A sensor array of this type may be located at a single site, in which case we consider it as one unit, or it may be distributed among many sites.

In other cases the sensor sites can provide different types of target related data, in particular:

(i) Range only (ranging radar, active sonar)
(ii) Bearing only (optical, infrared sensor, direction finder)
(iii) Bearing and range (search, tracking radar)
(iv) Target velocity (Doppler radars, MTI)

Different data types lead to different location estimation techniques. For example, range only or bearing only measurements are related to target association techniques (section 2.2). Bearing and range data is usually associated with tracking algorithms for moving targets.

Data from a single omnidirectional passive sensor is treated by time-of-arrival methods (section 2.1) or beamforming and array processing techniques (section 2.4). The estimation method also depends on the type of signals provided by the sensor sites: coherent/noncoherent, "raw" or filtered data (linear processing), data after detection (nonlinear processing), etc.

Classical methods of processing sensor data have generally been of the centralized type, that is, all of the sensor data was collected at one site and then processed. An alternative is to process much of the data at the collection site and to send only the relevant data to either a central site or (more generally) to the appropriate user. In Section 3 we shall discuss the different types of distributed processing and their advantages.

In the last section we will describe our own approach to the development of distributed algorithms for the estimation of position, location and other characteristics of sensors and sources. (Dorf et al.). We will give a short description of some algorithms of a fully distributed nature that have desirable features. We shall also outline several of the nonclassical approaches to solving these problems.

2.0 Techniques for estimating target location

This section provides a brief summary of the solution techniques associated with the target

---

This work was supported by ARPA under contract N00014-73-C-0179.
location problem. Only the basic ideas are presented; the details can be found in the references.

2.1 Time-of-Arrival Estimation

A signal emanating from a remote source and measured in the presence of noise at two spatially separated sensors can be modeled as

\[ x_1(t) = s(t) + n_1(t) \]  
(1a)

\[ x_2(t) = s(t+\tau) + n_2(t), \]  
(1b)

where \( s(t) \), \( n_1(t) \), \( n_2(t) \) are assumed to be stationary, independent, random processes. One common method of estimating the time delay, \( \tau \), is to compute the cross-correlation function

\[ R_{x_1x_2}(\tau) = \int x_1(t)x_2(t+\tau) \, dt \]  
(2)

It follows directly that

\[ R_{x_1x_2}(\tau) = R_{sT}(\tau) \]  
(3)

where \( R_{sT}(\tau) \) is the signal autocorrelation function. An important property of autocorrelation functions is that \( R_{sT}(\tau) \) will occur at \( \tau = 0 \). This provides us with a way for finding the delay by calculating the estimated cross-correlation function. If \( s(t) \) is a white noise source, \( R_{sT}(\tau) = \delta(\tau) \), and the peak will be sharply defined. In general, \( R_{sT}(\tau) \) will be "spread out" which tends to broaden the peak, making it more difficult to pinpoint the actual delay. Furthermore, when multiple targets (and multiple delays) are present, the "tails" of the autocorrelation functions for different targets will be overlapped and more difficult to separate. Thus, it is desirable to pre-process the sensor measurements \( x_1, x_2 \) so that after cross-correlation sharper peaks will result, as in Fig. 1. In the absence of measurement noise this can be done by passing \( x_1(t) \), \( x_2(t) \) through a "whitening" filter for \( s(t) \), canceling the correlation of \( s \) is removed. When noise is present, the filter has to take into account both signal and noise spectra.

![Fig. 1: A time-delay estimator.](image)

Different choices for the pre-filter are possible depending upon the performance criterion chosen by the designer: the likelihood function (Hann and Tretter), the deflection function (Knapp and Carter), etc. It should be noted that several estimator structures beside the multiplier-correlator estimator have been developed.

All of the information about the target location are encoded in the relative time-delays of the various sensors. To see this consider the following:

\[ Y = (x_1, y_1) \]
\[ T = \text{Target} \]
\[ S_1 = \text{Sensor 1} \]

![Fig. 2: Target-sensor geometry](image)

A special type of problem arises when multiple sensors which measure range but not azimuth (or vice versa) are used to estimate target location. If only a single target is present, its location is found by trilateration. For example, if azimuth measurements from several sensors are available, one has only to compute the intersection of the various lines-of-sight.

2.2 Target Association Techniques

There are several target association techniques. One common approach is to use the maximum likelihood approach [Bangs and Schnitzeiss, [MacDonald].]
The situation becomes more complex when multiple targets are present. This is illustrated by Fig. 3.

Each sensor is assumed to have detected the two targets T1, T2. However, these detections are not properly associated. It is not known which measurement of each sensor corresponds to which target. Thus, it is necessary to associate targets with sensor measurements before estimating the target locations (in fact the association and location problems are addressed simultaneously). Note also that if there are more targets than sensors, ambiguities ("ghost" targets) may result.

Several schemes have been proposed to solve the target association problem, and they are briefly described below.

