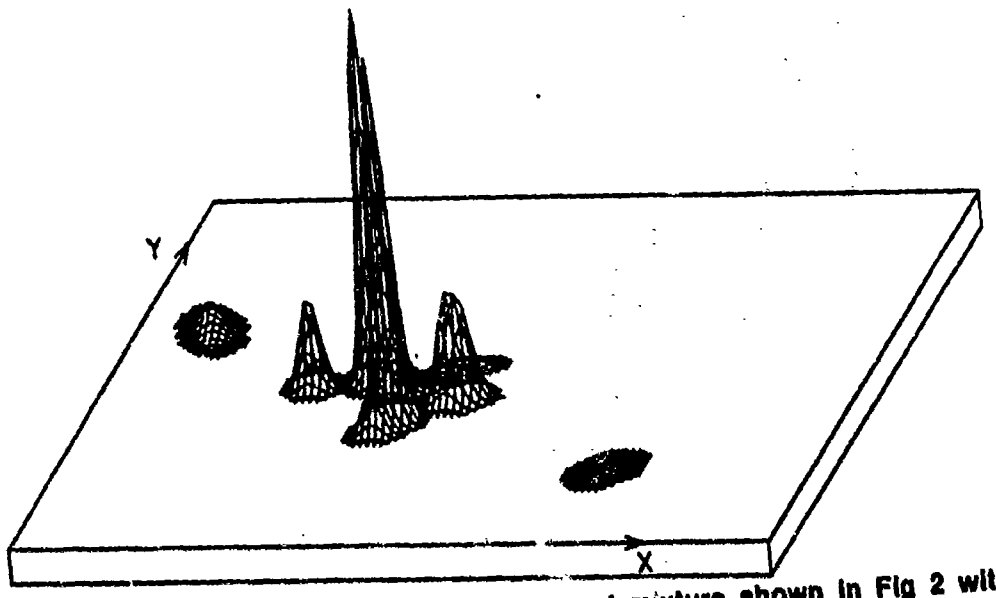


Fig 3 Joining Algorithm approximation of mixture shown in Fig 2 with N_T = 10 (9 components)

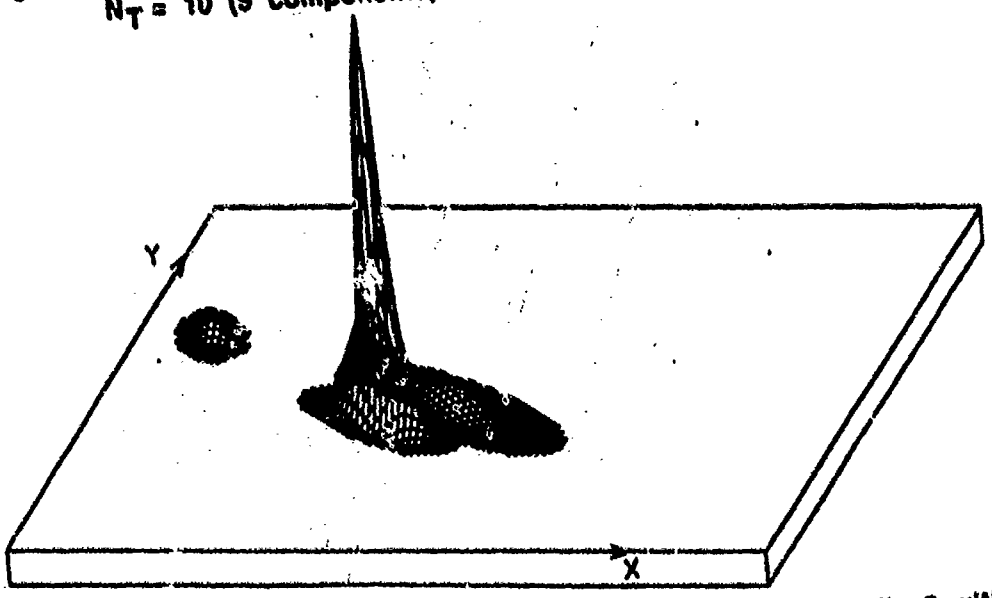


Fig 4 Joining Algorithm approximation of mixture shown in Fig 2 with N_T = 4 (4 components)

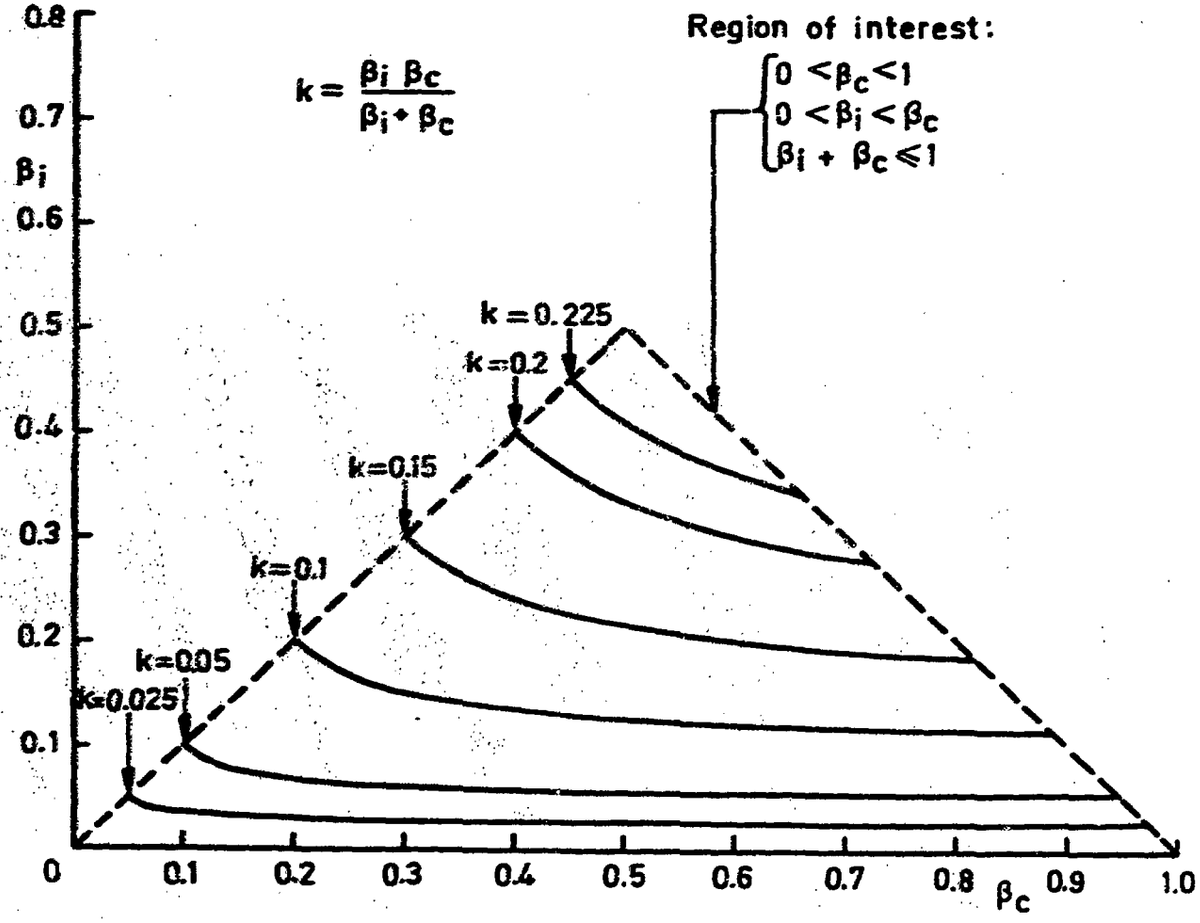


Fig 5 Contours of the modifying factor $\frac{\beta_i \beta_c}{\beta_i + \beta_c}$

