ESD-TR-78-398

LEVEL II

12

Semiannual Technical Summary

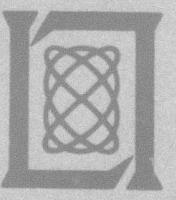
See 1473

Distributed Sensor Networks

30 September 1978

Prepared for the Defense Advanced Research Projects Agency
under Electronic Systems Division Contract F19628-78-C-0002 by

Lincoln Laboratory
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
LEXINGTON, MASSACHUSETTS



The work reported in this document was performed at Lincoln Laboratory, a center for research operated by Massachusetts Institute of Technology. This work was sponsored by the Defense Advanced Research Projects Agency under Air Force Contract F19628-78-C-0002 (ARPA Order 3345).

This technical report has been reviewed and is approved for publication.

FOR THE COMMANDER

Joseph C Syiek

Joseph C. Syiek
Project Officer
Lincoln Laboratory Project Office

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

LINCOLN LABORATORY

DISTRIBUTED SENSOR NETWORKS

SEMIANNUAL TECHNICAL SUMMARY REPORT
TO THE
DEFENSE ADVANCED RESEARCH PROJECTS AGENCY

1 APRIL – 30 SEPTEMBER 1978

ISSUED 2 MAY 1979

LEXINGTON MASSACHUSETTS

ABSTRACT

This report summarizes Distributed Sensor Networks research conducted during the period 1 April through 30 September 1978. One specific model for function decomposition and distribution is presented. A top level design of a strawman DSN for detecting and tracking low flying aircraft is described. High resolution acoustic signal processing algorithms for the strawman are specified and sized. Additional features of a multisite target search algorithm are presented. A moderately detailed model for simulation of an acoustic node under various scenarios is developed. New model simulation and experimental data analysis software are described and demonstrated. Technical issues involved with and plans for a digital acoustic data acquisition system are presented.

CONTENTS

SUMMARY

This third Semiannual Technical Summary (SATS) for the Distributed Sensor Networks (DSN) program reports research results for the period 1 April through 30 September 1978. Topics covered include decomposition and distribution models, strawman DSN description, single- and multiple-site processing algorithms, acoustic-node models, model software and experimental-data software, and experimental data acquisition facilities.

Our present ideas for the decomposition and distribution of detection and tracking tasks in a DSN are oriented geometrically. Very briefly, sensors are grouped into sets and the data from a set is used to detect and track in a specified region. Regions may overlap and, in general, both the sensor sets and the regions are functions of time to account for variations of system load or available resources. Chapter I, presenting a more formal statement of these ideas, constitutes one general approach to DSN decomposition and distribution of which the strawman system presented in Chapter II is an example.

In Chapter II we present a description of a strawman DSN that uses microphones and modest radars to detect and track low-flying aircraft and cruise missiles. This system is not being proposed as a system design to be developed and deployed to solve a specific problem. Rather, it is the stepping-off point for further analysis, simulations, and redesign. It is a specified system context within which all system issues can be explored and DSN technology might be developed and demonstrated. Subject to future design efforts, the system presented will have the following characteristics:

1. Multiple geographically distributed cooperative sensors of limited individual capability to detect and track low-flying aircraft.
2. Distributed intelligence for data screening, and in fact, to accomplish all major system functions.
3. Adaptability to resource and load variations.
4. Packet radio for communication.
5. Large amounts of computer power to apply as necessary.

Topics addressed in this general design document include: system configuration, operational modes, multiple-node processing, reporting of surveillance information to users, communications, sensors and sensor data processing, and digital hardware. Communication is given substantial attention because it appears to be a major DSN consideration.

Chapter III is a more detailed presentation and sizing of the high-resolution acoustic detection and azimuth-estimation algorithms that are incorporated into the design cited in Chapter II. The algorithm is computationally intensive and represents a worst-case load of some 11-million floating-point operations per second for the acoustic array at a DSN node.

Chapter IV deals with multisite detection and location algorithms. The results of the last part of our exploratory work on simple, decision-theoretic, surveillance-space-search algorithms are reported. This was done in the context of using multiple-acoustic azimuth measurements. A change from using approximate algorithms with three-dimensional cells to using exact algorithms with two-dimensional cells was made and the expected improvement obtained. Also, the beneficial results obtained by using smaller cells in conjunction with interpolation between actual measurements was demonstrated.

A model for an acoustic node in a DSN has been formulated and corresponding simulation software has been written. Chapter V contains the model, information about related software, and some results obtained with the model and software. The software package can be used for exploratory analysis of experimental data as well as simulations. This is partly true because the simulation approach was to use actual wavenumber software to process the power-spectral-density matrices that can be generated by the model without using real data. This matrix is calculated theoretically from the range and source levels of targets and from additional information about noise sources. The first-order validity of the model is demonstrated and the value of multiple-dyad estimates of power-spectral-density matrices for the multiple-target case is also confirmed.

Chapter VI discusses digital acoustic data acquisition issues and alternatives for data acquisition systems in support of future research.

CONTRIBUTORS
TO
DISTRIBUTED SENSOR NETWORKS PROGRAM

GROUP 22

Lacoss, R. T.
Demko, P.
Landers, T. E.
Walton, R. L.

GROUP 27

Duckworth, G. L.

DISTRIBUTED SENSOR NETWORKS

I. DECOMPOSITION AND DISTRIBUTION OF DSN FUNCTIONS

The functions of target detection, location, tracking, and identification are critical Distributed Sensor Networks (DSN) functions that must somehow be decomposed and distributed over the nodes and computer hardware of the DSN. We present one approach to the problem in this chapter. (The strawman DSN design presented in Chapter II adheres essentially to the same viewpoint, although differing in detail.)

Our geographically oriented approach to decomposition and distribution is summarized in the following three sections. We:

1. Take a very high level minimally decomposed view of the overall surveillance problem making use of many sensors.
2. Present a geographically motivated decomposition of the problem.
3. Discuss mappings of the decomposed problem onto the processors in a DSN.

A. OVERVIEW OF A MULTIPLE-SENSOR SURVEILLANCE SYSTEM

Consider a surveillance system that contains a number of sensor sites. Each sensor site may contain any number of sensor types. The data from these sensors are processed and combined to give the best possible overall picture of activity in the area under surveillance. Finally, a number of users with distinct interests wish to view the available data in independent ways.

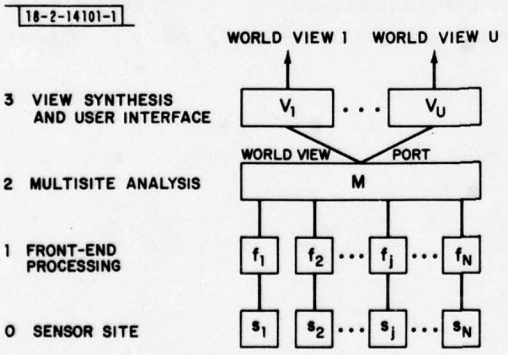


Fig. I-1. Structure of an undecomposed multiple sensor and multiuser surveillance system.

Figure I-1 shows schematically a proposed multilevel structure for such a surveillance system. Raw data from a sensor site, s_i, are operated upon by process, f_i. In most cases, f_i will reduce data quantity and generate more significant signal parameter estimates or target features. The f_i produces as much useful information as possible about targets in the area without recourse to the information that can be extracted from other sensor sites. Just what that useful information might be depends on the overall surveillance algorithms. For acoustic-array sensors the output data might be measurements of acoustic azimuths to targets and a

summary of the spectral features of the target. For a simple radar it might include range and azimuth, but with possibly very poor azimuth information. The f_i might also deliver raw data on command, but that would be the exception rather than the normal operating condition.

The outputs of the f_i are processed and combined by a second level of analysis, M, to produce a comprehensive picture of activity in the area. It may not be the best picture for all purposes, but it should be the best overall picture we know how to produce routinely.

User interfaces and specialized views of the world at level three can be tailored to the needs of individual users or tasks. They do sifting and specialized organization. For the most part, they will reorganize and interpret summary data produced by M, but, through M, could also make use of f_i outputs including raw data.

As described, the surveillance function, exclusive of the different user views, is accomplished by processes, $f_1 \ldots f_n$ and M. Although this represents some decomposition of the problem, we consider this to be an essentially undecomposed version of the surveillance tasks because M must handle all of the f_i in the system.

B. DECOMPOSITION OF SURVEILLANCE FUNCTIONS

Our approach to the distribution of the surveillance function in a DSN is to first find a natural way to decompose the function into subfunctions. The subfunctions can then be implemented by individual processes and mapped onto the processors in the system. More than one of the processes may map onto a single processor, but the processes themselves will not be distributed over more than one processor. We describe such a decomposition of the surveillance functions in this section.

There is a natural decomposition for the DSN problem. Sensors have limited range and can at any instant in time contribute to surveillance of the region in which they are located. This leads to a natural problem decomposition that is indicated schematically in Fig. I-2.

The surveillance region and the set of available sensors are decomposed into subsets that cover the surveillance region and the complete set of sensors. These sets are $\Omega_1 \ldots \Omega_D$ and $S_1 \ldots S_D$. The entire surveillance space is the union of the Ω_i $i = 1, \ldots D$. The union of the

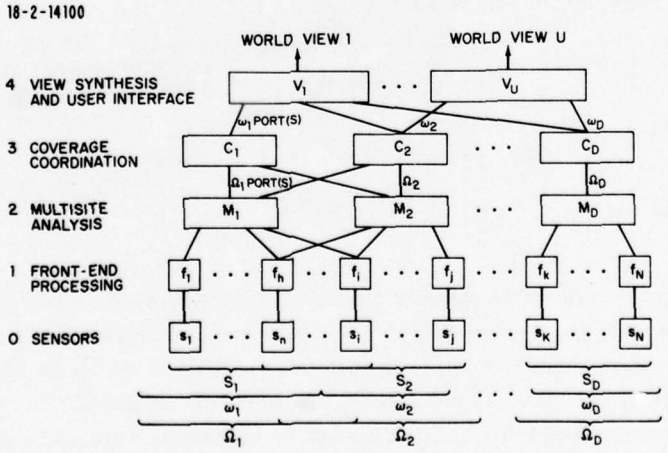


Fig. I-2. Structure of a decomposed multiple-sensor and -user surveillance system.

S_i i = 1,... D is the entire set of available sensors. The intersection of S_i and S_j is, in general, not empty nor is the intersection of Ω_i and Ω_j. S_i is a set of geographically neighboring sensors. Ω_i is the largest area that could be monitored meaningfully using only the sensors in S_i. In most cases, all the sensors in S_i will be located geographically in Ω_i, although that is not strictly required.

