capacity in the presence of failure uncertainties. In this paper, we have demonstrated the undesirable effect of actuator failures and the desirable effectiveness of failure compensation, by developing three failure compensation control schemes. This work is illustrated in a framework of application to vibration control (stabilization and regulation) of a rocket payload fairing structural-acoustic model with unknown actuator failures. The robust control failure compensation scheme is based on an LMI method, under an LMI design condition. The two adaptive control failure compensation schemes are based on robust adaptive control and failure parameterization methods, under a matrix rank condition. While all three schemes ensure signal boundedness, the failure parameterization based design is able to achieve asymptotic state regulation, in spite of the failure uncertainties. Simulation results verified the effectiveness of the developed failure compensation schemes.

XIDONG TANG
GANG TAO
LINGFENG WANG
Dept. of Electrical and Computer Engineering
University of Virginia
Charlottesville, VA 22903
E-mail: (gtgs@virginia.edu)

JOHN A. STANKOVIC
Dept. of Computer Science
University of Virginia
Charlottesville, VA 22903

REFERENCES

[1] Agarwala, R., and Ozcelik, S. (2000)
Active vibration control of a multi-degree-of-freedom structure by the use of direct model reference adaptive control.
In *Proceedings of the American Control Conference*, Chicago, IL, 2000, 3580–3584.

[2] Bodson, M., and Groszkiewicz, J. E. (1997)
Multivariable adaptive algorithms for reconfigurable flight control.
IEEE Transactions on Control System Technology, **5**, 2 (1997), 217–229.

[3] Boskovic, J. D., and Mehra, R. K. (1998)
A multiple model-based reconfigurable flight control system design.
In *Proceedings of the 37th IEEE Conference on Decision and Control*, 1998, 4503–4508.

[4] Eaton, D. C. G. (1997)
An overview of structural acoustics and related high-frequency-vibration activities.
ESA Bulletin, number 92 (Nov. 1997).

[5] Forrai, A., Hashimoto, S., Funato, H., and Kamiyama, K. (2000)
Fuzzy logic based vibration suppression control of flexible structure.
In *Proceedings of the 6th International Workshop on Advanced Motion Control*, Nagoya, Japan, 2000, 378–383.

[6] Gao, Z., and Antsaklis, P. J. (1991)
Stability of the pseudo-inverse method for reconfigurable control systems.
International Journal of Control, **53**, 3 (1991), 717–729.

[7] Glaese, R. M., and Anderson, E. H. (1999)
Active structural-acoustic control for composite payload fairings.
www.csaengineering.com/technicalpaperpdfs/CSA1999_ActiveStructuralAcoustic.pdf.

[8] Griffin, S., Lane, S. A., Hansen, C., and Cazzolato, B. (2001)
Active structural-acoustic control of a rocket fairing using proof-mass actuators.
Journal of Spacecraft and Rockets, **38**, 2 (2001), 219–225.

[9] Ioannou, P. A., and Sun, J. (1996)
Robust Adaptive Control.
Upper Saddle River, NJ: Prentice-Hall, 1996.

[10] Lam, J., and Cao, Y-Y. (1999)
Simultaneous linear-quadratic optimal control design via static output feedback.
International Journal of Robust and Nonlinear Control, **9** (1999), 551–558.

[11] Leo, D. J., and Anderson, E. H. (1998)
Vibroacoustic modeling of a launch vehicle payload fairing for active acoustic control.
In *Proceedings of Adaptive Structure Forum*, Long Beach, CA; paper AIAA-98-2086.

[12] Lewis, F. L. (1992)
Applied Optimal Control and Estimation.
Upper Saddle River, NJ: Prentice-Hall, 1992.

[13] Subramonian, V., and Gill, C. (2001)
Towards a pattern language for networked embedded software technology middleware.
In *Proceedings of the 16th Annual Conference on Object-Oriented Programming, Systems, Languages, and Applications*, Florida, Oct. 2001.

[14] Tao, G., Chen, S. H., Tang, X. D., and Joshi, S. M. (2004)
Adaptive Control of Systems with Actuator Failures.
New York: Springer, Mar. 2004.

[15] Wang, H., Huang, Z. J., and Daley, S. (1997)
On the use of adaptive updating rules for actuator and sensor fault diagnosis.
Automatica, **33** (1997), 217–225.

Tracking with Distributed Sets of Proximity Sensors using Geometric Invariants

We propose a new approach to forming an estimate of a target track in a distributed sensor system using very limited sensor information. This approach uses a central fusion system that collects only the peak energy information from each sensor and assumes that the energy attenuates as a power law in range from the source. A geometrical invariance property of the proximity of the distributed sensors relative to a target track is used to generate potential target track paths. Numerical simulation examples are presented to illustrate the practicality of the technique.

Manuscript received March 2, 2004; revised July 7, 2004; released for publication July 27, 2004.

IEEE Log No. T-AES/40/4/839862.

Refereeing of this contribution was handled by P. K. Willett.

I. INTRODUCTION

The problem of determining a target track from a spatially distributed set of sensors using limited (passive) sensor information has received considerable attention. Many applications in tracking and surveillance benefit from the increased coverage provided by a distributed set of sensors over that provided by a single sensor platform. To make distributed systems practical, each sensor must be relatively simple. In that context, we are concerned with proximity sensors; that is, sensors that only report a simple energy observation, from which a relative distance measurement to the target may be inferred. In this way, we expect to derive an architecture that has very low communications bandwidth and is very robust in its performance across a variety of target types. We consider a central fusion center that fuses only the limited energy information of the proximity sensors. The methods developed are useful in multiple application areas with little modification.