Fig 6

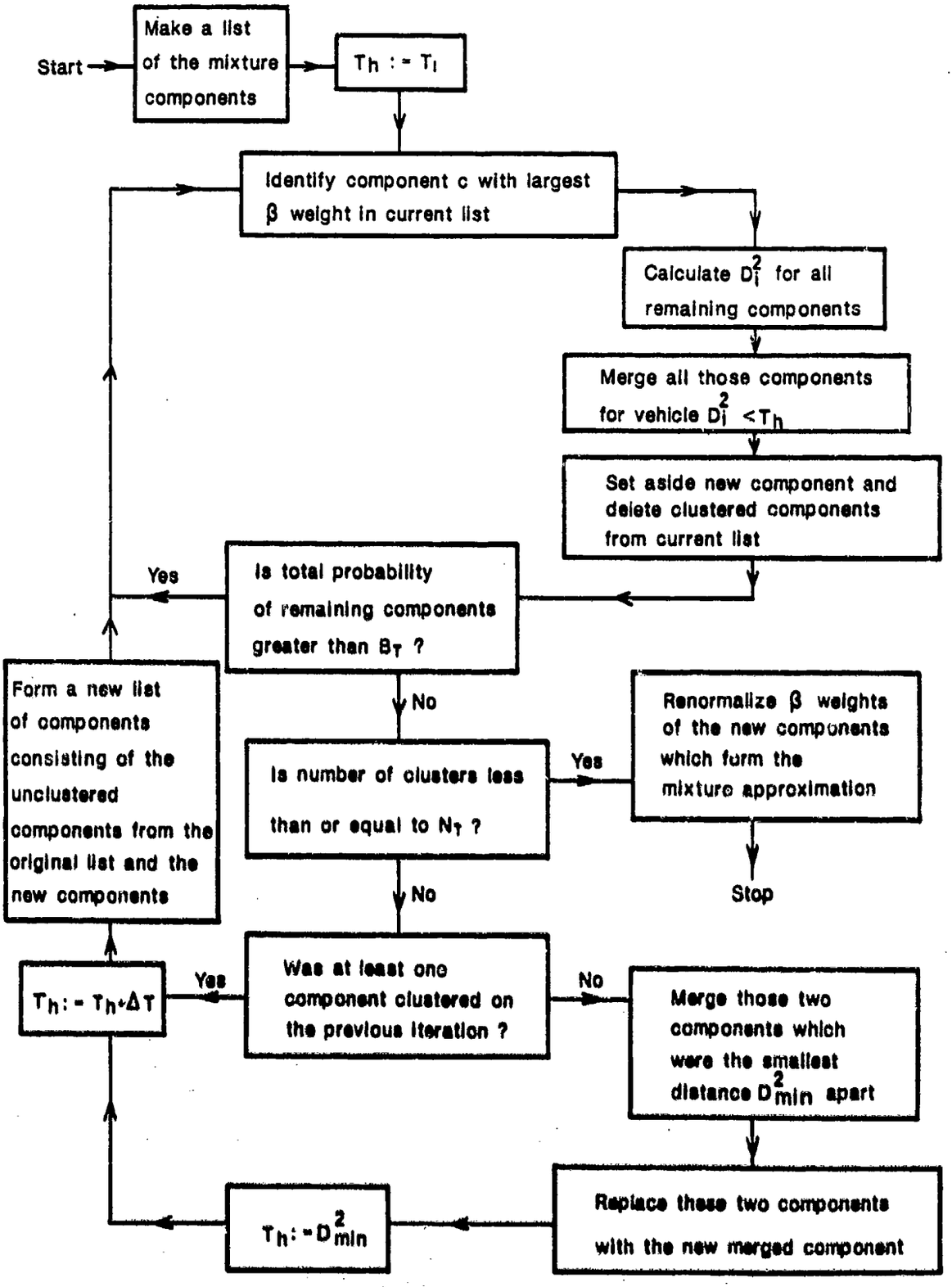


Fig 6 Flow diagram of the Clustering Algorithm

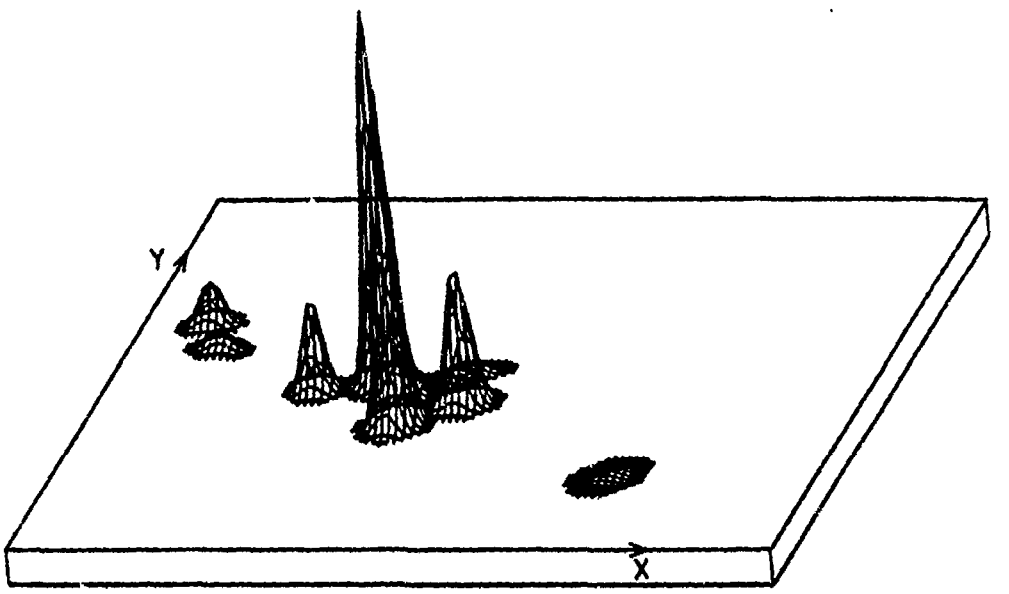


Fig 7 Clustering Algorithm approximation of mixture shown in Fig 2 with N_T = 10 (10 components)