**List Forming**

Pick a pair of sensors and compute all the intersections of their lines-of-sight to potential targets (i.e. directions in which they detected something). These intersection points are potential target locations. Now pick a third sensor and check whether its line-of-sight passes through any of the intersection points. If not, delete these points from the list of potential targets. By proceeding this way with the other sensors, the list will finally include only those target locations which are consistent with all the observations. It should be emphasized that this is a highly simplified description of more realistic list forming algorithms.

**Back-projection or Space-Search**

The space to be searched is divided into cells of a size corresponding to the system resolution. The number 1 is added to those cells of the space which lie along the line of sight of a given sensor detection. This process is repeated for all lines-of-sight of all sensors. As can be seen from Fig. 4 the target locations can be identified as those having the highest number (a the number of sensors) written in them. Note that relatively high values can be obtained at locations other than those of the real target ("ghost" locations).

![Fig. 3: The target association problem](image)

![Fig. 4: Association by back-projection](image)

**Image reconstruction techniques**

It is possible to view the sensor measurements as line integrals through an image consisting of "points of light" at the target locations. The line integrals are over the infrared emissivity map (for IR sensors) or the radar reflectivity map (for ranging radars). The problem of reconstructing images from their line-integral projections has been extensively treated in literature (e.g., Brooks and Di Chiro, Horn). Recently it was shown how these techniques can be applied to the target association problem (Friedlander et al., Denton et al.) by reconstructing the "brightness map" of the area under surveillance and identifying targets as the "bright spots". It should be noted that the image reconstruction method requires that the sensors provide the actual energy measured in each direction (range) and not just target/no target information. The detection takes place after processing the information from all the sensors; in list forming and space-search, only the results of the detection performed at each individual sensor are passed on.

**2.3 Spectral estimation**

Multiple-sensor measurements can be considered as samples of a time-space function \( y(t, z) \) where \( P \) represents a point in \( 3-D \) space. The notions of (temporal) correlation function and (temporal) spectral density can be extended to time-space functions of this type. We define a random field \( y(t, z) \) as stationary and homogeneous if \( E[y(t, z)|z=0] = y(t, z) \) and

\[ E[y(t, z)|z=0] = y(t, z) \]
Thus, target bearing estimation reduces to the problem of estimating \( P(\omega, k) \) from the measurements \( y(t_i, k) \), where \( r_i \) represents sensor locations and \( t_i \) are the sampling times of the output of that sensor.

Many spectral estimation techniques have been used in this context; several are described in the references.

2.4 Beamforming and Array Processing

Perhaps the most common operation in processing signals in a sensor array is that of beamforming. Antenna arrays (for radar, communication, etc.) and acoustic sensor arrays ( SONAR) are typical examples of beamforming. Beamforming consists of a summation of time-delayed (or phase-shifted) versions of the sensor outputs. i.e.

\[
Z(t) = \sum_{i=1}^{N} y(t-t_i, k_i)
\]

where \( t_i \) represents the time-delay for sensor \( i \) and \( P_i \) its location. Proper choice of the delays \( t_i \) enhances the signals received from a particular direction and attenuates signals from other directions. This operation is the spatial equivalent of a temporal narrow bandpass filter.

The output \( Z(t) \) of the beamformer is a (SONAR) time function, which is processed so as to obtain a measure of the signal energy in an optimal way. The most common processing schemes include

- Matched filtering
- Wiener filtering or least-squares estimation
- Maximum likelihood estimation

This linear filtering is often followed by a nonlinear operation, e.g. squaring and integration. The order of linear filtering and transforming (also a linear operation) is often reversed, the exact structure depending on the application. The implementation of these processes is usually done in the Fourier domain (with phase shifts replacing time delays) but time domain implementations are also used. A sample of the vast literature on beamforming and the associated signal processing (referred to as "array processing") is given in the bibliography.

A class of array processors of particular interest are the different types of adaptive arrays. The need for adaptive arrays arises for many reasons. Some examples:

- Null steering, to minimize interference from sources other than the target of interest.
- Adaptive filtering, to handle unknown noise and signal statistics.
- Adaptive beamforming. Beamforming requires precise knowledge of the sensor locations (within fractions of a wavelength) in order
that the steering delays be computed. Relatively small errors can lead to serious performance degradation. Thus, when sensor locations are imprecisely known or are constantly changing, a fixed processing scheme is infeasible.

3. The Need for Distributed Computation

There are many advantages in distributing computations for a large sensor network; some of the main arguments are the following:

1) Reduction of Computational Complexity

Distributed processing is often used as a means for solving problems related to large-scale systems. This approach leads to the decomposition of a high-dimensional problem into a sequence of smaller-dimensional ones. This often results in considerable computational savings; many fast algorithms, such as the Fast Fourier Transform, are of this type. Also, certain large-scale problems simply cannot be solved in a direct manner (e.g., inversion of very large matrices) and ways have to be found to decompose the problem into smaller parts that can be handled. This can, of course, be done in a centralized manner, but the distributed approach often leads to natural decompositions and valuable insights.