The use of fusion systems to combine information from multiple detecting sensors has received much attention, beginning with the observation that a centralized host applying a Neyman-Pearson test combined with likelihood ratio tests at the sensors provides the optimal detection decision [1]. Further efforts have shown the details of these optimal decision schemes in [2], [3], [4], [5], and [6]. These methods are attractive since they can be shown to produce optimal detection decisions under the constraints of the problem assumptions. However, such methods require a priori information about sensor probability of detection and false alarm. Since we are interested in proximity sensors that have no a priori knowledge of the target track, the central processor only knows as much about the target as can be inferred from the proximity reports. Some efforts at developing learning approaches to obtain these probabilities have been made (for example, see [7], [8], [9]), however, these approaches are based on a set of many sensors observing the same phenomenon over the same time interval, and rely on a large observation space to derive probabilities. Since the proximity information is very limited in distributed sensor fields, these methods do not apply to the problem at hand.

We limit our data to only the proximity information at the closest-point-of-approach (CPA) for each sensor, and thus do not have sequential observations for multiple sensors available to the fusion center. Using geometric invariants, we show that this CPA information is enough to compute a reduced set of potential target track paths. We further illustrate that this limited-information approach combined with track path clustering is robust to false alerts from sensors, making it amenable to tracking targets of varying unknown characteristics. This initial analysis considers a single target only, moving at a constant speed and heading through the sensor field, and maintaining a constant source amplitude. Furthermore the field of sensors is assumed fixed, sensor positions are known, and the attenuation characteristics due to the environment are known and fixed. Extensions to multiple targets, maneuvering targets, and variable environments are all subjects of future work.

II. TRACK DETERMINATION ALGORITHM FOR ERROR-FREE MEASUREMENTS

We consider an event-based target track path determination algorithm. This algorithm assumes that each sensor receives a signal from the same target, and the received signal is isotropic energy attenuated via a power law from that target. In particular,

$$y_j(t) = cFr_j^{-\alpha}(t) \tag{1}$$

is the signal received from the target, where $r_j(t)$ is the range (distance) from the jth sensor to the target, F represents a target source level that is independent of both time (independent over the track observation interval) and sensor location, and c is a target-independent scaling constant that is based on the physics of the problem. The nondimensional exponential attenuation coefficient α depends on the particular physical mechanism of the energy that is being received. For example, the geometrical spreading of a power density (in three dimensions) implies $\alpha = 2$. Equation (1) is the proximity sensor model used throughout this paper. Many physical phenomena (such as magnetics, acoustics, optics) that are governed by linear wave models based on energy conservation follow (1) with different values of α. The value of α depends on both the physical mechanism and the environment; errors in the assumed value of α have the effect of exaggerating the impact of measurement errors in the tracking algorithm.

The autonomous sensors in this notional distributed sensor system each perform their own detection processing to determine when a detection event occurs. As a limiting case of minimal information transfer (and hence small communications bandwidth) between the sensors and the central track determination processor, we consider the situation where each sensor reports only its location (in two-dimensional space) and a single value for the received signal level at the sensor-to-target CPA, given by

$$y_{j,\mathrm{CPA}} = y_j(t)|_{t_{j,\mathrm{CPA}}} \tag{2}$$

where $t_{j,\mathrm{CPA}}$ is the time of CPA for the jth sensor. The CPA time is not reported to the central processor. The density of the sensor field is assumed to be large enough so that multiple detections from a single target occur within a specified time period (at the central

processor) and the target does not maneuver within that time period.

The host processor takes the measured values of $y_{j,\mathrm{CPA}}$ and computes a parameter β_j for each reporting sensor, given by

$$\begin{aligned}\beta_j &= [y_{j,\mathrm{CPA}}]^{-1/\alpha} \\ &= S \cdot r_j(t_{\mathrm{CPA}}) > 0 \end{aligned} \tag{3}$$

where $y_{j,\mathrm{CPA}}$ is as shown in (2) and $S = (cF)^{-1/\alpha}$ is a scaling factor that is both range and sensor invariant. It is clear that the β_j parameters are measurements of a range-to-CPA scaled by a constant which is an unknown, but is an invariant of the target under consideration. From these values of β_j, the host processor determines the most likely target track path. If the value of α is in error, any errors in the detected CPA level $y_{j,\mathrm{CPA}}$ will be exaggerated in the error of the reported value of β_j.

LEMMA 1 *Given a set of $\beta_j = S \cdot r_j(t)$ error-free measurements at sensor CPA for sensors located at $\mathbf{x}_j$, the target track path of a constant heading and speed target is a line that is jointly tangent to all circles $C_j(S) = \{\mathbf{x} : \|\mathbf{x} - \mathbf{x}_j\| \le \beta_j/S\}$ for a fixed value of S.*

This lemma is a direct extension of the notion of CPA for a set of multiple sensors. It is important to note that the circles $C_j(S)$ defined in the lemma are all sized according to the same value of S. Thus, if a set of circles are drawn according to a fixed value of $S = S_0$, there may not be a line tangent to them all; however, the circles can be scaled simultaneously so that a line can be placed jointly tangent to all of them. This lemma is for error-free measurements; the extension to measurements with errors is given in the next section.

The track path estimation algorithm is based on having a small number of potential track paths for every set of three measurements. The algorithm is summarized as follows.

Track Estimation Algorithm for Error-Free Measurements:

1) Enumerate all possible groups of three detections.

2) Determine all potential track paths for each group of three detections.

3) Choose the track path that occurs most prevalently as the estimate.

The rationale for selecting groups of three tracks is based on an extension of a classical geometry theorem of D'Alembert [10]. D'Alembert stated the collinearity properties of interior and exterior points between circles in (projective) planar geometry which is a central component of this tracking approach. We define these points in the context of sensor detections as follows.

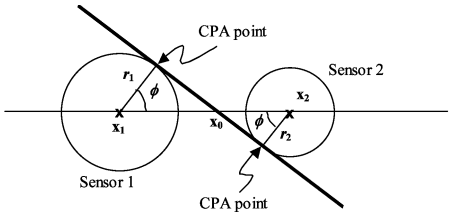

Fig. 1. Geometry of potential track that crosses between two sensors (interior).