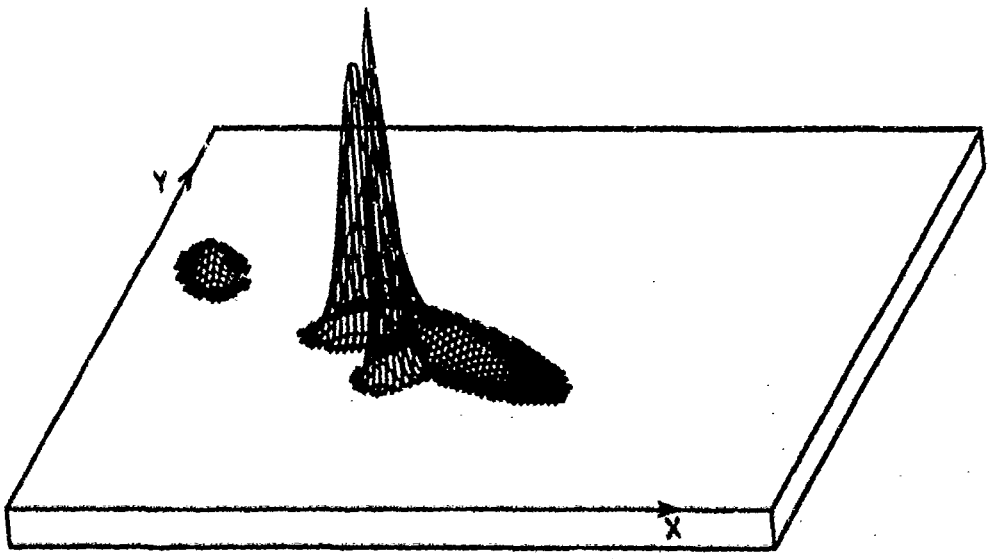


Fig 8 Clustering Algorithm approximation of mixture shown in Fig 2 with N_T = 4 (4 components

Fig 9

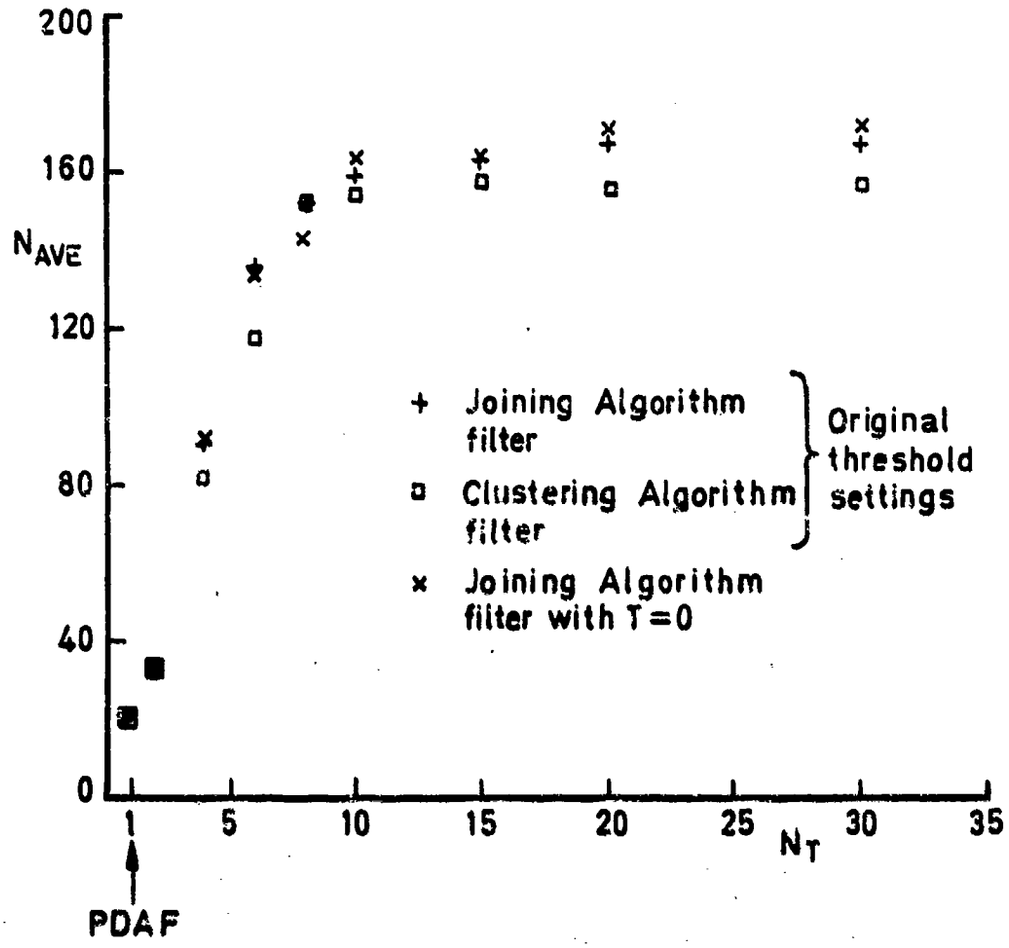
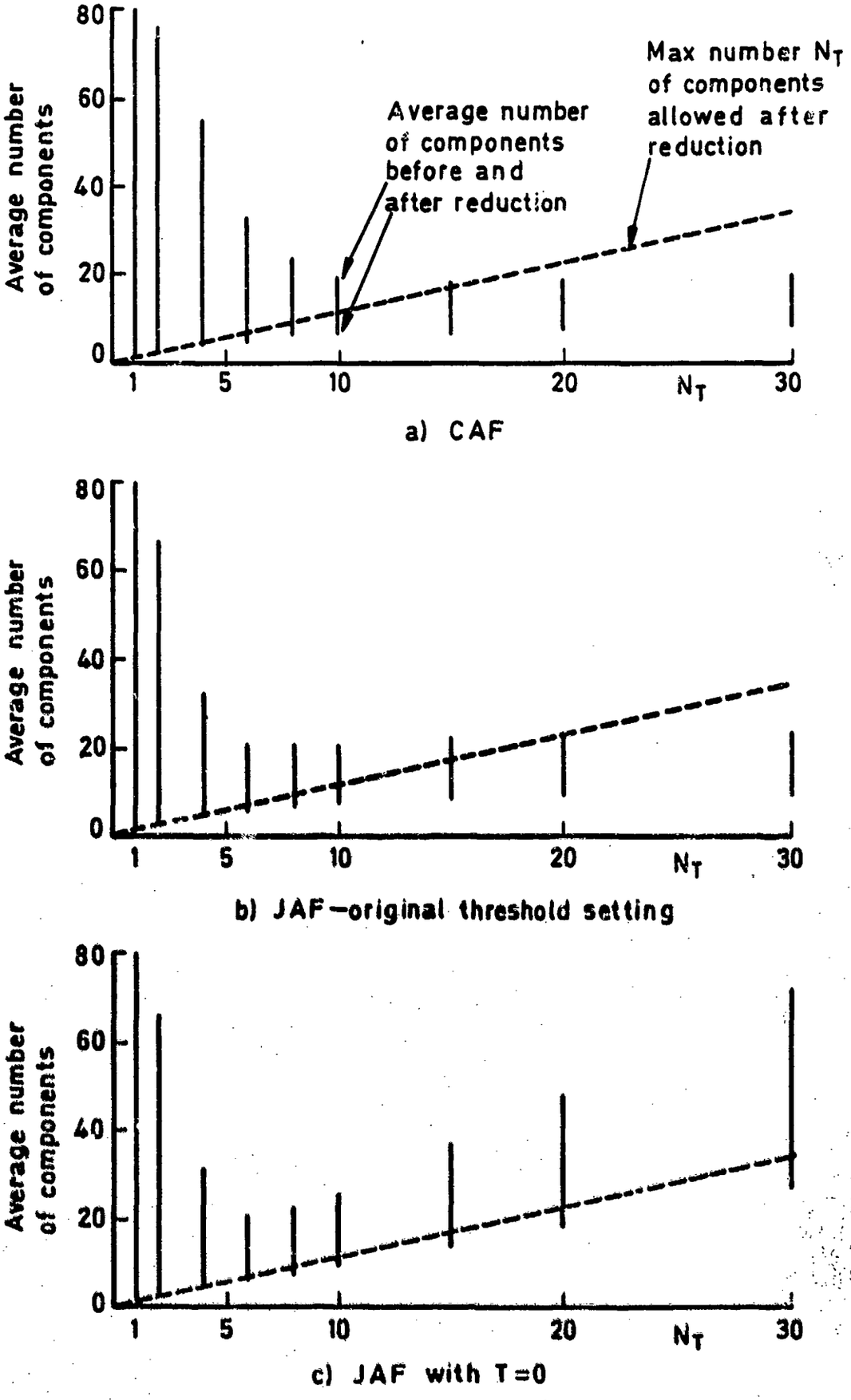