Given S_i, there is a multisite process, M_i, that would use those sensors without external information to perform surveillance of Ω_i. Such processes are shown in Fig. I-2 and are exactly like M shown in Fig. I-1. Thus the basic DSN functions can be broken down into many very similar functions with a smaller number of sensors and areas of surveillance. However, the overall surveillance function now requires some coordination of these otherwise independent elements. That coordination is the function of the C_i in Fig. I-2.

Each C_i not only deals with a number of M_j to coordinate surveillance information, but also becomes the prime system source of information about a region ω_i contained in Ω_i. The ω_i are all disjoint and completely cover the overall surveillance area. As shown in the figure, the structure is a strictly hierarchical tree structure. However, the C_i may actually have either explicit or de facto connections with other C_j. One result of this would be that under certain failure conditions, users could continue to get essentially undegraded surveillance information by using information from other than the primary C_i.

A single level of users is shown in the figure. In general and in practice, there will be a hierarchical structure of user functions built on the basic DSN. Although this is important, we defer discussion to the following section where we consider distribution. We do note, however, that the figure should not be interpreted to mean that users do not have access to information at any depth in the DSN. The system will allow the user to get data and information from any process in the system on command. However, the prime user-mode will not and cannot be to transmit large amounts of minimally processed data from many sites to a single site for processing and interpretation. This must be done in a distributed manner within the basic DSN.

C. DISTRIBUTION OF DECOMPOSED FUNCTIONS IN THE DSN

By definition each of the functions, f_i i = 1,... D, performs front-end data processing that is specific to a single-sensor site. The amount of processing at a single site can be substantial and easily justify the use of a dedicated computer. Also, the nature of front-end processing is such that it is an ongoing real-time process with only minor load variations as a function of target activity. For example, an acoustic site must continue to perform frequency-wavenumber array processing even when there are no targets present. This means that there would be no obvious advantage to implementing the individual f_i by processes distributed over more than one DSN node or to implementing several of them, serving several sensors on a single processor at a single node. Thus, each DSN sensor node will be colocated with a processor of sufficient size to support the associated front-end processing required by the f_i function. Any other alternative appears potentially to increase communication requirements with no compensating advantage.

Somewhat more arbitrarily, we constrain C_i and M_i, not only to present further division for distribution, but that they both reside on the same processor. The sense in which they are decomposed no further is that the major multisite detection, tracking, and target identification code experiences no further decomposition.

One system issue is whether DSN nodes should be homogeneous with respect to processing capability or not. For a homogeneous system, any node could support multisite-processing functions, M, C. However, if the number of multisite regions, D, into which the surveillance space is split is small relative to the total number of nodes, N, this might lead to considerable unused processor capacity at many nodes. In this case, there would be an argument for smaller processors and distributed realization of the individual M_i. Our proposed solution is to make N = D and require the processor at DSN node i to support M_i and C_i as well as f_i. This is a natural decomposition into small enough subfunctions to match a uniform set of node processors. The DSN trial design (Chapter II) adheres essentially to this proposal.

For the decomposition and distribution approach outlined here, adaptation to changes in available resources and variations in system load reduces primarily to adjusting the sets S_i, ω_i, and Ω_i for i = 1,... D. Of course, D will vary as the number of DSN nodes changes. Initial DSN designs to be considered will do the adaption using some simple algorithms that adapt to system configuration, but not necessarily to target load variations. For example, suppose f_i, M_i, and C_i run on a processor, P_i, located at node i of the DSN. We can assign any point in surveillance space which is geographically closer to P_i than to any other processor to be in the region ω_i associated with the functions running on P_i. The set, S_i, can be determined by either a geographic or communication distance from P_i; i.e., sensors less than some maximum distance away are included in the set.

Finally, we note that user interfaces will generally be hierarchically organized with some parts supported by the same P_i as the rest of the DSN, and some parts supported by independent processors. In the context of the trial design, the sector reporting function, which is included as part of the basic design, is essentially one particular, but quite general, user interface. In that case, it is supported completely by DSN processors. It has a distributed implementation in that surveillance data from some few tens of C_i are collected at a single site and there is an implied possible filter of the C_i outputs. That is, when the communication capacity cannot support all of the traffic information in the sector, then reports from the C_i are made selectively. That selection must be considered as constituting part of the view being implemented by that particular user interface. The general decomposition and distribution of user interfaces is beyond the scope of this section, which focuses on the most elementary DSN functions.

II. DSN STRAWMAN DESIGN

A preliminary high-level design for a DSN to detect and track low-flying aircraft and cruise missiles is being prepared. The design incorporates small radar and acoustic-array sensors and makes use of packet radio as the basic communication mechanism. The design is for a system that might be developed and demonstrated by the mid-1980s. However, there is at this time no plan to actually develop the system. The idea is that an attempted design of a realistic deployable system is one of the best ways to identify basic DSN problems and to evolve solutions. We expect that in the future, the design process will be iterated several times with major system changes resulting from lessons learned during earlier design phases. Repeated designs and system evaluation without commitment to or actual development of a specific system should be a cost-effective way to develop DSN technology and to eventually develop a specific DSN that is far more optimal than could be achieved by immediate commitment to a specific system development program. The design is a true strawman. It is expected to evolve into a design for a system that could be developed, but it is expected that the required changes may be very major indeed.

The design summarized in this Chapter constitutes our current version-1A DSN design. In the coming year the version-1A design will be the stepping-off point for system analysis and software test bed simulations that will be used to further develop the design and demonstrate how the system represented by the design might perform. The simulated systems will be denoted 1B, 1C, etc. The plan is that they should all be outgrowths of the 1A design, but may differ substantially from it.

Future plans also call for the development and demonstration of an experimental DSN consisting of three or more nodes with real sensors. Such an experimental system, at least partially because of the limited number of nodes, will be somewhat different from the large-scale systems represented by the version-1 systems. The design discussed here is not for an experimental system, but for a hypothetical full-scale system.

A. VERSION-1A DESIGN FEATURES

The following are major features to be exhibited by the version-1A design or follow-on designs to be demonstrated in a software test bed:

(1) The System is to Detect and Track Low-Flying Aircraft

The primary task of the DSN is to detect, locate, track, and identify low-flying aircraft. It is to make such surveillance data available selectively to users who may be within or outside of the DSN. Subject to physical constraints dictated by sensor capabilities, the surveillance data are to be made available in real time with only small delays for processing and communication.

(2) Use of Multiple-Cooperative Sensors of Limited Individual Capability

The system will make use of sensors that cannot furnish the required surveillance information individually. At the present time, attention is focused on short-range radars with very limited azimuth-measurement capability and small acoustic arrays that can measure the azimuth of

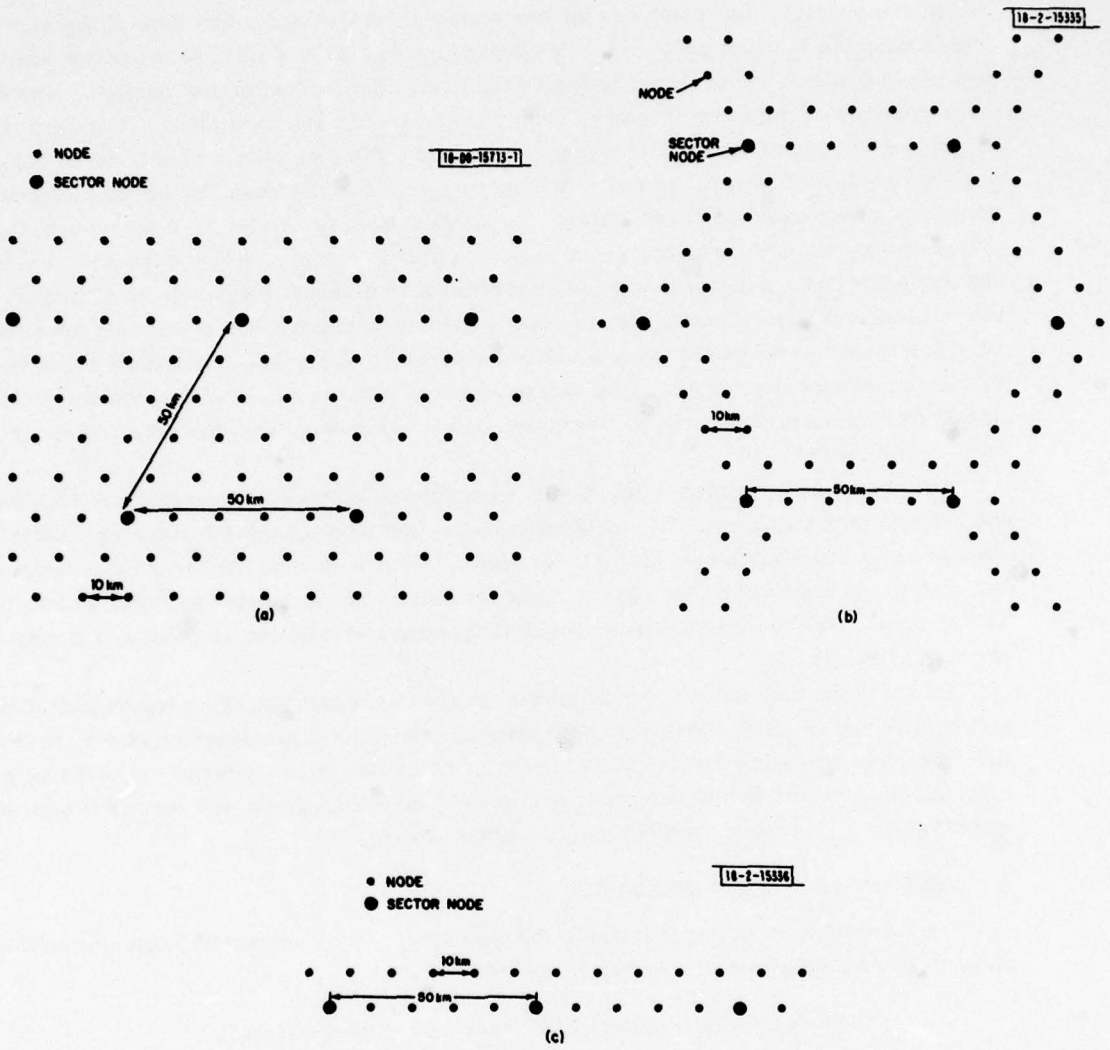


Fig. II-1. System configuration options: (a) area coverage, (b) honeycomb coverage, (c) linear barrier.

sound sources. Data from multiple sites will generally be required to locate and track a target and, as a target moves through the DSN, the set of sensors reporting new measurements will change continuously.