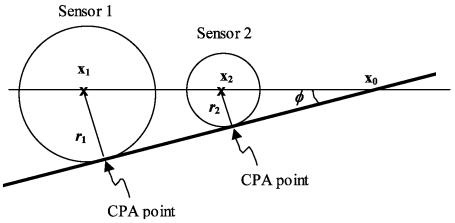

Fig. 2. Geometry of potential track that crosses outside two sensors (exterior).

Consider two sensor detection events at locations $\mathbf{x}_1$ and $\mathbf{x}_2$ as shown in Fig. 1. If the circles in the figure are drawn with radius $r_j = \beta_j/S$, where β_j is the β-measurement from the jth sensor and S is an arbitrary scaling parameter, a potential track path corresponding to this value of S is given by the line jointly tangent to the two circles (as per Lemma 1). Define the point $\mathbf{x}_0$ to be the point where the track path intersects the line that connects the two sensor locations. Form the triangle between $\mathbf{x}_1$, $\mathbf{x}_0$ and the CPA point from sensor 1. From this triangle the identity

$$r_1 = \beta_1/S = \|\mathbf{x}_1 - \mathbf{x}_0\| \cos\phi \tag{4}$$

is formed, with a similar identity for r_2. Combining these two relationships and solving for the point $\mathbf{x}_0$ yields

$$\mathbf{x}_{12}^{\mathrm{int}} \equiv \mathbf{x}_0 = \mathbf{x}_1 \left(\frac{\beta_2}{\beta_1 + \beta_2}\right) + \mathbf{x}_2 \left(\frac{\beta_1}{\beta_1 + \beta_2}\right) \tag{5}$$

which shows that the location of the crossing point $\mathbf{x}_0$ is independent of the scaling parameter S as well as any common proportionality factors in the βs (including units). The unique point defined by (5) is defined as the interior point of sensors 1 and 2.

While the track path shown in Fig. 1 crosses between the two sensor locations, there are other track paths that cross outside of the two sensor locations. Consider two sensor detection events at locations $\mathbf{x}_1$ and $\mathbf{x}_2$ as shown in Fig. 2. If the circles in the figure are drawn with radius $r_j = \beta_j/S$, where β_j is the β-measurement from the jth sensor and S is an arbitrary scaling parameter, a potential track path exterior to the two sensors is given by the heavy line that is drawn jointly tangent to the circles. As for the interior case, we define the point $\mathbf{x}_0$ to be the point where the track path intersects the line that connects

the two sensor locations. From Fig. 2, we form two triangles for the case of an exterior crossing, yielding

$$r_1 = \beta_1/S = \|\mathbf{x}_1 - \mathbf{x}_0\| \sin\phi \tag{6}$$

with a similar expression for r_2. Combining and solving for the point $\mathbf{x}_0$ yields

$$\mathbf{x}_{12}^{\text{ext}} \equiv \mathbf{x}_0 = \mathbf{x}_1 \left(\frac{\beta_2}{\beta_2 - \beta_1} \right) + \mathbf{x}_2 \left(\frac{\beta_1}{\beta_1 - \beta_2} \right) \tag{7}$$

for the exterior crossing point $\mathbf{x}_0$ which is again independent of the scaling parameter S and any proportionality factors in the βs, including units. This point is furthermore independent of which side of the circles the track path line is drawn. The unique point defined by (7) is defined as the exterior point of sensors 1 and 2.

For the degenerate case of $\beta_1 = \beta_2$, the exterior point defined by (7) is indeterminate. In this case, a point on the line connecting $\mathbf{x}_1$ and $\mathbf{x}_2$ at a distance that is infinitely far from either point is consistent with the definition of "the point at infinity" in projective geometry. For the purposes of tracking, however, the track paths are still well defined in this case.

LEMMA 2 *All target track paths corresponding to two sensor detection events must pass through either the interior point or the exterior point of the two sensors.*

This lemma is verified by recognizing that only four lines are jointly tangent to any two nonoverlapping circles. Two of these lines go through the interior point (by construction of an interior point) while the other two go through the exterior point (also by construction).

Finally, define the sensor triangle formed by connecting the sensor locations corresponding to the three detection events. For all sets of three noncollinear sensor detection events, the sensor triangle is well defined with positive area. A sketch of the sensor triangle and the set of interior and exterior points for three typical sensor detections is shown in Fig. 3. The circles on the figure are centered at the sensor locations and the circle sizes have been drawn so the jth circle's radius is proportional to β_j. We now state the theorem upon which the tracking algorithm is based.

THEOREM 1 *For three spatially separated and noncollinear detection events at locations* $\mathbf{x}_j$ *with exact measurements* β_j*, there are exactly four linear track paths that meet the condition in Lemma* 1.

PROOF The proof of the theorem is based on the property of the invariant S as a circle size scaling parameter. For a line to correspond to a track path, it must cross all three sides of the sensor triangle for a single value of S (the case of a track path parallel to one of the sides is a simple extension where the crossing occurs at an exterior point at ∞). By

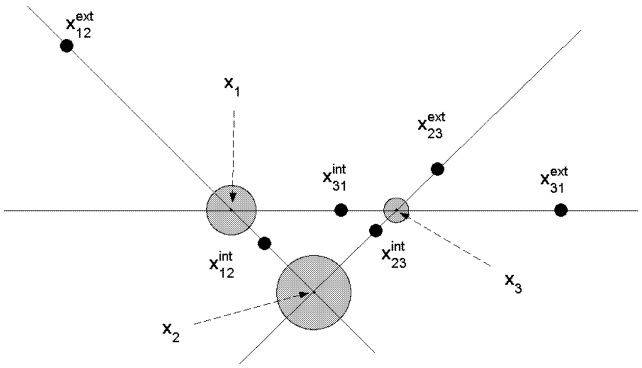

Fig. 3. Geometry of sensor triangle, interior points, and exterior points for set of three sensor detections (detection circle size scaled by β_j).