Fig 9 The average number of time steps until track loss as a function of N_T



TR 88004

Fig 10 Average number of mixture components before and after reduction

Fig 11

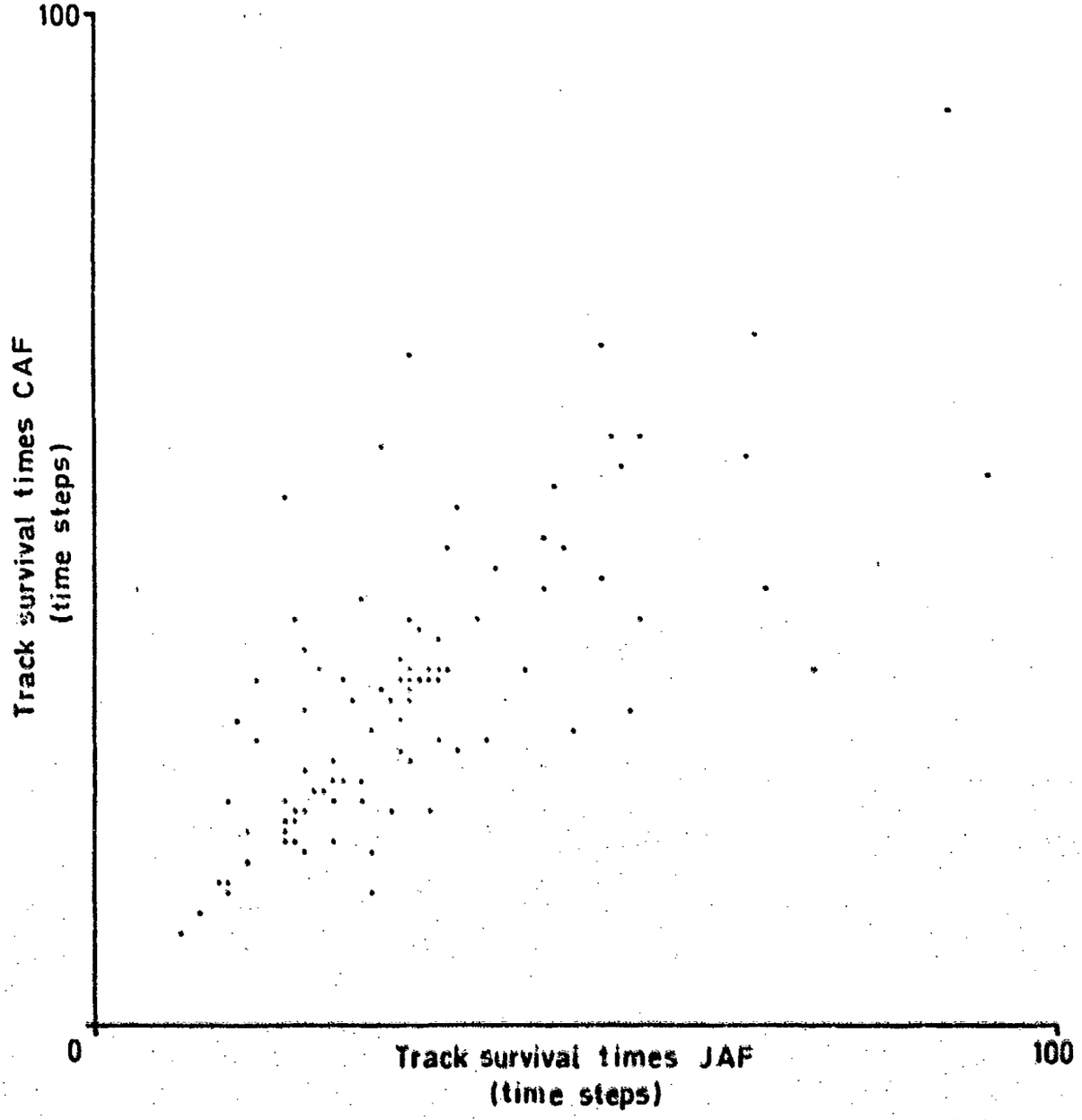
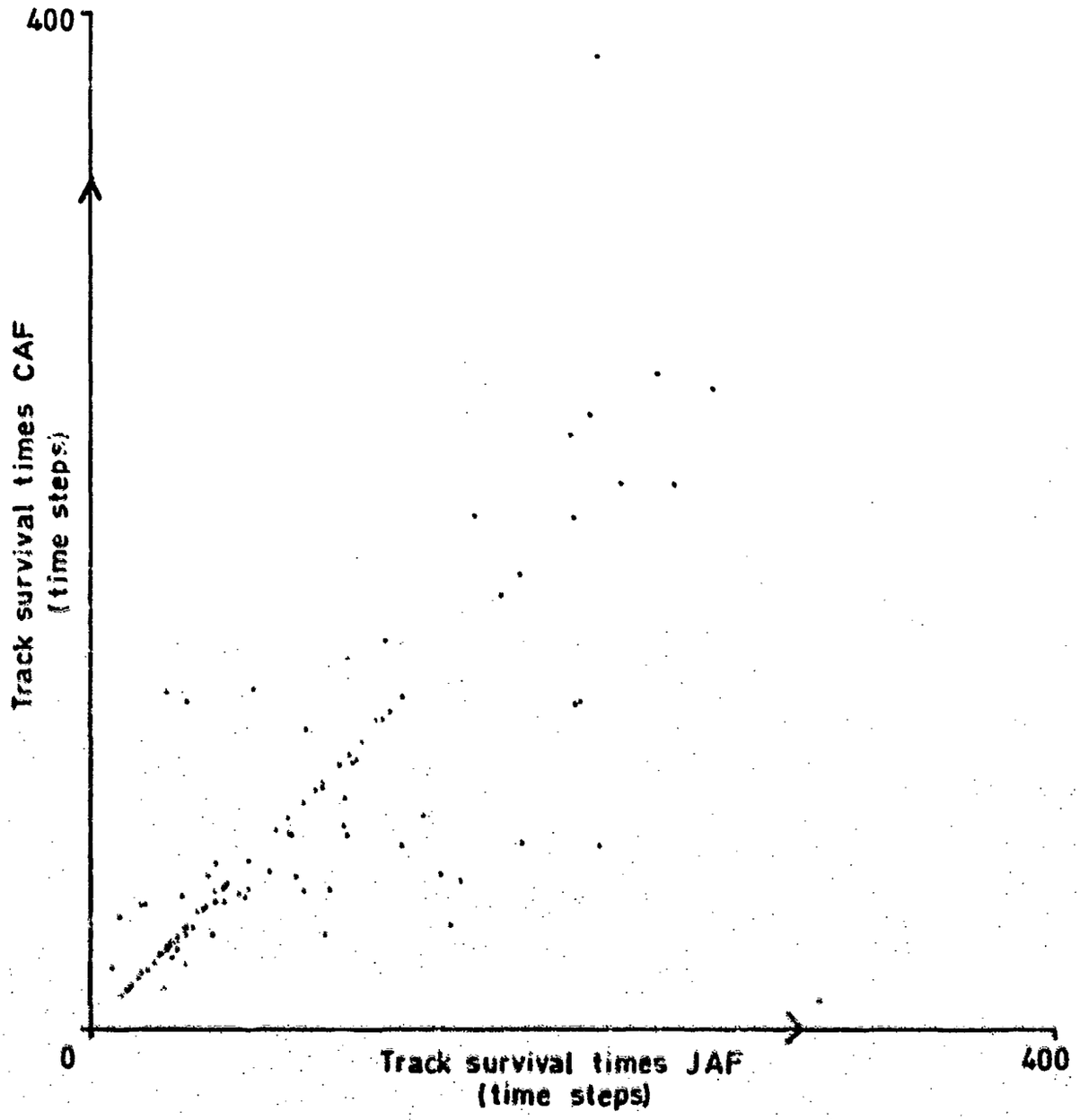