(3) Distributed Intelligence is Used for Data Screening

The tremendous amount of raw data generated by large numbers of sensors requires that computer data screening and analysis replace manual inspection, and that this be done near the sensor because limited communications bandwidth suggests that only near-final results be communicated.

(4) Adaption to DSN Configuration and Load Variation

One potential advantage of a system that makes use of very large numbers of distributed sensors, and which also distributes the surveillance function is the ability to adapt to and continue effective operation in the face of failure of various nodes in the system. The node failures may be natural, due to damage resulting from hostile action or communication failures resulting from jamming. Adaption to changing situations is a generic feature of DSNs and must be included in version-1 designs.

(5) Packet Radio is Used for Digital Communication

Packet radio is a suitable communication system meeting the requirements of rapid deployability, moderately high rate of secure digital data transfer, and unlimited system size. It is also one of several important research efforts in digital communication, and will benefit from evaluation in a variety of different application systems.

(6) Very Large Amounts of Computer Power are Applied as Necessary

Large gains in the utility of sensors can be made using sophisticated computer processing of the raw data, and large amounts of computer processing power can now be made available cheaply using recently developed integrated-circuit technologies. As a result, a large amount of computing power can be located with each sensor, with the potential for improved sensor performance and good results from cheaper sensors.

B. SYSTEM CONFIGURATION OPTIONS

Three different general configurations for DSNs are: area, linear, and honeycomb. Examples of each are shown in Fig. II-1. The 10-km basic spacing is the nominal spacing for the 1A system. In the 1A system, each node has a packet radio, substantial processing capability, an acoustic array and/or a small radar. The basic 10-km spacing has been selected to be consistent with the communication system capabilities, acoustic sensor capabilities, and terrain masking difficulties for low-flying aircraft. Also shown (Fig. II-1) are nodes on a 50-km spacing grid that are denoted as sector nodes. These are surveillance-data collection points and are discussed in Section E.

The 10-km distance, like other parameters appearing in the version-1A design, is neither unchangeable nor is it readily changed. Changing the value can have substantial global side effects. We have selected a single value and just note possible difficulties that might be encountered by increasing or decreasing the value.

Decreasing the 10-km value results in DSN systems that may be less cost-effective. Much of the node cost is electronic and subject to tremendous decreases in price over the next 10 years. Very large-scale integration (VLSI) chips may someday result in the ability to furnish node processing for only a few thousand dollars. In this case, much smaller spacing options may become more interesting. This might involve an acoustic-only system or the use of extremely low-power, cheap, omnidirectional radars. Decreasing the 10-km value would then increase problems with radars interfering with each other. This should not be a problem for the strawman system since the radars do have some limited amount of directionality. Increasing the 10-km value exceeds the useful range of acoustic sensors and introduces masking problems with radar.

Version-1A design efforts have concentrated on area coverage geometry [Fig. II-1(a)]. This is primarily because the uniform distribution of nodes will tend to generate a greater communication load than other configurations. Thus, if the communication issues for the area geometry can be handled, the others should cause no difficulty. The design could be adapted to either the linear or honeycomb situation. In the case of the linear configuration, the overall system would tend to be much simpler. For the honeycomb, the system might be slightly simpler but the problem of data association and tracking across sensor-free areas would be more difficult. In the long run, an area system should function when large holes occur in the DSN. Thus, the area system must ultimately treat the problem of association and tracking across sensor-free areas.

C. OPERATIONAL MODES

Our strawman DSN has both small acoustic arrays and small radars at each node. This provides for operation in a number of possible modes. A brief description and discussion of three such modes follows. The discussion is in terms of modes of a single system, but it should be clear that they could also correspond to systems using different mixes of sensors at the nodes.

1. Acoustic-Only Surveillance and Tracking

Detection and tracking can be accomplished using measurements made only by small acoustic arrays located at each system node. Individual arrays are used to obtain a series of single-node detections and estimates of possible target azimuths. Such data are combined from several such nodes to both perform surveillance for new targets and to track targets.

In such an all-acoustic mode the system is passive except for the communications. It would use less power than with radars in operation and the elements of the DSN would be less detectable than with the radars in operation. The lower detectability follows because radiated power needed to communicate over 10-km distances is very much less than the radar power needed to detect targets at 10-km ranges. However, in an all-acoustic mode the quality of present location estimates and predictions of future positions would probably not be as high as would be possible when making use of radar. This is at least partially due to the inherent acoustic-propagation time delay to the sensors.

2. Radar-Only Surveillance and Tracking

Each DSN node has a small radar that can give good range information but very limited azimuth and no elevation information. As noted previously, if the system operates using only these radar sensors it would consume more power, be more detectable, but probably be capable of more precise estimates of present and future positions. The improved locations would follow from multisite combining of range measurements and not from the use of radar azimuth measurements. Since the radars furnish information for both surveillance and tracking the most obvious mode for individual radars is to scan and search for targets continuously. The system can perform tracking by combining reports from individual radars that are functioning physically in the search mode.

In a hybrid mode, some radars would scan some of the time, and keep beams physically pointed at specific targets some of the time, to accomplish better tracking of already acquired targets. Multiple sites would still be required for good location, but range measurements for targets being tracked would not be constrained to one observation per radar scan time in the surveillance mode. The utility of this hybrid mode and the impact on radar complexity and cost are issues to be considered.

The hybrid mode assumes enough radar redundancy so that some fraction of the radars can be diverted from the surveillance task and redirected to improve tracking information on selected targets. This observation is true independent of whether radars are steered mechanically or electrically since the transmitted radar beam will be directed in only one direction at any one time.

Our strawman system does not contain a directional acoustic receiver that is moved physically to point at targets. Thus, there is no acoustic-only hybrid mode that is entirely equivalent to the hybrid radar mode. The closest equivalent would be to process data selectively for different directions and frequencies rather than to scan the entire space of interest routinely. This could potentially reduce processing loads, but we do not currently know how to do this without compromising performance.

3. Simultaneous Acoustic and Radar Surveillance and Tracking

Making use of both acoustic and radar sensors is the option that generates the richest system alternatives. Rather than try to mention a large number of them, we cite only a simple scenario that represents a prime alternative to be investigated.

To minimize system detectability and conserve power a potentially useful mode of operation is to perform surveillance using acoustic sensors and to operate radars only on cue for tracking. Such a mode of operation is indicated schematically in Fig. II-2 for the situation where a target is entering the DSN from outside of the DSN coverage area. Initial detection and rough location is obtained acoustically as the target enters. The radars in the immediate and adjacent target areas are triggered into operation. As the target goes deep into the DSN the cueing of radars will be from extrapolation of target tracks. Internal acoustic sites will furnish backup as well as continuous surveillance and target-signature information. Our current view is that individual radars will operate in the scanning mode with tracking done from the radar measurements obtained. However, the more complicated radars suggested in the context of the all-radar-hybrid mode could also be considered.

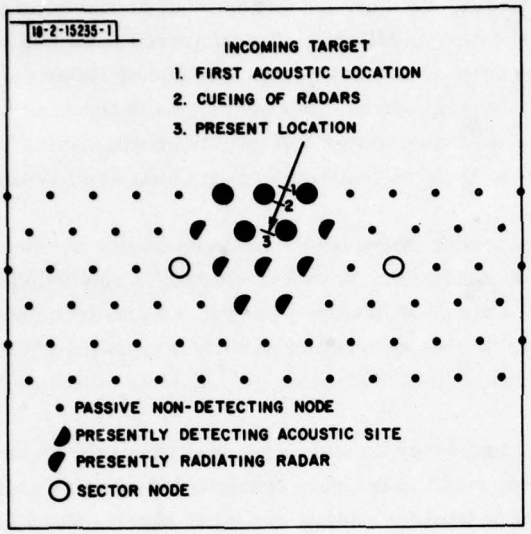


Fig II-2. Potential cooperative use of acoustic and radar sensors. Radar tracking cued by acoustic surveillance conserves power and maintains low probability of detection.

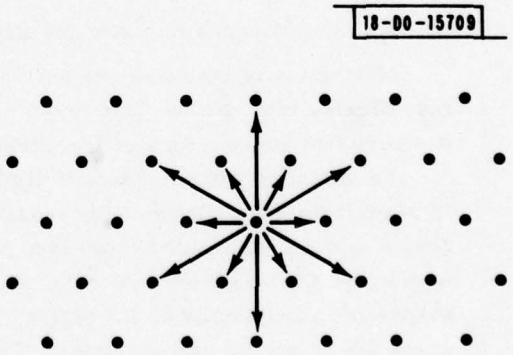


Fig. II-3. Nominal 12 nodes that may typically receive a one-hop message from a transmitting node.

D. MULTISITE PROCESSING

Multiple-site target detection and tracking functions are to be performed at every DSN node independently of other nodes. Each node will use all the information available to it for this purpose. The input information will consist of sensor reports and tracking reports received from neighbor nodes. In general, only reports received directly on a single radio hop from a neighbor will be required or used for basic multisite processing. Figure II-3 shows the 12 nodes that we assume would nominally receive a one-hop sensor report from a single node. This assumption is discussed further in Section G-1.

The sensor reports generated by a node summarize target information that can be obtained by analysis of the data obtained from the sensors at that node without reference to any other nodes. For acoustic sensors, this can be a list of the most likely directions for targets and a summary of spectral characteristics for sound from each of those directions. For radar, this can be a list of possible ranges and rough azimuths plus some target doppler information. Tracking reports combine sensor reports from several nodes plus other available tracking reports to give actual target positions and trajectory information in a directly usable system coordinate system. The tracking reports may also summarize other target signature or identification information.

There are a number of reasons for making every node a multisite node. One is the built-in redundancy that this achieves. In effect, a target may be under surveillance by, and be included in, the tracking reports of several nodes. Loss of a few nodes, apart from the fact that sensor data from those nodes will no longer be available, will not have a serious impact on system performance. Another factor is the anticipated computational load of multisite processing and sensor processing. In the current design, the computational requirements for sensor processing at each node are very substantial. A belief yet to be confirmed is that basic multisite-processing computational load is not substantial relative to the basic sensor-processing load. Thus the cost of routinely doing multiple-site processing with high redundancy will not substantially add to computational hardware in the system. Another factor involves communications. With each node being a multisite node we are able to design a system in which all sensor reports require only on a single communication hop. By excluding multiple-hop routes for sensor reports we avoid an increase in sensor communication requirements that would occur for multiple-hop distribution.

At this time, the multisite processing of radar data has not yet been specified and only the general ideas of minimal acoustic multisite processing have been considered. We include here a brief discussion of an approach to multisite tracking for acoustic data. More specific and detailed algorithms and procedures will be specified in the process of implementing simulations of a functional DSN.