2) Reliability

Distributed systems have good properties from a reliability standpoint due to their inherent parallelism. Failure of a computational module does not necessarily result in system failure since the computational load can be re-distributed among the remaining modules. Thus, a distributed system may have the ability to reconfigure and continue operation. Depending on the type of the system and its structure, its operation after reconfiguration may be at a reduced performance level. This would be the case if the remaining computational resources were insufficient to complete the solution of the problem, or if the loss of a computational module was associated with the loss of a sensor site. The system displays graceful degradation of performance, which contrasts with the "catastrophic" failure mode of centralized systems.

3) Flexibility

The distributed nature of computations is often associated with distributed system structures; such a system structure might be a collection of sensor/computer/communication modules interconnected in a network. This organization leads to a more flexible structure, possessing desirable properties which are not always present in a centralized system:

- Easy system growth
- The capacity to handle topological changes in the system structure (e.g., adding or deleting nodes during maintenance without interrupting system operation).

4) Our Approaches

Distributed processing has by now become a term that is applied to many types of systems and is not very well defined. Since the distributed sensor net problem can be well described, see e.g. [12], our approach leads to the decomposition of optimal parallelism. Failures since the computational load can be re-distributed, and continue operation. Depending on the Centralized and Independent algorithms, and cooperatively distributed.

1) Centralized — all sensor data is passed to a central site, where computation is performed, and the pertinent results are then returned to the appropriate remote sites.
2) Independent — all sensor information is communicated to every other site, and each site then makes its best estimate of the environment.
3) Cooperative — sites exchange processed information, and at most partial sensor data.

Centralized algorithms are now quite well understood, as a perusal of the extensive literature indicates. However, very few attempts have been made to unify and integrate all these different approaches and results. A typical look on radar or sonar signal processing is a rather ad hoc collection of data, methods and theory. It is extremely difficult to get a coherent picture and to make intelligent inferences in applying these methods in the design of systems. A systematic representation of this knowledge, by itself a tremendous task, is required in order to make effective use of the available alternatives.

4) The Search for Distributed Algorithms

Our approach to the development of (cooperatively) distributed algorithms can be summarized under the following headings:

1. Partitioning of optimal centralized algorithms, such as maximum-likelihood, extended Kalman filtering, and beam forming. This approach is useful when the system under consideration can be...
divided into subsystems with sparse interaction.

2. Application and extension of methods developed in the context of decentralized control and estimation, e.g. team decision theory and differential games, hierarchical and multilevel systems, aggregation methods, singular perturbation and other perturbation techniques, periodic coordination and spatial dynamic programming (SDP).

3. Development of new optimal distributed algorithms for specific subsets of sensor data. For example,
- Time/Frequency Difference of Arrival (TDOA/FDOA) data, using ARMA modeling.
- Range only or Angle only data, using range reconstruction techniques or nonincremental versions of the backprojection technique described in Section 2.2.
- Range and Angle data, using Nonlinear Estimation techniques.
- Mixed, possibly inconsistent/incomplete data, using a hierarchical approach.

4. Advanced Concepts
- The physical problem of locating and identifying sources has much mathematical structure; for example, it is heavily dependent upon the choice of coordinate system. Non Euclidean Geometry and Non Classical Statistics (Non-Gaussian) are very probably of great benefit.
- Signal processing of one-dimensional signals is a very well developed field; spatial and other multi-dimensional problems, however, require more advanced mathematical tools.
- Our preliminary investigations indicate that most candidate algorithms for distributed processing require high communication bandwidths. As an alternative approach, we are investigating probabilistic algorithms; these algorithms potentially require lower bandwidths, are naturally suited to parallel and distributed organizations, and they too are very robust.
- From a systems-design perspective one should consider interactions between software and hardware architecture early in the development of algorithms. For this reason, we are considering the potential impact of VLSI/VHSI designs.

Using these approaches, examples of fully distributed processing and communication algorithms can be proposed. One such example is the combination of the TDOA approach (Sommer, the distributed estimation algorithm in [ASI-BSN report]) and a distributed protocol a la [Merlin and Segall]. These algorithms have the desired robustness and low communication bandwidths that characterize desirable distributed algorithms.

References

Time of Arrival Estimation


Tracking Moving Targets


Target Association


Image Reconstruction


Kolker, A.B., "Covariance Between Factor Coefficients Representing the Time waveforms Observed from an Array of Sensors," JASA, Vol. 57, no. 3, March '76.


**Spectral estimation**

**spatial**


**Temporal**


Marple, L., "EE Dissertation, Stanford University, '77.

**Reforming**

Oceanic Engineering vol.06-1, No. 1, September '76.


Steinberg, B.D., Principles of Aperture and Array System Design, John Wiley & Sons, '76.


Array Processing


Rockmore, A.J., Lecture notes, Stanford University, Summer 1977.


Adaptive Arrays


MISCELLANEOUS