Fig 11 Track maintenance times for each replication $N_T = 2$

TR 88004



TR 88004

Fig 12 Track maintenance times for each replication $N_T = 4$

Fig 13

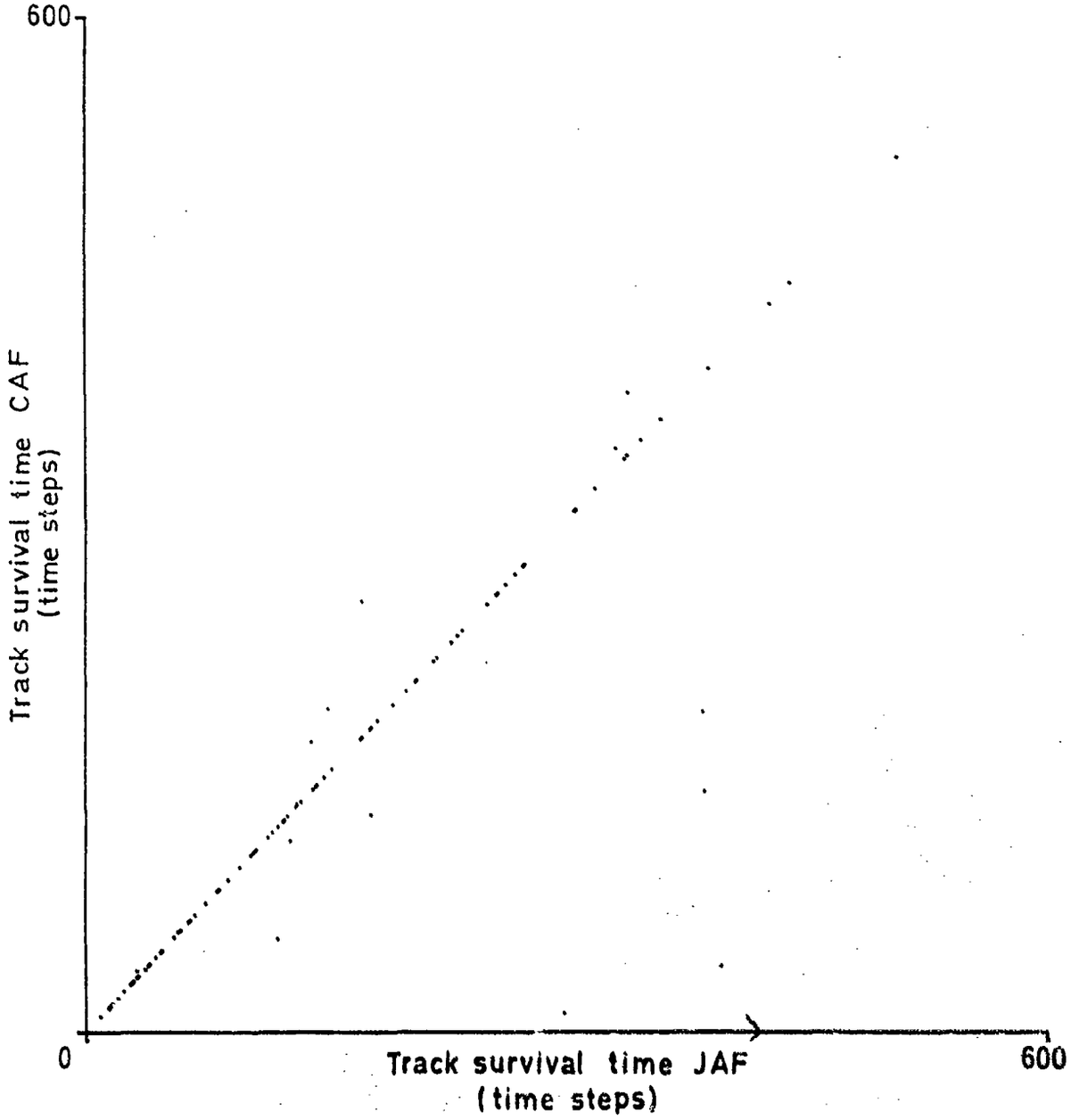
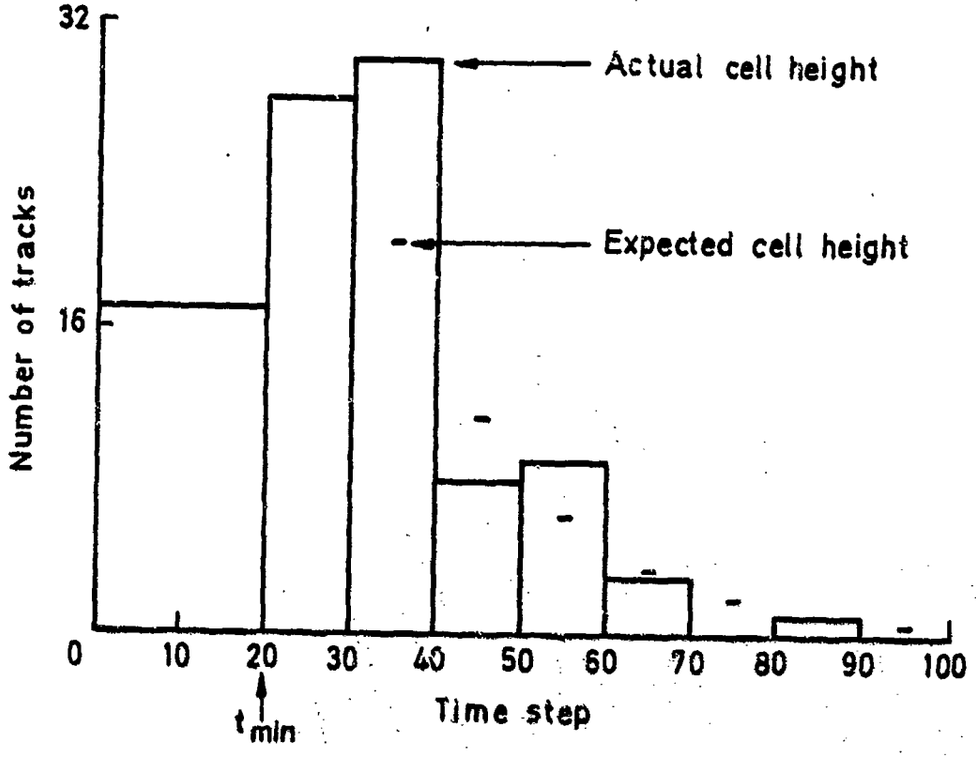
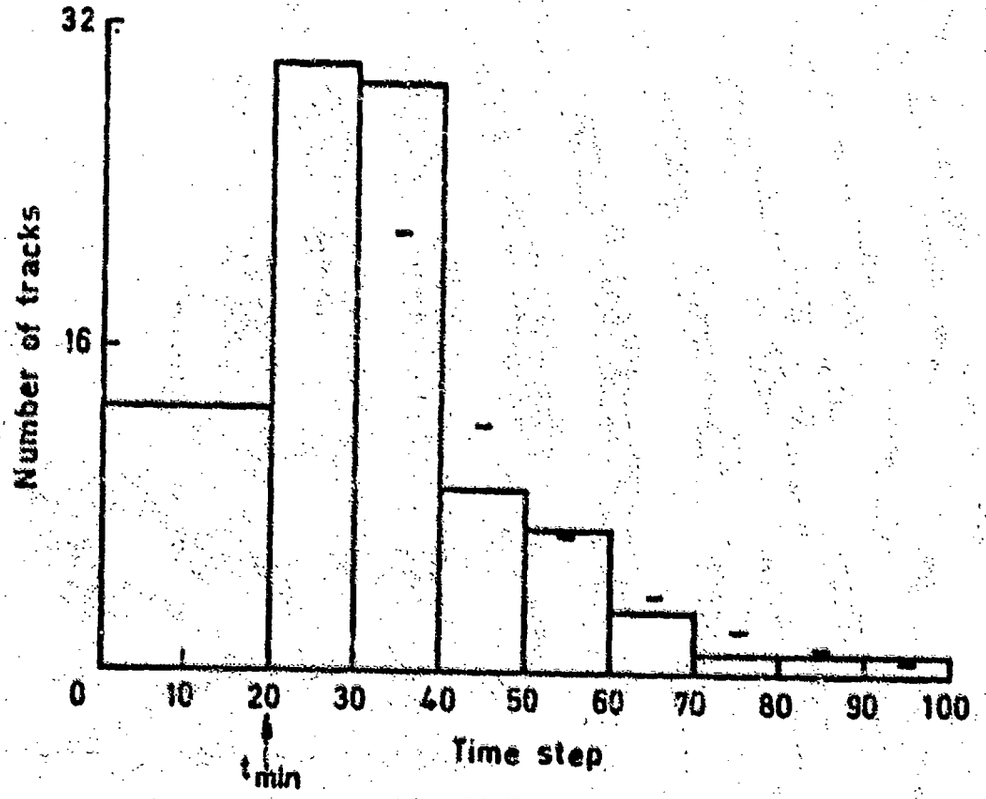