First we consider the multisite use of acoustic data for location. Measurements of acoustic azimuth are made at each DSN node every two seconds. The node locally tries to associate successive observations so that it can deliver sequences of azimuth measurements that are associated with a single target. The node cannot locate the target, but it can often furnish this association between sequences of observations. Thus the multisite process has available as input a collection of lists of azimuths from several sensors with each list corresponding to measurements of a single target from a single node. Some lists may have weak links and some lists which should be a single list may be broken into two or more due to difficulties in correctly

associating sequential observations. The basic multisite task is to use these data to locate and track targets.

As described in more detail in the March 1978 SATS report, each azimuth versus time curve can be mapped into a possible position curve in the horizontal plane. A surveillance time, T, earlier than the present time, is associated with a possible position curve and the mapping from azimuth versus time to possible positions depends on T. Imagine that T is fixed and that all of the possible position curves are displayed using a different color for each contributing node. It should be relatively easy for a person to locate targets using such a display. The targets are where segments from different nodes intersect. The DSN will require a program to sift this display and find targets. Results reported in the March 1978 SATS show good results with only two nodes and only one target, provided the target is not in certain areas relative to the two nodes where its position becomes ambiguous.

Exactly how radar will fit into the picture depends on yet undecided details of the radar and how it is to be used in the system. The strawman radar capability discussed in Section II-G supplies good range information, but poor azimuth resolution. Each such radar measurement then, corresponds to a possible position curve which is a segment of circular arc. This fits nicely into the above picture if the time of the radar observation is the time T. Such radar-possible position curves would, typically, be shorter than those generated by acoustic sensors. This could be used to resolve ambiguities and possible ghosts that could not be resolved completely using only the acoustic data. However, there may be problems when the radar observations do not occur at time T. This a problem to be investigated in conjunction with continuing design and first functional simulations of a DSN. One possibility is that radar data, as mentioned in our discussion of system modes, will not be used extensively for initial target detection and location, but will be used only for refining tracks for targets (and false targets) tentatively detected and located acoustically. This would be the case if individual radars were focused on specific limited potential target areas rather than being operated in a general surveillance mode.

Independent of just how radars are used, there is always the problem of refining the track estimate of a target given the ever-increasing amount of data. There are considerable possibilities for track improvement as more data arrives. Most likely, some form of Kalman filtering will be used for this purpose.

E. REPORTING TO SECTOR NODES

For reporting purposes, DSN geometries are divided into reporting sectors. Each sector has at its center a sector node with which system users interface either directly or through communication links. The reports from tracking programs operating at every node in the sector must be summarized and surveillance information for the sector must flow to the sector node. The flow of reports from the edges of the reporting sector to the sector node is by a patterned sequence of broadcasts. The pattern is such that during normal operation and under good communication conditions the reports in a sector are collected with time delays of one second. Figure II-4 shows the configuration of reporting sectors for normal system operation. Normal operation is defined as operation when all sector nodes are functional. The reporting sectors are defined so that the sector associated with a sector node constitutes all of the area that is closer to that sector node than any other sector. As shown, it happens that in addition to the sector node itself, the reporting sector contains 30 sensors of which 12 are on the boundary.

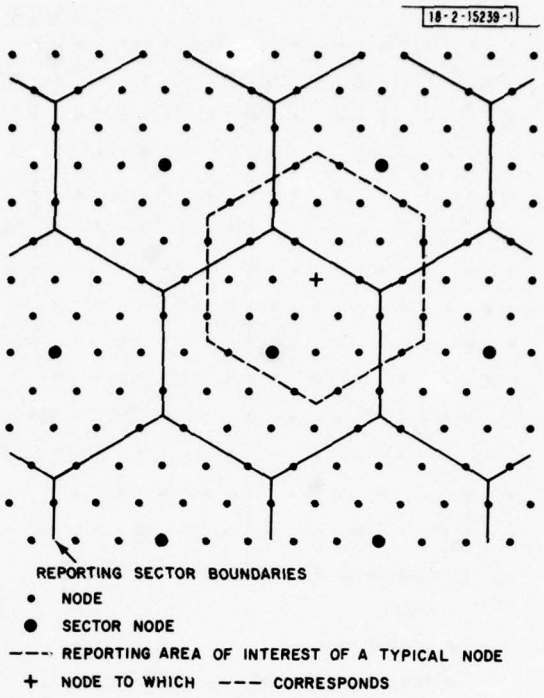


Fig. II-4. Reporting sectors for a fully operational system.

Reports flow from the edges of reporting sectors to the sector nodes as follows. Each node produces and broadcasts a report roughly once each second. This report is the best report summary that the node can generate for an area around the node, which is known as its report area of interest. Figure II-4 also shows the area of interest for a typical node when the DSN is in full operating condition with all sector nodes functioning. Each node's report covers a particular area and the areas for different nodes overlap extensively. In preparing its report, a node uses all reports it has heard as well as its own multisite processing of sensor reports. It is clear how the sector node can generate a picture of its sector from the reports it obtains from neighbors (Fig. II-4); and report coverage is quite redundant so the sector node can get reports for a particular part of its sector from many paths. This compensates automatically for malfunctioning nodes, communication errors, and missing paths. Provided there are not too many communication link failures, the sector node will obtain new sector surveillance information with no more than one-second delay. More link failures may increase the possible delay. Of course, the sector node will also produce and broadcast its sector report once each second.

Under normal operation, as described, each node in the DSN produces and broadcasts a complete summary report once each second for a region of the same size and shape as the report sectors associated with the sector nodes. Each node's report is limited in length to about 1500 data bits. This is enough to accommodate 10 targets. If there are up to 20 targets, the extra targets will be reported at the cost of halving the reporting rate by sending half the report each second. Targets in some region may be "preferred" and sent every second. Beyond 20 targets, node computers must discard targets of no interest from some reports. For example, slow targets may be reported 2 seconds out of every 8. The performance of the system is not optimal above 10 targets and system behavior is not yet specified above 20 targets.

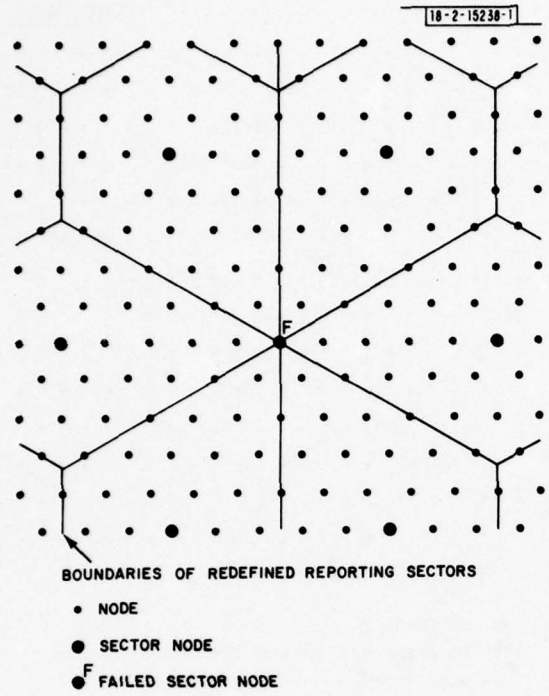


Fig. II-5. Adaptive redefinition of reporting sectors in the face of a failed sector node.

Saturation of the 1500 bits of a report by excessive numbers of targets is a problem that can be more serious when some sector nodes are not functioning and their report sectors must be absorbed into other report sectors. Figure II-5 shows the redefinition of report sectors that results when a single sector node fails. Each neighbor has increased the area of its report sector by one-sixth and problems can result. The report area of interest for nodes other than the sector nodes, if unchanged from that shown in Fig. II-4, is not sufficient to accomplish the flow of surveillance information from the area near the failed sector node to the still-functioning sector nodes. The shape of the report area of interest could be maintained and the size increased. However, this would mean that the same number of bits would be available to summarize a larger area. The result would be a reduction in the target density, which could be handled. The alternative (actually selected) is to allow different shapes of report areas of interest for different nodes. For example, nodes near a functioning-sector node and on the side toward a failed-sector node will have adjusted areas of interest to encompass areas further away toward the failed node without increasing the size of its area of interest. The nature of the communication pattern used to cause reports to flow to sector nodes is such that reports from the area near the failed node will reach functioning nodes with delays on the order of two seconds if there are not too many communication-link failures.

These issues would be more serious if more nearby sector nodes were to stop functioning.

F. COMMUNICATIONS

The DSN system will make use of packet radios for communication. However, the DSN application and traffic is quite different from that for which packet radio networks have been designed. The normal Packet Radio Net (PRN) application is for large numbers of independent

nodes each generating bursty traffic with low average rates. Although there are statistical delay specifications, occasionally, very large delays may be experienced and tolerated. Errors and lost messages are universally not tolerated and the system is designed and organized to enforce this as much as possible. In general, there is no guaranteed performance for any individual user.

The DSN requirements differ substantially. The basic DSN is an organized set of nodes working cooperatively. It is most likely that high target-density in an area will cause DSN communication traffic to be high in that area. Satisfactory performance must be assured at such critical times. In particular, some DSN functions will require assured data rates and delays even in the worst conditions. Fortunately, the fact that the DSN nodes are not independent should allow us to coordinate their communication and thus provide this essential guaranteed response. In some cases, lost messages or messages received in error (which can be treated as lost messages) may be somewhat tolerable.

In this Section we review a very simple packet radio model being used for preliminary DSN communications system design and analysis, review the major DSN communication requirements, and outline a strawman approach for using packet radios to furnish the required service. Only area surveillance with a uniform grid of sensor nodes is discussed. This is the most stressing of the DSN system geometry options from the communication point of view. We have used a regular hexagonal grid with the understanding that it must be possible to adapt to any reasonable regular or irregular distribution of sensors.

1. Assumed Packet Radio Characteristics for Initial Design Purposes

The DSN project will use packet radio for communications. In general, the characteristics of an upgraded packet radio (UPR) are assumed if they differ from the current packet radio.

Our model of the packet radio is very simple. We assume that in any given 10 millisecond period a packet with up to 1500 data bits can be transmitted. The actual number of bits in the packet may be as large as 2000 with the extra 500 being error control and other information not considered specifically to be data. We assume that the error control allows the receiver to detect and discard all packets received in error. Nominally, this packet might be heard by any neighboring node, 10 to 20 km away, with error rates discussed hereafter.

We assume that the time a packet is transmitted is under the control of the node's DSN computer. We assume that a node cannot receive more than one packet at a time. We also assume that if two nearby neighbors transmit overlapping packets, then the two packets will collide and neither will be received correctly. (We do not use the potential of UPR to receive the first packet and ignore the second.)