Lemma 2, the crossing of the track path with each side of the sensor triangle must occur at either an interior point or an exterior point. Consider a track that crosses two of the exterior points. Since such a line cannot intersect the interior of the sensor triangle (geometrically impossible since each exterior point is on a separate line that is collinear with a different side of the triangle), it must intersect the third exterior point. For such a track to be valid, the three exterior points must be collinear. We form the two vectors

$$\mathbf{v}_A = (\mathbf{x}_{23}^{\text{ext}} - \mathbf{x}_{12}^{\text{ext}}), \qquad \mathbf{v}_B = (\mathbf{x}_{31}^{\text{ext}} - \mathbf{x}_{12}^{\text{ext}}) \tag{8}$$

which are linearly dependent if and only if the three exterior points $\{\mathbf{x}_{12}^{\text{ext}}, \mathbf{x}_{23}^{\text{ext}}, \mathbf{x}_{31}^{\text{ext}}\}$ are collinear. This is readily verified and, thus, there is exactly one line that includes at least two of the exterior points, and that line includes all three of the exterior points.

Since all track paths must contain either $\mathbf{x}_{ij}^{\text{ext}}$ or $\mathbf{x}_{ij}^{\text{int}}$ for each of the three i-j combinations, and the combinations with at least two exterior points have been enumerated (to one track path), we are left with combinations that contain at least two interior points. We again apply the notion of three points being collinear if and only if the vectors between any two pairs of the points are linearly dependent. It is algebraically shown that the following sets of triples are all of the collinear sets with two interior points and one exterior point:

$$\{(\mathbf{x}_{12}^{\text{int}}, \mathbf{x}_{23}^{\text{int}}, \mathbf{x}_{31}^{\text{ext}}), (\mathbf{x}_{23}^{\text{int}}, \mathbf{x}_{31}^{\text{int}}, \mathbf{x}_{12}^{\text{ext}}), (\mathbf{x}_{31}^{\text{int}}, \mathbf{x}_{12}^{\text{int}}, \mathbf{x}_{23}^{\text{ext}})\}.$$

Furthermore, it is similarly shown that the set of three interior points are not collinear.

We have enumerated all of the cases of potential track paths, based on the requirements of Lemma 2, by searching for collinear combinations of points interior and exterior points on the sensor triangle, and this enumeration has yielded exactly four lines in the general case. This enumeration thus completes the proof.

We note that the degenerate case of $\beta_1 = \beta_2$ that was previously mentioned does not impact the

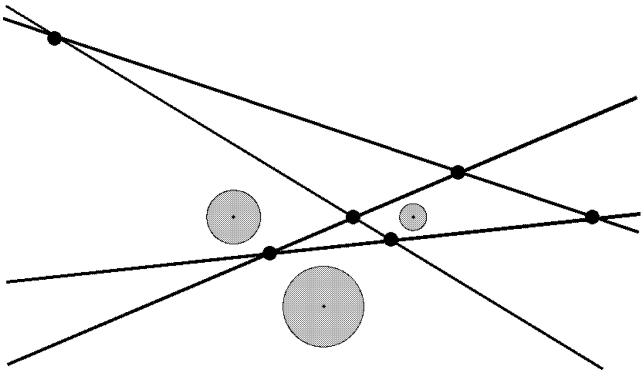

Fig. 4. Potential track path locations for set of three detections from Fig. 3.

construction of track paths. In this case, the exterior point from those two sensors ($\mathbf{x}_{12}^{\text{ext}}$) corresponds to "the point at infinity," yet the remaining two points for each path that contains this point are well known. Thus, the construction of the path that contains ($\mathbf{x}_{23}^{\text{int}}, \mathbf{x}_{31}^{\text{int}}, \mathbf{x}_{12}^{\text{ext}}$) is given by the line that connects ($\mathbf{x}_{23}^{\text{int}}, \mathbf{x}_{31}^{\text{int}}$) and the path that contains all three exterior points is given by the line that connects ($\mathbf{x}_{23}^{\text{ext}}, \mathbf{x}_{31}^{\text{ext}}$). The degenerate case of $\beta_1 = \beta_2 = \beta_3$ cannot be resolved in this manner, in which case the path that corresponds to all three exterior points is not used (only three paths are thus generated in this case).

Fig. 4 illustrates an example of the four tracks that are determined from a set of three sensors, using the sensor detections as shown in Fig. 3. For more than three sensor reports, we create all of the combinations of three sensors reports and generate four potential track paths for each combination. Thus, for N total detection reports from sensors, we create

$$N_{\text{tracks}} = 4 \cdot \binom{N}{3} = \frac{2N!}{3(N-3)!} = \tfrac{2}{3}N(N-1)(N-2) \tag{9}$$

potential tracks. Assuming there exists a "true" track path from which the detections were generated, the set of track paths corresponding to each set of three sensor detections contains one path that lines up with this true track path. This path is readily identified since it is consistent within each set of three sensor detections, and the other paths will generally be distributed somewhat sporadically. In the case of measurement errors, the track paths do not lie directly on top of one another, but are clustered about a neighborhood of the correct track path.

III. CLUSTERING OF POTENTIAL TRACK PATHS WITH MEASUREMENT ERRORS

Both measurement errors and false detection alerts cause deviations from the perfect alignment of track paths described in the previous section. However, by using an automated clustering algorithm applied to the track path locations, we reduce the large number of track paths (cf. (9)) to those track paths that lie in geometrical clusters and then can assign a nominal (or average) track path to each cluster. The size of these geometrical clusters is based upon the expected target speed and the window of time that the central processor uses to collect sensor detection information. The results of clustering shown in this section were all generated from computer simulations.