Fig 13 Track maintenance times for each replication $N_T = 30$



a) CAF $N_T = 2$

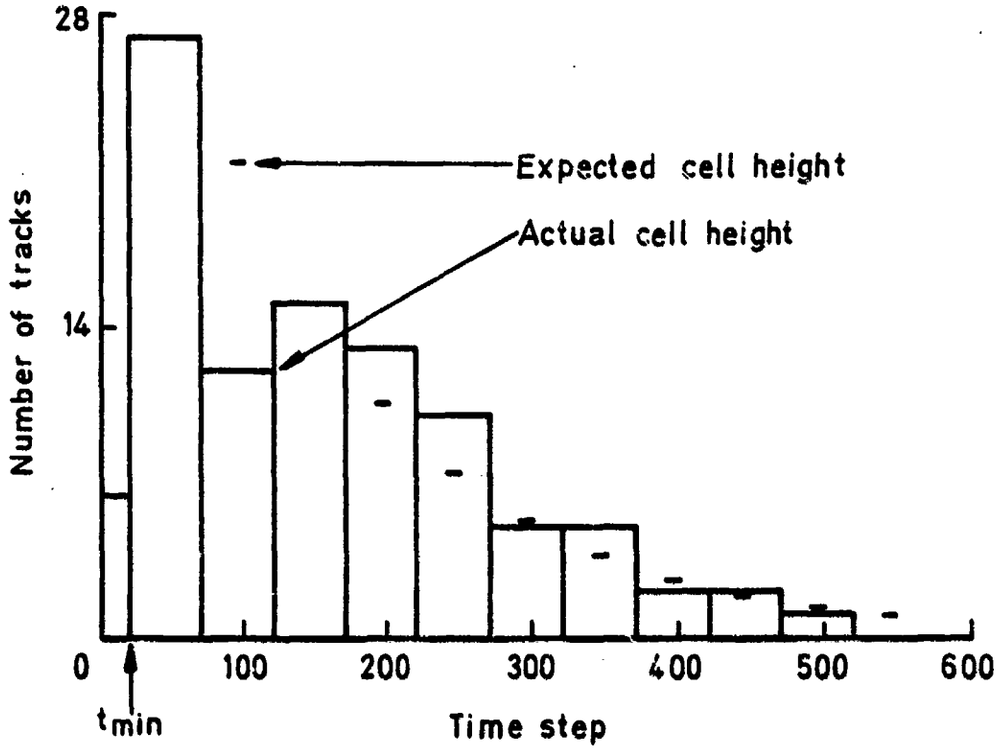


b) JAF $N_T = 2$

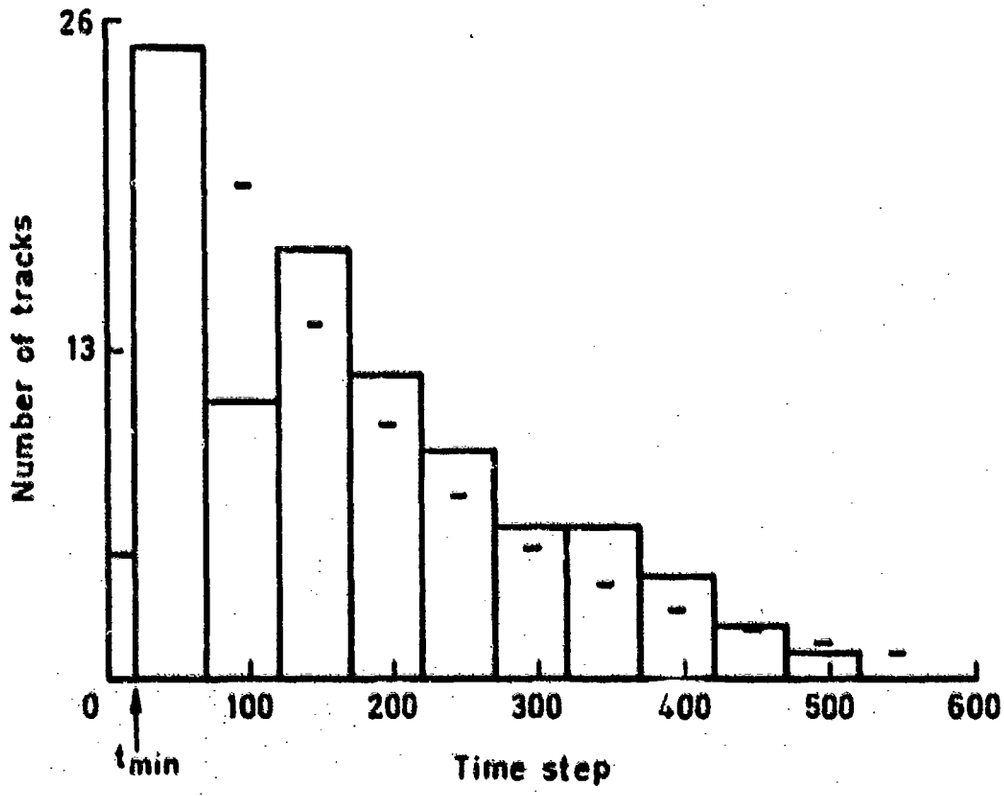
Fig 14 Histograms of track survival times for $N_T = 2$