For the purpose of preliminary system analysis required to confirm that the DSN design might work we have made simple assumptions about communication network connectivity and probability of correct reception of a packet over a single hop. We assume that 70 percent of all single-hop transmission paths of 10-km length will be usable. The rest will not be usable due to siting, transmission path effects, and masking. For paths of 17-km length (second nearest neighbors) the probability of usability is reduced to 50 percent. Longer paths are assumed to not be usable at all. Finally, the single-hop error-free transmission over a usable path is assumed to be 70 percent. Such a figure should be achievable in practice.

2. General Communication Requirements

The DSN communication system must furnish several kinds of service:

(a) Sensor reports
(b) Surveillance reports and command distribution
(c) Special point-to-point.

Each of these is discussed briefly below.

(a) Sensor Reports Service

Each sensor node will broadcast a sensor report once each second to all of the tracking sites that are within one communication hop. Such reports are a summary of the results of processing recent sensor data. In general, a node may take more than a second to update its local world view based only on its own sensors. For example, the current design calls for this to be done once every two seconds for acoustic data and once every four seconds for radar. Thus, with this situation, more than one sensor report will be available to distribute each new world view generated from the sensors at the node.

A sensor report will contain up to 2000 bits. Of these, 1500 may be data and the rest communication headers, error detection, time-and-site information, and other overhead. Given a message of this size, each tracking node will receive on average $(6 \times 0.5 + 6 \times 0.7) \times 2.0 = 14.4$ kilobits and transmit 2 every second. This follows from the geometry and crude link usability model discussed above.

There may be circumstances in which a single-hop link from a sensor to a tracking node that should use it will not exist. Tracking computers will be allowed to arrange to obtain data from selected multi-hop paths, but only insofar as the system has capacity to supply these data without harming the operation of the overall system. This will be done using the special point-to-point service. It is generally discouraged. The communication capacity for such multi-hop service is part of the capacity required for special service and is not considered to be part of the basic sensor report service. The basic sensor report mechanism is to be one hop.

The detailed use of the 2000 bits in a sensor report is not totally specified yet. However, it is not likely that the number of bits (and thus the traffic) can be reduced substantially. Although the typical number of real targets in range of a sensor will be small – perhaps one or even less – the system must be capable of handling several targets and, as a direct result of trying to report targets with poor signal-to-noise ratios (SNRs), will normally deliver several false alarms each reporting period. Thus if only 10 possible targets are reported (sum of actual target measurements and of possible ones, which, in fact, cannot be verified by multiple sites) we find that only 150 bits are allocated to each possible target. In general, we can think of the sensor report as a summary of important characteristics of frequency-wavenumber or range-azimuth-doppler power distributions rather than target reports as such. From this viewpoint, 150 bits would be used to describe each of 10 interesting looking peaks in those maps.

(b) Sector Surveillance and Command Distribution Service

The DSN trial design includes surveillance computers located every 50 km on a uniform grid. Each of these collects a summary of target information over the whole area within about 25 km of where it is located. It does this by collecting surveillance information from tracking computers within its surveillance sector. Tracking information is passed inward towards sector nodes.

In the absence of any communication problems the DSN must be able to report once each second on at least 10 targets in the normal sector surveillance area without regard to where in the area they are located. If more targets are present it may be possible to obtain one-second reports, but they will not be assured. In general, for more targets in the area the sector surveillance computer can either be selective of the targets it tracks or use more than one second to complete a single surveillance survey of the sector. If sector nodes are not functioning and sector surveillance areas are redefined and expanded, larger delays are allowed. Specifically, if there are no communication link failures, then approximately one-second delay is allowed for each normal surveillance sector that a report must traverse. In all situations, the delays-per-surveillance sector may increase if there are substantial communication failures.

Some 1500 bits should be sufficient to represent the surveillance information for 10 targets in the area. If so a node broadcasting its normal area of interest surveillance report as discussed in Section VI will broadcast 1500 data bits with 500 bits of error correction and other packet-header information. The concept that each node will broadcast a surveillance report each second results in 14.4 kbps of incoming surveillance reports and 2.0 outgoing at each node as was the case for sensor reports.

In addition to reports flowing into sector nodes, there will be a flow of commands from the sector nodes to the nodes within the surveillance sector. This will be low data rate compared to the reports themselves.

(c) Special Point-to-Point Service

Some users will need assured real-time point-to-point service in the DSN, for example, alternate routing when the standard one-hop-sensor-report distribution service is not adequate for a particular tracking computer. Such users should be guaranteed a communication rate and delay. The circuit will have assigned a probability that any particular message will actually be transferred from end to end. That probability will usually decrease linearly with the number of hops in the overall circuit. At any node along the route of the circuit the user must be able to extract and add data to messages as long as the outgoing link remains within the rated capacity of the circuit. There will be needs for assured service with various capacity. For example, the alternative routing of sensor reports to tracking computers would require 2-kb circuits. Here we only discuss the maximum assured capacity that the DSN must offer for users.

A point-to-point user could be located at a sector surveillance site and need to examine in detail a great deal of the data at a single tracking site in the sector. In this case, the rate to the tracking site is small – limited to a few commands – but the rate into the sector surveillance site could be large. The tracking node of interest could change as often as every 30 seconds to keep up with a target flying close to the speed of sound. On occasion, such data may also be passed on to a user many surveillance sectors away.

The high-rate data of interest might be sensor reports, sections of frequency-wavenumber power maps, range-azimuth-doppler maps, or even raw data. Acoustic data may likely be one kind of raw data. It should be possible to at least obtain raw data from a single channel of acoustic data. Such data are gathered at a 2-kHz rate with 16-bit samples. Assuming a very nominal factor of two data compression, this will give a 16-kbps data rate. Other data of interest might be all the sensor data reports being used by a particular tracking computer. This would amount to at least 8.2×1.5 kbps and probably somewhat more. Another might be a radar clutter map of $100 \times 36 = 3600$ power values. Based on such considerations we require the DSN to be able to deliver point-to-point service at rates up to 17 kbps.

If 17-kbps circuits are to be supported, then any node must be able to deal with 34-kbps total-circuit traffic, at least. This is the sum of the incoming and outgoing traffic. It can be split arbitrarily between input and output, in general, and each stream can, in fact, be any number of smaller capacity circuits. Each of the individual circuits will have its own error rates and guaranteed capacity and delays.

Delay characteristics for point-to-point transmission should be similar to those for surveillance reporting. That is, delays on the order of one-second-per-surveillance-sector traversed are desirable. Under best conditions such small delays will be achieved, but delays up to several seconds per surveillance sector will be allowed and sometimes experienced.

In general, a request for circuit service must be made, the system will decide if it can be supported, and it will be granted or denied. Priority requests may result in "ungranting" currently guaranteed service.

3. Strawman Communication Organization

The communication needs for DSN are considerable. As outlined in the requirements section above the message traffic (sum of incoming and outgoing) at a single node can easily be 60 kbps or more. To achieve this kind of throughput using packet radio, special effort will be required. Suppose the basic packet radio rate is 200 kbps. Random access or carrier-sense access to such a channel could seriously constrain DSN performance. There are various options to help the situation. One is a system very much like packet radio, but with rates increased by almost an order of magnitude by increasing bandwidth. A second is to use a layered PRN to effectively increase capacity. This would involve running several PRNs simultaneously in the same frequency band, but with different codes. Each radio would have multiple decoders and encoders of the signal. A third option is to recognize the structured nature of the DSN system and to make use of this to coordinate node transmissions so that the available channel is very heavily used and self-interference is minimized. This last is the option that is further developed in this section to indicate how a packet radio with a 200-kbps rate might furnish all DSN communications. Minor changes, such as having a broadcast mode as well as a point-to-point mode and allowing packets that do not get retransmitted if not acknowledged will be required.

(a) Overview of Version-1A Circuits

Version-1A DSN communications is modularized by splitting all communications into separate "circuits." A circuit is a particular conceptual and algorithmic construct for communications. A circuit is a time-ordered geometrical pattern of packet transmissions. Roughly, it is a statement that specific nodes will transmit packets in some specific order during a given time. The algorithmic details of choosing which nodes transmit when are flexible. But there must be an explicit algorithm for each type of circuit.

We assign specific time intervals to each circuit. Some circuits have the potential to interfere substantially with other circuits and we separate them in time to avoid this. Other circuits may not seriously interfere with each other, and need not be separated. It takes 10 milliseconds to transmit a maximum-length 2000-bit packet. Therefore, we will divide each second into 100 intervals of 10 milliseconds each, and assign some number of these 10-millisecond intervals to each circuit.

Version 1A defines three kinds of circuits: sensor, report, and special (query) circuits. These correspond to the three kinds of services discussed in the requirements section. Of the 100 10-millisecond time intervals in each second, 25 are allocated to the sensor circuit, 25 to the report circuit, and 50 to the various special circuits.

The sensor circuit moves messages from each node to all its neighbors. There are no acknowledgments, and any data that are to arrive with high probability must be retransmitted repeatedly. The messages in the sensor circuit are sensor reports.

The report circuit moves surveillance messages from each node to sector nodes, which, in turn, are expected to move the messages to a manned computer center. The report circuit also moves occasional control messages in the reverse direction from the sector nodes to all other nodes. Acknowledgment is optional for each message. Because of the many paths available to or from sector nodes, an unacknowledged message has quite a high probability of arriving at its destination, though this probability goes down when report circuit traffic becomes very heavy and the number of paths is reduced to accommodate the load. Report circuit messages are broadcast so that they may be heard by all surrounding nodes, and thereby, are also the means by which each node knows what its neighbors are tracking.

Special circuits are established on request between any two nodes for data rates up to 17 kbps in one direction, and a low data rate in the reverse direction. Usually these circuits are used to query a node and transmit data, principally from the node to the nearest sector node. Acknowledgment and retransmissions are automatic for special circuits, except that messages can be marked to be thrown away if the backlog in the special circuit exceeds one of several limits (which, in normal situations, happens very rarely).

Higher level system functions will generally use both report and special circuits.

(b) Communication Patterns

We assume that simultaneous broadcasts by packet radios separated by at least 50 km cannot interfere with each other for any DSN purpose. Therefore, we want to organize communication so that only nodes separated by at least 50 km will broadcast simultaneously. This is done by breaking the DSN into communication cells with 50-km sides and arranging for no more than one node in a cell to broadcast at any given time. We illustrate the idea here in terms of the reports circuit.