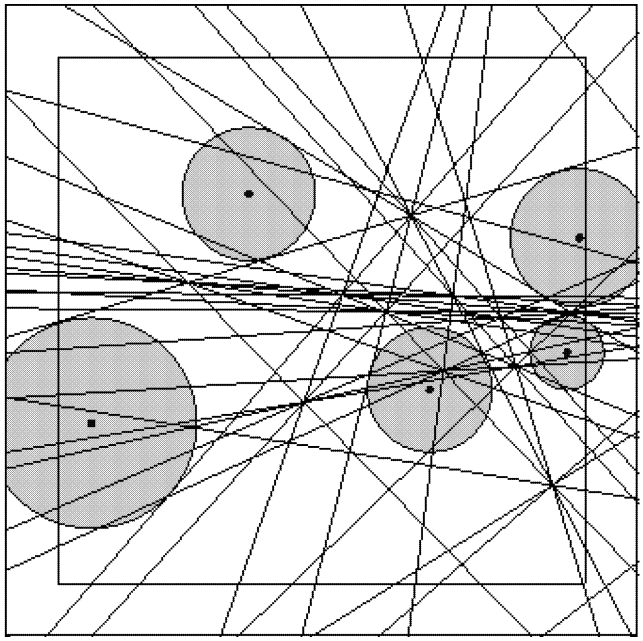

Fig. 5. Detection circles and potential track paths for set of 5 true detections.

We form all of the potential tracks for each group of three detections and then search for a cluster of many tracks near each other. These considerations yield a track path determination algorithm as follows.

Track Estimation Algorithm for Measurements with Errors:

1) Enumerate all possible groups of three detections.

2) Determine all potential track paths for each group of three detections.

3) Perform cluster analysis on the set of potential track paths.

4) Choose the largest cluster of potential tracks.

5) Average the tracks within the cluster to determine the estimated track path.

In Fig. 5, the set of $N_{\text{tracks}} = 40$ potential track paths for a set of 5 detections is shown. The actual track path in this simulation is a horizontal path in the center of the figure. In this figure, each sensor detection event is represented as a circle of a size scaled by β_j. The values of β_j were all modified to account for measurement error by multiplying the true value by a random number from a normal distribution with mean of one and a standard deviation

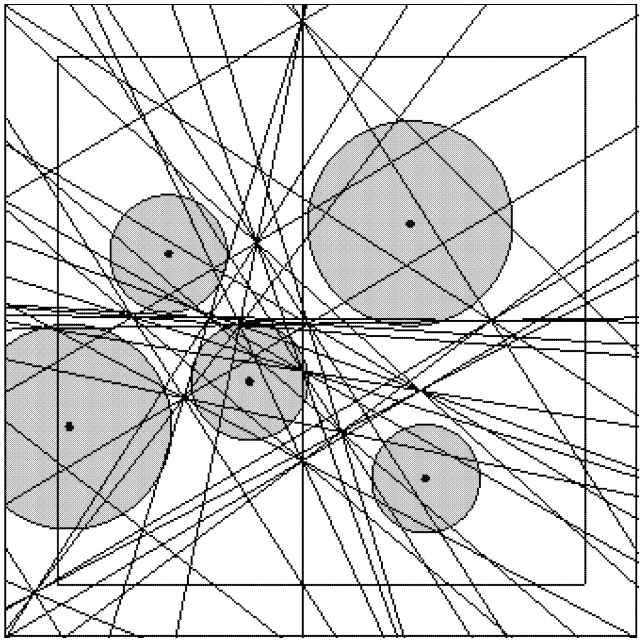

Fig. 6(a). Detection circles and potential track paths for set of 4 true detections plus 1 false alert. False alert is detection circle in lower right-hand section.

of 0.2 (20% random errors). For each group of three detections that are "true" (not false alerts), one of the four potential track paths drawn will be within a neighborhood of the correct track. This track path will be consistent between all groups of three true detections (within the error in the reported values of β_j), whereas other track paths will be distributed sporadically. This cluster is obvious as the nearly horizontal lines in the center of Fig. 5.

By applying a clustering approach, we reduce the impact of false alerts by only looking at a consistency of the detection reports from all of the groups of three. False alerts are represented in the simulation by a random value of β_j. Fig. 6(a) is a set of track paths for 5 detections where 1 of the detections is a false alert. Once again the actual track path is a horizontal path in the center of the figure. Here the clustered track path is not as obvious as in Fig. 5, but is still in the proper location and not impacted by the false alert. Note that additional clusters are obvious in the case with added false alerts.

With many potential track paths as shown by (9), it is desirable, and perhaps even necessary, to have an automated procedure to find clusters of potential track paths and also eliminate track paths that do not belong in a cluster. To cluster groups of lines (potential track paths) in an automated manner, the lines are first mapped to points in a two-dimensional space. Since the sensors cover a finite area, we consider a large circle Γ that surrounds the area of the detection events. Every track path under consideration must enter and exit the area, so it will generically intersect the circle Γ at two points (track paths that don't enter and exit the area are of no interest to the sensor system). The track paths are all representative of different chords of the same circle Γ. The chords of a circle are uniquely identified by the two points where they cross the circle, which we label by their polar angle with respect to an arbitrary fixed zero point on the circle Γ, such that each track T_j is represented as $T_j \mapsto (\phi_j^A, \phi_j^B)$. Within this geometrical setting, a representation of the distance between two track paths (T_1 and T_2) is given by the distance between their two points in the two-angle space, so that

$$\text{dist}(T_1, T_2) = \|(\phi_1^A, \phi_1^B) - (\phi_2^A, \phi_2^B)\|. \tag{10}$$

The two-angle space is a natural choice for this problem since the distance in two similar dimensions is readily calculated in a meaningful manner.

The ordering of angles ϕ^A and ϕ^B is arbitrary since we do not specify a track direction, thus the mapping from angle space to track space of $T_j \mapsto (\phi_j^A, \phi_j^B)$ is a one-to-two mapping ($T_j \mapsto (\phi_j^B, \phi_j^A)$ for the same T_j for all j). To add logic to remove this ambiguity, all points are mapped to $\phi^A > \phi^B$. This distance measurement is beneficial since two track paths that cross at a ninety-degree angle have a large distance measurement, and also two track paths which are parallel but separated by a large distance have a large distance measurement. However, track paths that visually appear "close" to each other maintain a small distance measurement.