TR 88004

Fig 15



a) CAF $N_T = 30$



b) JAF $N_T = 30$

Fig 15 Histograms of track survival times for $N_T = 30$

TR 88004

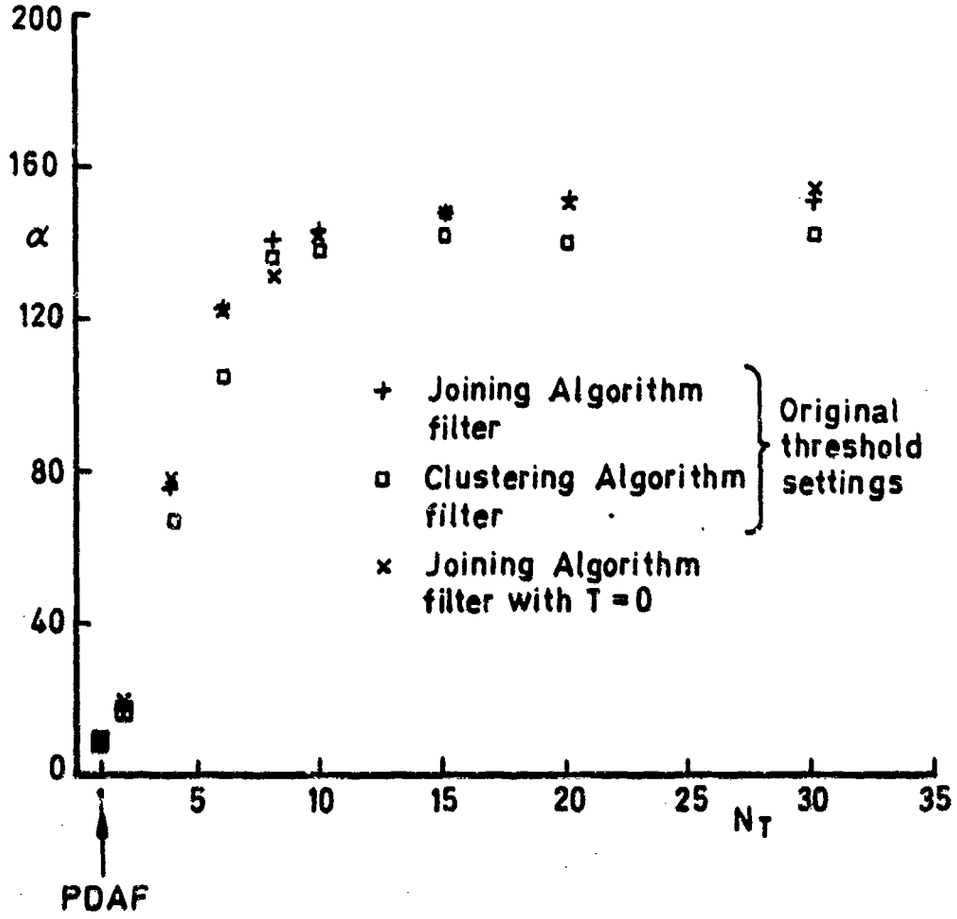
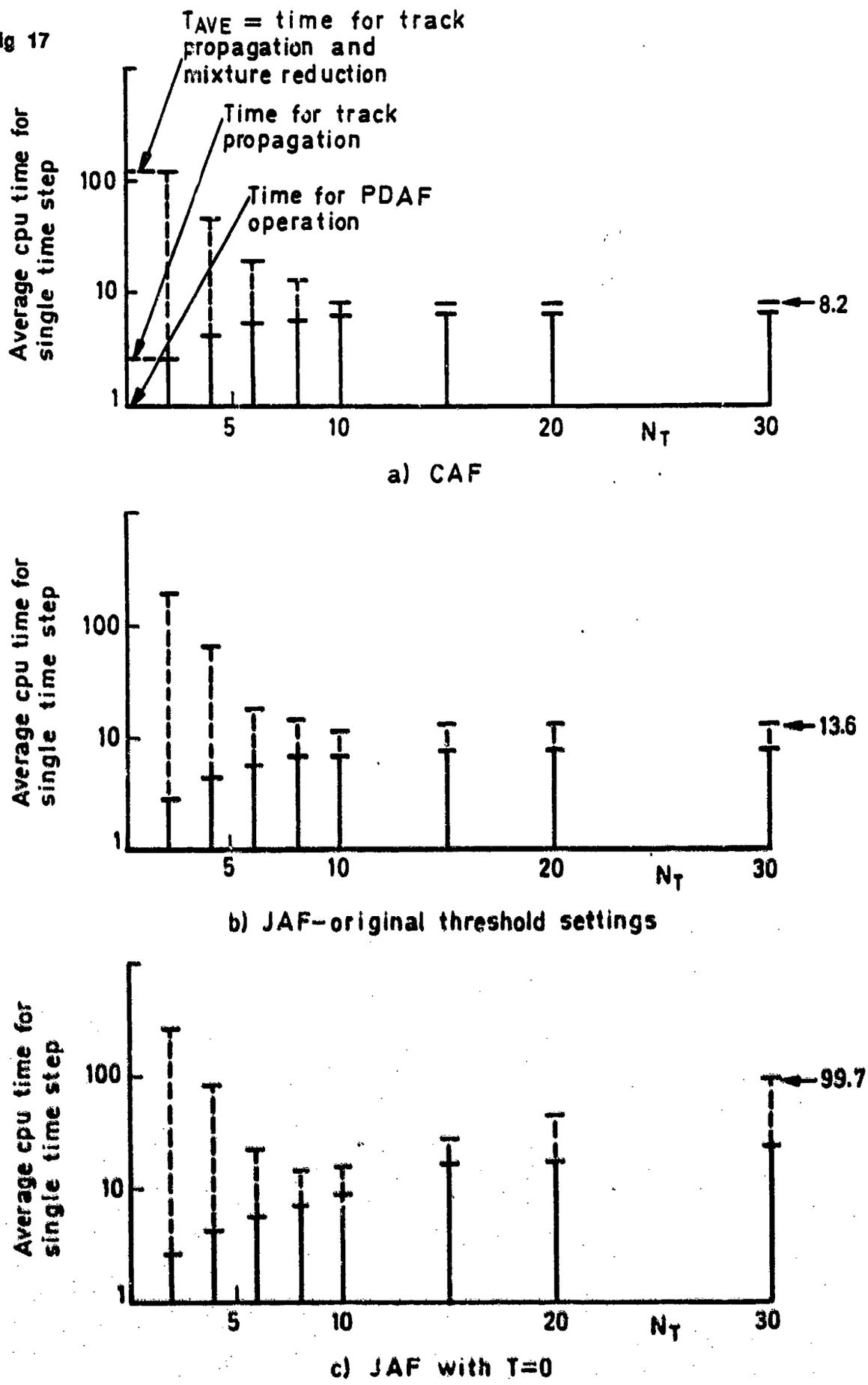


Fig 16 Estimates of average track survival times in steady state conditions

Fig 17



TR 88004

Fig 17 Average cpu time to perform a single time step

REPORT DOCUMENTATION PAGE

Overall security classification of this page

UNCLASSIFIED

As far as possible this page should contain only unclassified information. If it is necessary to enter classified information, the box above must be marked to indicate the classification, e.g. Restricted, Confidential or Secret.

1. DRIC Reference (to be added by DRIC)	2. Originator's Reference	3. Agency Reference	4. Report Security Classification/Marking
	RAE TR 88004		UNCLASSIFIED

5. DRIC Code for Originator	6. Originator (Corporate Author) Name and Location
7673000W	Royal Aerospace Establishment, Farnborough, Hants, UK
5a. Sponsoring Agency's Code	6a. Sponsoring Agency (Contract Authority) Name and Location

7. Title
Mixture reduction algorithms for uncertain tracking

7a. (For Translations) Title in Foreign Language

7b. (For Conference Papers) Title, Place and Date of Conference

8. Author 1. Surname, Initials	9a. Author 2	9b. Authors 3, 4	10. Date	Pages	Refs.
Salmond, D.J.			January 1988	58	11

11. Contract Number	12. Period	13. Project	14. Other Reference Nos.
			Attack Weapons 28

15. Distribution statement
(a) Controlled by –

(b) Special limitations (if any) –

If it is intended that a copy of this document shall be released overseas refer to RAE Leaflet No.3 to Supplement 6 of MOD Manual 4.

16. Descriptors (Keywords) (Descriptors marked * are selected from TEST)

Tracking filters. Mixture distributions. Bayesian filters. Kalman filters.

17. Abstract

Bayesian solutions of tracking problems that involve measurement association uncertainty, give rise to Gaussian mixture distributions, which are composed of an ever increasing number of components. To implement such a tracking filter, the growth of components must be controlled by approximating the mixture distribution. A popular and economical scheme is the Probabilistic Data Association Filter (PDAF), which reduces the mixture to a single Gaussian component at each time step. However, this approximation may destroy valuable information, especially if several significant, well spaced components are present.

In this Report two new algorithms for reducing Gaussian mixture distributions are presented. These techniques preserve the mean and covariance of the original mixture, and the final approximation is itself a Gaussian mixture. The reduction is achieved by successively merging components or groups of components. The two algorithms have been used to control the growth of components which occurs with the solution to the problem of tracking a single object, in the presence of uniformly distributed false measurements. Simulation results are presented which compare the performance of the resulting tracking filters and the PDAF.