The report circuit performs three functions:

> Conveys report messages from arbitrary nodes to sector nodes from whence they are sent to manned control centers.
>
> Broadcasts tracking results from each node to all neighboring nodes as an aid to the tracking programs in neighboring nodes.
>
> Broadcasts control messages from each sector node to all surrounding nodes.

The report circuit is a circuit with a typical broadcast pattern (Fig. II-6). The nodes within the parallelogram constitute one communication cell. Such cells are repeated throughout the DSN. With rather minimal local coordination the pattern will effectively avoid communication conflicts. The pattern is repeated once every second. As shown, the sequence of broadcasts is 1A...1F, 2A...2F, 3A...3F, 4, 5A...5F. Such a pattern can be used to move data toward or away from the sector nodes.

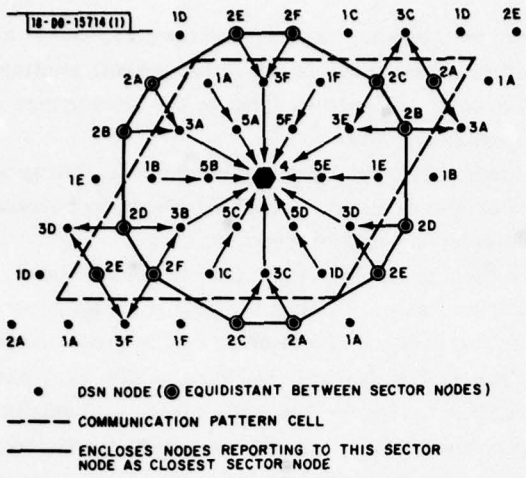


Fig. II-6. Report circuit broadcast pattern. Broadcast sequence is cyclic: 1A...1F,...5A...5F, 1A, etc.

To see the flow away from the sector node, consider the cycle starting with 4 and proceeding, 5A...5F, 1A...1F, 2A...2F. This can be used to move data away from one sector node toward another in case the first sector node is nonfunctional, or has lost touch with the manned computer center. Because it is capable of moving data away from sector nodes, the report circuit is also used to broadcast control messages originating at manned computer centers and routed through the sector nodes.

The sequence, 1A...1F, 2A...2F, 3A...3F, 5A...5F (4 is not used) obviously, can be used to move data towards the sector node. Since the cycle is repeated once each second, reports from the more distant nodes in the sector will be refreshed each second as long as there are not too many single-hop communication failures. As communication failures increase, the information flow paths to the sector node become less redundant, and eventually, the probability of a single-cycle path existing will become low enough so the effective refresh time will become larger than one second.

This communication scheme assumes that the DSN can control the time of a broadcast. Also, the flow over multiple hops is not accomplished strictly by multi-hop communication paths under control of the communication system. The DSN computers have access to messages in transit and modify these messages before they are retransmitted. For example, in moving data toward the sector node the node at 5C in the pattern has probably heard 1C, 3B, and 3C with enough lead time to incorporate their information in the packet it will transmit.

Finally, we note that even with separation of 50 km between simultaneously transmitting nodes there may be some self-interference of communications. If so, the code-division multiple access (CDMA) capabilities of UPR can be exploited to solve that problem.

G. SENSORS AND SENSOR DATA PROCESSING

As noted in other sections of this chapter, each strawman DSN node includes a small acoustic array and small radar. In the following brief paragraphs we summarize the general strawman sensor characteristics and processing load required for those sensors. These sensors and loads are only representative. This is particularly true of the radar option for which there is a need for considerable design and analysis work before a sensor can be proposed with much confidence. Low-flying targets, in general, and cruise missiles, in particular, present difficult sensor

problems. More detailed discussion of acoustic-signal processing and sizing will be found in other chapters of this report.

Each node in the strawman design includes an array of 10 high-quality microphones deployed over a plane aperture of three meters. Digital data are collected at the rate of 2000 samples per second per channel. Once every two seconds one-second's worth of array data are processed to detect targets and estimate the direction from which sound is coming at a large number of different frequencies. Alternate one-second intervals of data are not collected and the time may be used for sensor calibration or other purposes. Single-node target detection and direction estimation makes use of high-resolution frequency-wavenumber signal processing. Detection and estimation depend on finding increases in the amount of power that appear to arrive at the array as a function of direction and frequency. The single-node processing load amounts to about 11-million real adds or mutliplies per second.

A small monostatic, coherent-pulsed-doppler, two-dimensional radar has been tentatively selected as the strawman DSN radar. The strawman radar at this time is not a serious proposal for a specific radar, but rather, is a somewhat arbitrary selection of some possible radar characteristics for the purpose of sizing computational requirements. The radar frequency and other characteristics are subject to substantial changes as a result of further study. Even the type of radar may be changed, but it is likely that our present selection would be the most computationally demanding alternative so it has been selected as a worst-case starting point. The strawman radar is L-band (1.3 GHz). It uses a 10-kHz pulse repetition rate (PRR) to operate out to about 12.5-km ranges. The antenna has approximately a 10-degree azimuth resolution and a 30-degree elevation beamwidth. The radar scans 360 degrees using 36 beams displaced from each other by 10 degrees. Nine beam directions are used each second. State-of-the-art digital processing is used to detect targets in radar ground clutter and to make most effective use of doppler shifts resulting from target motion. Targets are located within 120-meter range cells, but with poor direction information. The processing load, about half of which is fast Fourier transform (FFT) computation amounts to 25-million real adds or multiplies per second for a single such radar.

H. NODE HARDWARE CONFIGURATION

The hardware configuration for each node in the Version-1A DSN is shown in Fig. II-7. Neither the sensor nor the packet radio are discussed further in this Section. The other electronic hardware is assumed to be built with currently existing integrated circuits. We have specified this hardware to avoid compromising system performance due to the use of slower commercial equipment. The specification of considerable computer hardware at each node will give the most flexibility for software options. Even more computer power would be technically feasible, but no well-defined requirement has yet developed. The option shown is sufficient for the worst-case radar and acoustic signal processing at a node with enough spare power for other DSN functions. Ongoing design, simulation, and experimental efforts will be used to determine if the processing capability represented by Fig. II-7 will actually be required.

Except for a small amount of sensor specific and interface electronics the electronics consist of a main computer, memory, and single-instruction multiple-data (SIMD) stream processor. The outline of a design for the main electronics has been completed and the number of integrated circuits required for each electronic subsystem have been roughly estimated. For the estimates obtained and shown on the figure the cost of the entire node built to military specifications would

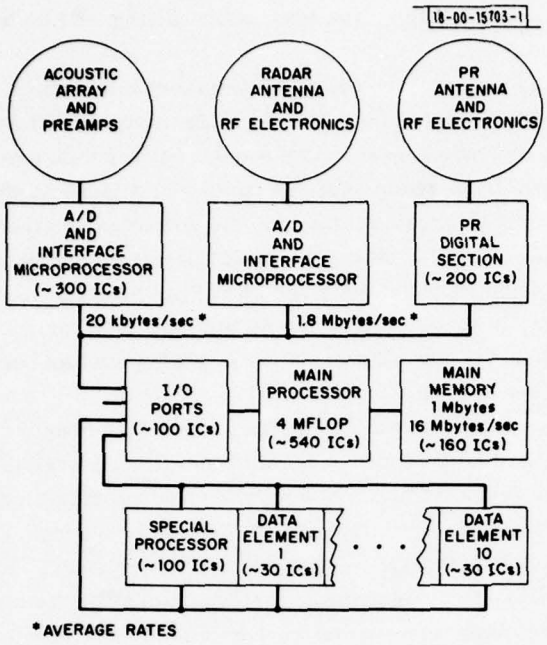


Fig. II-7. Strawman hardware configuration for a DSN node.

be $120,000 if we assume that system cost is proportional to the number of integrated circuits and the system cost per circuit is $60. The $120,000 figure is reasonable at the present time and will probably decrease by a substantial percentage, between 30 and 50 percent each year in the immediate future as the number of chips required are reduced. The design and these estimates do not make use of special developments in very large-scale integration or of special custom chips. Such alternatives could substantially reduce cost, size, and power consumption.

The main processor serves general-purpose computing requirements and also as an input/output (I/O) channel processor for main memory. In its role as a general-purpose processor it will perform all processing at the node, which is not more appropriately done by the faster, essentially parallel, SIMD processor. This could include a substantial amount of sensory artificial intelligence (AI) processing. Such sensory AI would be to make more effective use of multidimensional power maps produced by single-sensor signal processing as the first step toward target detection and parameter estimation. The strawman design has the main processor running an extension of the PDP-11 instruction set so that it can emulate the PDP-11 with high efficiency. The main differences are:

> Instruction lengths are not the same: the instructions have been reformatted to allow 32-bit addressing, 32 registers, etc.
>
> Instructions must be treated as read only because the main processor design includes an instruction cache that is not updated by data writes.

The main processor, although it is 20 times faster than typical commercial minicomputers in tight-loop arithmetic computations, is still not fast enough to handle the several tens of millions of arithmetic operations per second required for full acoustic and radar data processing. For this purpose, a special arithmetic processor is used. This processor is a 10-element SIMD: a single-instruction stream, multiple-data stream processor with 10 data units and one

instruction execution unit. Both acoustic and radar processing at a single node naturally decompose to be efficiently accomplished using such an architecture. The SIMD design takes advantage of currently available high-speed, 16 × 16, multiplier chips.

Almost all circuitry is intended to be low power. Micro-cycle times are 250 nanoseconds, and only a little processing is done in each micro-cycle. For example, single-precision floating-point adds are done in 750 nanoseconds in the main computer, whereas Schottky circuitry would do them in only 200 nanoseconds. The main exception to low power are 16 × 16 multiplier chips to be used by the SIMD processor. Each of those chips consume several watts of power.

The processor outlined briefly here is not presented as the only practical option for the node electronics. We have pursued a specific option as part of our strawman design only to be able to put on display one option that is feasible and will satisfy our requirements.

III. STRAWMAN ACOUSTIC SENSORS AND SINGLE-SITE DATA PROCESSING

The strawman DSN acoustic sensor mentioned in Chapter II consists of 10 microphones set in a planar geometrical pattern within a 3-m aperture. In this Chapter we present more details about acoustic signals and microphone requirements including description and sizing of the strawman signal-processing algorithms. These algorithms are the basis for target detection and acoustic azimuth determination.

A. ACOUSTIC SIGNALS AND MICROPHONES

The sound pressure levels of the acoustic signals of interest range from 10 to 100 dB, where the 0-dB reference level is 20 micropascals (0.0002 dynes per square centimeter) the threshold of human hearing. (The unit for sound pressure level, the pascal is equivalent to a force of one newton applied over an area of one square meter.) The sensitivity of the microphones that will be used in the small arrays is typically −26 dB with reference to 1 V/pascal or 50 mV/pascal. The microphone output voltage is 1 μV for a 0-dB signal and 100 mV for a 100-dB signal. Maximum output of the microphone, 10 volts, occurs at a sound pressure level of 140 dB.