It is not within the scope of this work to propose new clustering algorithms, or to compare the many such algorithms that are available. Nonetheless, it is necessary to choose a clustering method to show that the four invariant lines is an excellent screening algorithm for proximity sensors. A few good nonparametric clustering algorithms for detecting clusters of unknown size in a noisy background have been recently developed. Most promising among these for applications with large data sets are algorithms by Byers and Raftery [11], Cueves et al. [12], and Ester et al. [13]. Wong and Moore [14] have examined the various computational issues that arise in approaches to this type of clustering problem. Based on the computational needs and mathematical formalism, we chose to use the algorithm DBSCAN [13], [15]. DBSCAN was designed for solving automated clustering problems in very large data warehouse problems with no a priori knowledge of the cluster sizes or content. Such problems are similar to this problem of finding clusters of unknown size in a large set of points in the (ϕ^A, ϕ^B) plane.

The algorithm DBSCAN works on the intuitive notion of a cluster as a region of the space with a high density of points. In this case the space is the two-dimensional (ϕ^A, ϕ^B) space and the points are the locations corresponding to potential track paths. DBSCAN begins by searching the set of all points

to find a set of core points that are defined as those points that have a minimal number M of other points within an ε-region of it. That is, for a set of points P, we define all points within an ε-region of a point p_j as the set $P_\varepsilon(p_j)$ where

$$P_\varepsilon(p_j) = \{p \in P \mid \text{dist}(p, p_j) \leq \varepsilon\} \tag{11}$$

where dist$(\cdot)$ represents an appropriate distance measure. In our case the distance is the Euclidean distance in the (ϕ^A, ϕ^B) space as given by (10). A core point is now defined as any point where the size of $P_\varepsilon(p_j)$ is greater than or equal to M.

Given a set of all core points, the algorithm then searches to find connected sets of core points which are formed from grouping core points that have any similar points in their ε-regions. These connected core points are the building blocks of clusters. From each connected set of core points we label all of the points within the ε-regions as interior points $p_j(C_i)$ of the cluster C_i. The algorithm seeks any points $q \in P$ that have dist$(q, p_j(C_i)) \leq \varepsilon$ and adds that point to cluster C_i. If the same point is added to two clusters, the clusters are merged. In this manner a set of points $p \in P$ is found that are connected via a high-density path, which matches the intuitive notion of a density cluster. This approach requires no a priori knowledge of cluster size nor number of clusters; however, it does require parameters M (number of points within a neighborhood to call it a core point) and ε (the size of the neighborhood).

In the absence of any knowledge of the number of "valid" detections, a small number should be used for M, so that no clusters are missed. In this case $M = 3$ is a reasonable choice. The separation of the correct cluster from other track paths requires a proper choice of ε. To set this parameter, a Monte Carlo computer simulation with varying target tracks and sensor locations was run with the number of valid detections fixed at 4 and the number of false alarms varied from zero to 4. The values of sensor position were assumed correct, and the β_j values were varied with relative errors normally distributed with a standard deviation of 20%. These simulations show that $M = 3$ and $\varepsilon = 0.12$ rad provide the correct clustering (the correct track paths get clustered together). Smaller values of ε should be used for higher accuracy sensors (more accurate β_js) and larger values of ε should be used for lower accuracy sensors. The proper values of these parameters for a larger range of conditions is a subject of future research.

In Fig. 6(b) we show the resulting clustered track paths from the case in Fig. 6(a) that are found by automated clustering using DBSCAN. We recall that the true track path is horizontal and in the center of the figure. It is apparent that the algorithm presented does a favorable job of finding the neighborhood of the true track path from the limited information of proximity alerts. Figs. 7(a) and 7(b) show the initial and resulting clustered track paths from a case with multiple false alerts (with four real detections and three false alerts) where it is clear that the false alerts do not impact the resulting cluster, but do add some spurious clusters.

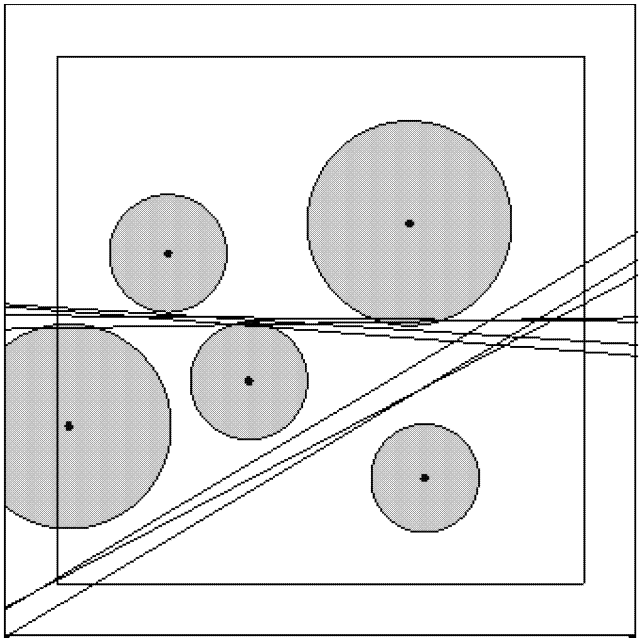

Fig. 6(b). Clustered track paths found in set of potential track paths of Fig. 6(a).