Microphone and preamplifier noise is usually between 12 and 20 dB or 4 and 10 μV. Ambient acoustic noise levels range between 20 and 70 dB, and it is hoped that a 40-dB processing gain will be achieved. Such processing will allow detection of signals at levels as low as perhaps −10 dB.

The analog voltage outputs of the 10 microphones must be appropriately digitized for signal processing. The large dynamic range of acoustic signals and the precision that must be maintained for all levels of the signals require a sophisticated analog-to-digital (A/D) conversion system for the DSN sensor node. Details of the specification for such a data acquisition system are discussed in Section VI. The A/D system digitizes each microphone waveform at a 2000 sample-per-second rate, and each sample is stored in a 16-bit floating-point number format.

B. ACOUSTIC ARRAY PROCESSING OVERVIEW

The small acoustic-array algorithms that were described briefly and sized computationally in the 31 March 1978 SATS have the capability to detect and compute the acoustic azimuth of a target. However, in these algorithms, power levels are only calculated for approximately 20 azimuths at 20 different frequencies. The computations consider only horizontally propagating acoustic signals; i.e., those signals with phase velocities equal to the speed of sound. Thus, when targets are detected, their azimuths are computed with low resolution – to within roughly 20 to 30 degrees.

The acoustic array data are, however, of sufficient richness that algorithms may be applied to the data to compute azimuth measurements with resolution of 1 to 2 degrees. This "high-resolution" algorithm computes power levels at essentially many more frequency-wavenumber points; and the high-resolution data analysis has at least an order of magnitude more computational and storage requirements.

It was originally thought that the DSN-sensor-node computational requirements could be kept relatively modest by doing the high-resolution analysis only when targets were detected by the lower-resolution algorithm and analyzing the data for only that range of azimuths originally computed by the low-resolution algorithm. Thus, the low resolution would run continuously; a

certain duration of past raw acoustic data would be maintained on a sliding-window basis, and the high-resolution analysis would be performed only occasionally to refine target azimuth measurements.

However, if 6 to 10 targets are in the coverage area, the high resolution would have to be done for full 360-degree coverage, essentially all the time. The computational sizing for the front-end sensor processor would have to include the worst case of the low- and high-resolution analyses running concurrently. Thus, the current approach is to size the DSN-sensor-node-hardware requirements for continuous high-resolution data analysis. There are two major motivations for this decision. First, by continuously using high-resolution analysis on the acoustic data, one always has much more accurate azimuth measurements that can be used in the multisite location and tracking algorithms. One does not have to worry about when to switch between low- and high-resolution algorithms. A processor designed to handle this processing load poses no limit to the number of targets that can be detected and tracked. Furthermore, from a research point of view, the capabilities of acoustic sensors can be more thoroughly examined with high-resolution analysis. Secondly, considering the trajectory of hardware costs, it appears appropriate to design systems assuming very large computational capabilities.

C. DETAILED ALGORITHM DESCRIPTION AND SIZING

The high-resolution frequency-wavenumber analysis required for DSN acoustic arrays is characteristic of signal processing that is done for sonar and seismic arrays. In this Section, the details of the processing are reviewed, the magnitudes of the computation and storage requirements are sized, and it is noted that the processing can be broken into modules, a fact that allows the use of many parallel processors executing the same operations on different data.

1. Overview

There are two basic steps in the high-resolution analysis. First, the acoustic array data are processed to compute estimates of the power-spectral-density covariance matrix for the multivariate-time waveform. Such matrices are often estimated for a set of uniformly spaced frequencies within the Nyquist band. However, one can just as well compute estimates of the matrix for a general, not necessarily uniform set of frequencies, f_k, $k = 1, 2, \ldots, F$, where F is the number of frequencies of interest. In the DSN trial design, F is expected to be on the order of 50, all frequencies between 50 and 250 Hz.

Second, at each of the frequencies of interest, a spatial wavenumber analysis is performed. The number of wavenumber points computed at each frequency is estimated to be approximately 800 to 1000. Both the azimuth and phase-velocity resolution depend on frequency, and the number of wavenumber points that are calculated at each frequency depend on this resolution. The number of azimuths range from 50 to 250 and the number of phase velocities range from 3 to 10.

The algorithm is described followed by detailed sizing of all the arithmetic computations and storage requirements.

2. Input Data

The input data are M sampled waveforms collected from the M microphones in the DSN acoustic sensor. Current specifications call for M = 10 microphones in an array. Each of the microphone waveforms or channels will be sampled at an R = 2-kHz rate. An analysis interval of TA = 2 sec is being considered. During this interval, data from only T = 1.024 sec will be

analyzed – a total of 2048 (R* T) samples per channel. Some of the remainder of the time is used for calibration of the data acquisition system.

The input time series from each channel will be broken up into K = 5 overlapping blocks, each of N = 512 points. Each block is multiplied by an appropriate window and a Fourier transform is performed on the block. Thus, M* K N-point FFTs are performed, resulting in K estimates of the matrix periodograms.

3. Spectral-Density Covariance Matrix

It is important to note that all the computations described from this point are performed for each frequency of interest. As mentioned for DSN acoustic sensors, it is expected that there will be approximately 50 frequencies of interest between 50 and 250 Hz.

Two basic techniques are typically used to estimate the power-spectral-density covariance matrices, $C(f_k)$, $k = 1, 2, \ldots, F$. One is to smooth in the frequency domain the matrix periodogram of all the data for the analysis interval. The other, which is described and sized in this document, is to average (or sum) the K-matrix periodograms. Hybrid techniques, combining some of both methods can be used, but such techniques are not discussed here.

An estimate of the power-spectral-density covariance matrix for frequency, f_k, is given by:

$$C_{ij}(f_k) = \sum_{l=1}^{K} X_i^l(f_k)\, X_j^{l*}(f_k) \quad ,$$

where * denotes complex conjugate and $X_j^l(f_k)$ is the value at frequency f_k of the Fourier transform of the l^{th} interval of time series data from the j^{th} microphone in the array. From a statistical point of view, the stability of this estimate is roughly the same as for the smoothing in frequency method if K is equal to the number of frequencies in the smoothing kernel. The computational sizing takes into account that the $C(f_k)$ are Hermitian matrices. For each frequency, the resulting matrix is then normalized to equalize power in each channel:

$$R_{ij}(f_k) = \frac{C_{ij}(f_k)}{[C_{ii}(f_k)\, C_{jj}(f_k)]^{1/2}} \quad .$$

All the elements along the diagonal of $R(f_k)$ are one.

4. Wavenumber Power Estimation

Wavenumber power estimation corresponds to a discrete two-dimensional spatial Fourier transform with spatial sampling points determined by microphone positions. Judicious selection of the sensor positions can allow utilization of fast transform techniques, but in the current discussion, the general case will be considered.

Several power-spectral matrix modifications are done before the actual estimates of power. First, a small diagonal matrix is added to the $R(f_i)$ to force the matrices to be nonsingular:

$$R(f_k) = R(f_k) + \epsilon I \quad , \qquad k = 1, 2, \ldots, F \quad .$$

Then, the complex $R(f_k)$ must be replaced by their inverses. The matrices are Hermitian, but have no other special properties. Each complex matrix is decomposed into two real matrices that are each inverted and the results recombined for the required complex inverse. The standard gaussian elimination inversion technique has been assumed for the computational sizing.

The final step is to estimate power for each wavenumber of interest for each frequency. Selection of a wavenumber corresponds to computing the power of signals coming from a particular direction (azimuth) with a certain phase velocity. (A phase velocity that is equal to the propagation speed of sound corresponds to a signal propagating horizontally across the small array, while a phase velocity greater than the speed of sound corresponds to acoustic signals that are incident on the array at an angle from the horizontal; i.e., the noise source has significant elevation relative to the array.) In general, both the number of wavenumbers as well as their values are dependent on frequency. If $A(f_k)$ is the number of azimuths at frequency, f_k, and $V(f_k)$ is the number of phase velocities at f_k, then the product, $A(f_k) * V(f_k)$, is the number of wavenumbers at which power will be computed for frequency, f_k.

The estimation of power, given frequency and wavenumber, reduces to calculating a Hermitian quadratic form using $R^{-1}(f_k)$ as the matrix and a complex steering vector, E_{ik}. The elements of the steering vector are complex representations of the phase delay for each of the sensors (microphones) for a given wavenumber. Specifically, the frequency wavenumber estimate of interest is:

$$P_{ik} = \frac{1}{E_{ik}^{H} R^{-1}(f_k)\, E_{ik}}$$

where $i = 1, 2, \ldots, A(f_k) \times V(f_k)$, and $k = 1, 2, \ldots, F$. The superscript, H, implies conjugate transpose. When all the frequencies of interest are considered, this power computation will be done W times where:

$$W = \sum_{i=1}^{F} A(f_i) \times V(f_i) \quad .$$

Typical values of $A \times V$ for the highest frequencies of interest may be as large as 2500. Assuming that the frequencies of interest are distributed uniformly and that both A and V are proportional to the frequency, the computational sizing has assumed that W is approximately $835 \times F$ or 42,000. If the steering vectors were to be precalculated, stored, and looked up as required the storage required would be greater than 1.5-million bytes (each vector has 10 complex elements). Calculating them on the fly would require excess processing. A tentative compromise has been selected for the computational and memory sizing in this Chapter. The memory requirements are 2500 real values and 1000 complex values or a total of 9000 bytes. Each quadratic-form computation requires one steering vector which, in turn, requires only 11 real multiplications for its "look-up" and computation. The computational sizing of the quadratic form also takes advantage of the fact that $R^{-1}(f_k)$ is Hermitian and the magnitudes of all the elements of E_{ij} are all unity.

Power levels are computed at all appropriate wavenumbers for each frequency of interest. The results are examined for peaks with two-dimensional peak-picking algorithms. A significant peak at one or more frequencies most likely indicates a target at a particular azimuth and phase velocity. A time average of the results is kept such that the most recent power-frequency-wavenumber results may be compared to the recent past.