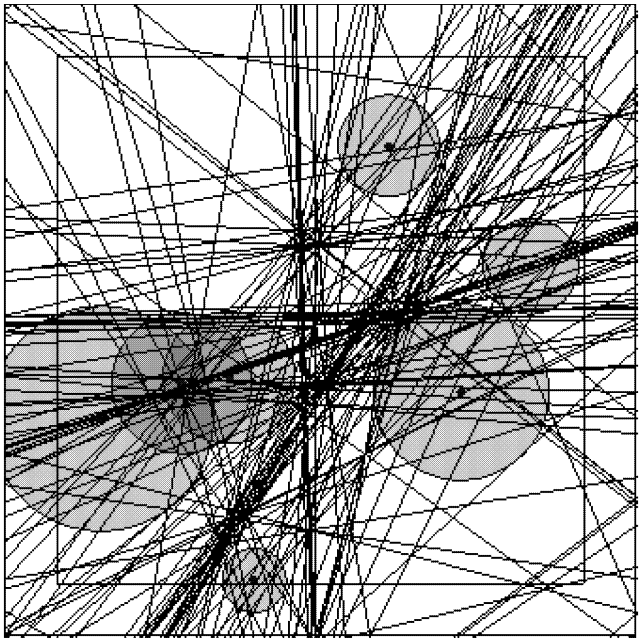

Fig. 7(a). Detection circles and potential track paths for set of 4 true detections plus 3 false alerts. False alerts are highest and lowest circles on plot, as well as left-most circle.

From the resulting multiple clusters, the most likely cluster must be chosen. The cluster that contains the most track paths is considered to be the most likely cluster, and it is chosen as the correct cluster (step 4 of algorithm). Since the clusters are naturally

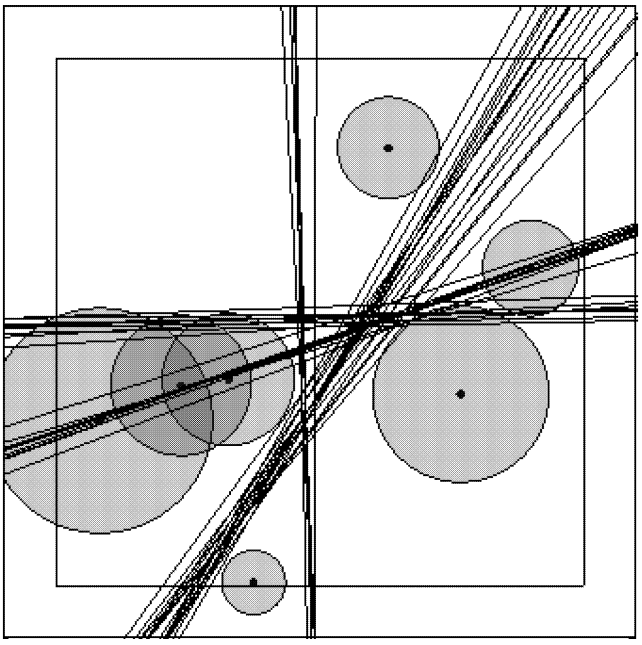

Fig. 7(b). Clustered potential track paths from case shown in Fig. 7(a).

a tight grouping of lines (by choosing a small value of ε), a reasonable target track path is any part of this resulting cluster. As a smoothing procedure, we choose to average the track paths within this cluster to obtain a single track path. This averaging is performed in the (ϕ^A, ϕ^B) angle space and the result is mapped back into track space. This process yields a track path that passes through the center of the most populous cluster.

IV. CONCLUSION

A method for determining a target track from simple proximity information for a distributed sensor system is developed. The new method addresses the problem where very limited data from each sensor is made available to the fusion center. The method exploits a geometric invariant of proximity between sensors and track paths for sets of three sensors. For each set of three sensors, four potential track paths are computed. The algorithm searches for clusters of track paths from the set, looking for a cluster that contains those track paths that are consistent between the ensemble of all sets of three sensors. The algorithm is developed for no communications between sensors, and very small bandwidth communications between the sensors and the fusion center. Furthermore, on simulated data, the algorithm performs well in the presence of false alarms, since each false alarm only impacts a small number of the ensemble of all potential track paths, and leaves the large cluster around the correct track path unchanged.

The method currently is limited to single targets with constant speed and heading, operating in an environment with known constant attenuation. Extensions to the method to alleviate these constraints are planned areas of future research. Such extensions include the integration of this algorithm as a component of larger algorithm, whereby small segments of track are generated and then combined into a more complex track using conventional statistical methods.

THOMAS A. WETTERGREN
ROY L. STREIT
JOHN R. SHORT
Naval Undersea Warfare Center
1176 Howell St.
Newport, RI 02841
E-mail: (t.a.wettergren@ieee.org)

REFERENCES

[1] Thomopoulos, S. C. A., Viswanathan, R., and Bougoulias, D. K. (1989)
Optimal distributed decision fusion.
IEEE Transactions on Aerospace and Electronic Systems, **25**, 5 (Sept. 1989), 761–765.

[2] Viswanathan, R., and Varshney, P. K. (1997)
Distributed detection with multiple sensors: Part I—Fundamentals.
Proceedings of the IEEE, **85**, 1 (Jan. 1997), 54–63.

[3] Blum, R., Kassam, S. A., and Poor, H. V. (1997)
Distributed detection with multiple sensors: Part II—Advanced topics.
Proceedings of the IEEE, **85**, 1 (Jan. 1997), 64–79.

[4] Xiang, M., and Zhao, J. (2001)
On the performance of distributed Neyman-Pearson detection systems.
IEEE Transactions on Systems, Man, and Cybernetics, Part A, **31**, 1 (Jan. 2001), 78–83.

[5] Yan, Q., and Blum, R. S. (2001)
Distributed signal detection under the Neyman-Pearson criterion.
IEEE Transactions on Information Theory, **47**, 4 (May 2001), 1368–1377.

[6] Yan, Q., and Blum, R. S. (2000)
On some unresolved issues in finding optimum distributed detection schemes.
IEEE Transactions on Signal Processing, **48**, 12 (Dec. 2000), 3280–3288.

[7] Ansari, N., Chen, J-G., and Zhang, Y-Z. (1997)
Adaptive decision fusion for unequiprobable sources.
IEE Proceedings of Radar, Sonar, and Navigation, **144**, 3 (June 1997), 105–111.

[8] Rao, N. S. V. (1997)
Distributed decision fusion using empirical estimation.
IEEE Transactions on Aerospace and Electronic Systems, **33**, 4 (Oct. 1997), 1106–1114.

[9] Liu, J., Chu, M., Liu, J., Reich, J., and Zhao, F. (2004)
Distributed state representation for tracking problems in sensor networks.
In *Proceedings of IPSN04*, Apr. 2004, 234–242.

[10] Dorrie, H. (1965)
100 Great Problems of Elementary Mathematics: Their History and Solution.
New York: Dover, 1965, ch. 32, 155.

[11] Byers, S., and Raftery, A. E. (1998)
Nearest-neighbour clutter removal for estimating features in spatial point processes.
Journal of the American Statistical Association, **93** (1998), 577–584.

[12] Cueves, A., Febrero, M., and Fraiman, R. (2000)
Estimating the number of clusters.
The Canadian Journal of Statistics, **28**, 2 (2000), 367–382.

[13] Ester, M., Kriegel, H-P., Sander, J., and Xu, X. (1996)
A density-based algorithm for discovering clusters in large spatial databases with noise.
In E. Simoudis, J. Han, and U. Fayyad (Eds.), *Proceedings of Second International Conference on Knowledge Discovery and Data Mining*, AAAI Press, 1996, 226–231.

[14] Wong, W-K., and Moore, A. (2002)
Efficient algorithms for non-parametric clustering with clutter.
In *Proceedings of Interface 2002*, 2002.

[15] Ester, M., Kriegel, H-P., Sander, J., Wimmer, M., and Xu, X. (1998)
Incremental clustering for mining in a data warehousing environment.
In *Proceedings of the 24th VLDB Conference*, 1998.

Comments on "Unbiased Converted Measurements for Tracking"

In this note, we show that there exists a compatibility problem in the derivation of the mean and covariance of the converted measurement errors in [1], and then present a modification to the computation of them, in which both the mean and the covariance are computed strictly conditioned on the measurements.

In [1], the authors showed that the exact compensation for the bias in the classical sensor-to-Cartesian conversion is multiplicative and depends on the cosine of the angle measurement errors, and then presented an unbiased converted measurements Kalman filtering (UCMKF) algorithm by using the multiplicative bias compensation factors.

While appreciating the work in [1], there exists a compatibility problem in the derivation of the mean and covariance of the converted measurement errors, as pointed out in [2]. In [1], the mean of the converted measurement errors was computed conditioned on the true range, bearing (azimuth) and elevation angles of the target, but for the corresponding covariance, owing to the unavailability of the true values, the authors proposed to compute it conditioned on the measurements directly. This can be shown in detail as follows.

Manuscript received January 27, 2004; revised May 5, 2004; released for publication July 27, 2004.

IEEE Log No. T-AES/40/4/839861.

Refereeing of this contribution was handled by P. K. Willett.

I. MODIFIED UNBIASED CONVERTED MEASUREMENTS

A. 2D Case

Because the compatibility problem is the same for all the elements of the covariance R_p of the converted measurement errors, we can just analyze (7a) in [1] to illustrate this.

From probability theory, (7a) in [1] should be

$$\begin{aligned} R_p^{11} &= \mathrm{var}(x_m^u \mid r_m, \beta_m) \\ &= E\{[\lambda_\beta^{-1} r_m \cos\beta_m - r\cos\beta]^2 \mid r_m, \beta_m\} \\ &\quad - E^2[\lambda_\beta^{-1} r_m \cos\beta_m - r\cos\beta \mid r_m, \beta_m]. \end{aligned}$$

The second term above is set to zero due to (5) in [1] directly. But for (5) in [1], the unbiasedness of the measurement conversions can hold only under the assumption that the true range and bearing of the target are known. That is to say, for (5) in [1]

$$E[x_m^u \mid r, \beta] = r\cos\beta, \qquad E[y_m^u \mid r, \beta] = r\sin\beta$$

are the requirements.

Thus it can be seen that, in deriving (7a) in [1], two different conditions are used: the first half is computed conditioned on the measurements directly, while the second half is computed conditioned first on the true range and bearing of the target and then on the measurements.

It thus can be clearly seen that there exists a compatibility problem in the derivation of the mean and covariance of the converted measurement errors in [1]. To overcome this, the first two moment estimates conditioned on the measurements should be rederived as follows.

Define $x_m^u = x + \tilde{x}_m^u$, $y_m^u = y + \tilde{y}_m^u$. The measurement conversion equations are still the same as (5) in [1]. Then the measurement-conditioned mean of the converted measurement errors are given by

$$\mu_p^1 = E[\tilde{x}_m^u \mid r_m, \beta_m] = (\lambda_\beta^{-1} - \lambda_\beta) r_m \cos\beta_m$$

$$\mu_p^2 = E[\tilde{y}_m^u \mid r_m, \beta_m] = (\lambda_\beta^{-1} - \lambda_\beta) r_m \sin\beta_m$$

and the corresponding covariance are given by

$$\begin{aligned} R_p^{11} &= \mathrm{var}[\tilde{x}_m^u \mid r_m, \beta_m] \\ &= -\lambda_\beta^2 r_m^2 \cos^2\beta_m + \tfrac{1}{2}(r_m^2 + \sigma_r^2)(1 + \lambda'_\beta \cos 2\beta_m) \\ R_p^{22} &= \mathrm{var}[\tilde{y}_m^u \mid r_m, \beta_m] \\ &= -\lambda_\beta^2 r_m^2 \sin^2\beta_m + \tfrac{1}{2}(r_m^2 + \sigma_r^2)(1 - \lambda'_\beta \cos 2\beta_m) \\ R_p^{12} &= \mathrm{cov}[\tilde{x}_m^u, \tilde{y}_m^u \mid r_m, \beta_m] \\ &= -\lambda_\beta^2 r_m^2 \sin\beta_m \cos\beta_m + \tfrac{1}{2}(r_m^2 + \sigma_r^2)\lambda'_\beta \sin 2\beta_m. \end{aligned}$$