5. Computational Sizing

The parameters of the high-resolution analysis for acoustic data are reviewed in Table III-1. Note that the computations are done on data measured over 1 sec in an analysis interval of

TABLE III-1

HIGH-RESOLUTION FREQUENCY-WAVENUMBER ANALYSIS: PARAMETERS

M = 10 microphones or channels

TA = 2 sec, data analysis interval

T = 1.024 sec, data measurement interval

R = 2 kHz, sampling rate per channel

RT = 2048 samples per channel processed during one analysis interval

K = 5 blocks into which the T data points are divided with approximately 20-percent overlap

N = 512 samples per block

F = 50 frequencies of interest

A = 250, maximum number of azimuths considered in the analysis (at the highest frequency of interest)

V = 10, maximum number of phase velocities considered in the analysis (at the highest frequency of interest)

2 sec. The results of the computational sizing are listed in Table III-2, while the results of the storage sizing are described in Table III-3. It is important to note that all the computational operations listed refer to real operations and the storage requirements assume all data are stored in a 16-bit floating-point format.

From the tables, one learns that the high-resolution analysis requires 22-million arithmetic operations per analysis interval or approximately 11-million arithmetic operations per second. The memory required is approximately 185,000 16-bit words. Such processing requirements represent a significant challenge to the capabilities of current minicomputers even when combined with currently available array processors.

TABLE III-2

COMPUTATIONAL SIZING: HIGH-RESOLUTION FREQUENCY-WAVENUMBER ANALYSIS

Task	Windowing	Fast Fourier Transforms	Covariance Matrix Average and Normalize	Matrix Inversion	Steering Vector	Quadratic Form
Multiplies:	9,236,570					
	MKN	$+ \frac{MKN}{4} \log_2 (N/2)(4)(1.15)$	$+ F\{ M^2(2K+2) - 2KM + 10M$	$+ (2/3) M^3$	$+ \frac{AV}{3}[M+1$	$+ 2M^2+2]\}$
			1200	667	213	
	25,600	235,520	179,509		177,642	
			8,975,450			
Adds:	12,396,780					
	0	$+ \frac{MKN}{4} \log_2 (N/2)(6)(1.15)$	$+ F\{ M^2(2K-1) - KM$	$+ (2/3) M^3$	$+ \frac{AV}{3}[0$	$+ 3M^2 - M - 3]\}$
			845	667	287	
		353,280	240,870		239,358	
			12,043,500			
Divisions:	42,200					
	0	$+$ 0	$+ F\{$ M	$+$ 0	$+ \frac{AV}{3}[0$	$+ 1]\}$
			844		834	

Assumption:
1 multiply = 1 add = 1 arithmetic operation
1 divide = 10 arithmetic operations

Total Number of Arithmetic Operations: 22,055,350

TABLE III-3

MEMORY REQUIREMENTS: HIGH-RESOLUTION FREQUENCY-WAVENUMBER ANALYSIS

	Words*
Input Data (double buffered)	40,960
K complex FFTs (N = 512 points)	51,200
Covariance Matrix and Scratch Pad Memory	4,096
Steering Vector Tables	4,500
Power vs Frequency Wavenumber Tables (latest and average)	84,000
Total	184,756

*A 16-bit floating-point format is assumed.

IV. MULTISITE DETECTION

The exploratory work on simple, decision-theoretic, state-space-search algorithms discussed in previous SATS has been completed. These general algorithms combine data from several sensor sites to detect and locate targets when the data from a single sensor alone may not be sufficient to locate a target. The exploratory work has focused primarily on acoustic arrays that provide only the acoustic azimuth of a target. The algorithms convert a range of possible geographic positions in a well-defined search-space cell at a specified search time into a range of possible azimuth-time measurements at the sensors. The conversion is a simple theoretical calculation based on geometry and the speed of sound. The azimuthal data that are predicted are actually compared to those data that are measured by the individual acoustic sensors. If the data match, as determined by simple interval criteria, the sensor measurement is said to be a corroboration of the hypothesis that a target did generate noise in a particular cell at the surveillance time. A target is declared to be in this search-space cell when data from several acoustic sensors corroborate this hypothesis.

A. IMPROVEMENT OF THE SEARCH ALGORITHM

Studies reported in the previous SATS divided the search space into three-dimensional boxes. Due to the computational load of calculating the exact range of azimuths corresponding to a spherical cut through a three-dimensional cell the range of azimuths was calculated only approximately. It was decided that the approximation could be improved by the expediency of considering only two-dimensional cells located at fixed altitudes and for those cells calculating exactly the possible azimuth ranges. Recent simulations have confirmed that the new approximation is superior.

A two-dimensional search-space cell is shown in Fig. IV-1. Times, t_2 and t_1, define the range of time in which azimuth measurements will be considered. The particular time that an azimuth measurement is made, t_m, is used to reduce the size of the predicted azimuth range to $\delta\beta$ from the larger range, $\delta\alpha$, that is calculated from only the cell sensor geometry. This smaller azimuth range, $\delta\beta$, is then compared to the azimuths measured by the sensor at time, t_m. With the previous three-dimensional cell algorithm the approximation had been to use the large azimuth range similar to $\delta\alpha$ because the calculation of the more restricted interval corresponding to $\delta\beta$ would require too much computation. Thus, in general, the previous "3D" algorithms generated larger azimuth intervals than the "2D" algorithm. Consequently, one expects more "false alarms" or many more corroborations in cells without targets in the cubical case.

To check the above assertion, several simple simulations were run in the same format as those reported in previous SATS. Three targets with velocities, M = 0.53, 0.60, and 0.65, (M is mach number) are moving through the 5 km-square search-space. The respective target altitudes are 50, 100, and 40 meters. The 2D-cell algorithm was run twice: with the assumed altitude of the grid of squares (each 0.5 by 0.5 km) covering the surveillance area at altitudes of 0.0 and 0.65 km, which is the approximate average of the target altitudes. The 2D results were identical at the two assumed altitudes. The 3D-cell algorithm was run with cell dimensions of 0.5 by 0.5 by 0.01 km with the base of the single layer of cells at zero altitude.

During the 20 to 30 seconds in which the targets were in the surveillance area, the search algorithms operated with 20-second delays from real time. The 2D-cell algorithms usually

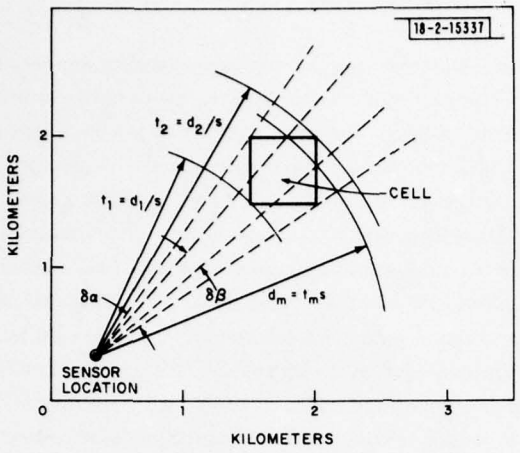


Fig. IV-1. Two-dimensional search-space cell.

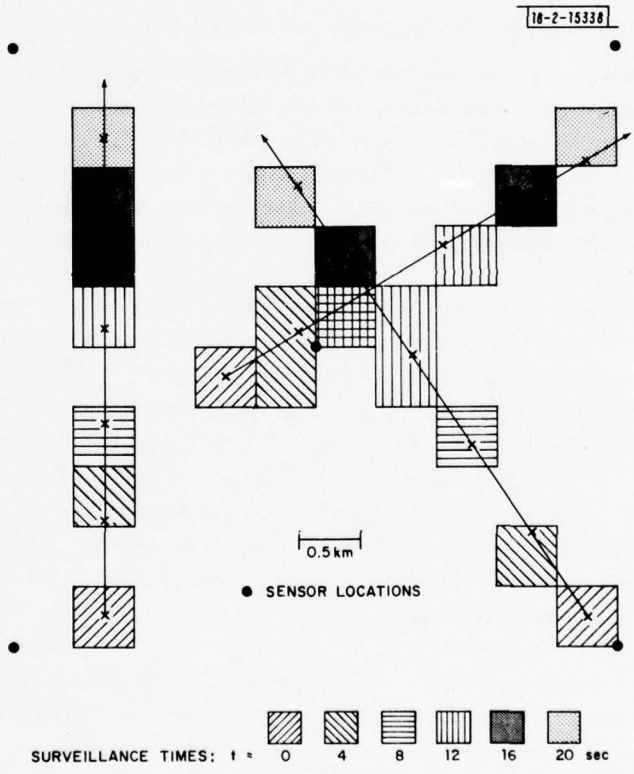


Fig. IV-2. Detection and location of three moving targets. Target cells are those cells with corroborations from all five sensors. Location of targets at surveillance times indicated by x's; target trajectories by solid lines.

located the three cells containing the targets by eliminating all cells with corroborations from fewer than all five sensors. The 3D-cell algorithm usually computed four or five cells with corroborations from all sensors – the three cells with targets and several erroneous ones close to target-containing cells. Thus, the tracking of targets was more difficult since the location of a target at particular surveillance times was often blurred across two or more cells. Furthermore, the 3D-cell algorithm always computed approximately 60 percent more total corroborations that tended to clutter the search and make locating those cells with targets more difficult.

Further simulations have shown that both the 2D- and 3D-cell search algorithms are rather insensitive to target altitude; i.e., as the target's altitude varies over a range approximately equal to the horizontal dimensions of the cell, the results of the simulation do not change. Furthermore, if the altitude of search cells is greater than two or three cell widths of the target's true altitude, the simulation detects targets in cells adjacent to those cells that contain the correct x- and y-coordinates of the target, rather than not detecting targets at all. This result is due to the fact that although targets with identical x- and y-coordinates and different altitudes have the same true azimuth, sound from the target with greater altitude has a longer path over which to propagate and a different acoustic azimuth occurs. When the difference between the cell and target altitudes is greater than three or four cell dimensions, no detections are calculated. Altitude insensitivity was also discussed in the previous SATS in the context of the related two-station detection and location algorithm.

B. HIGHER-RESOLUTION SEARCH ALGORITHMS

The size of the search-space cells of the simulations that have been discussed up to this point has been relatively large – 0.5 km squares or cubes. In these "low-resolution" algorithms, it has been assumed that the sensors calculate azimuth measurements once a second and do not attempt to relate sequential measurements to each other. In other words, it has been assumed that the front-end processing done at the sensor does nothing to identify sequential azimuth measurements with the track of a particular target.

Targets with mach numbers between 0.5 and 1.0 spend between 1.5 and 3.5 sec flying through a 0.5-km cell; thus, there is time for one or more azimuth measurements while the target is in the cell. If the cell size is reduced to 0.1 km (100 meters), such targets spend only 0.3 to 0.9 sec within the boundaries of a cell. Often an azimuth measurement is not made within this duration, corroborations are not calculated, and the target is not located at a surveillance time. Search algorithm simulations that utilize 100-m cells and once-per-second azimuth measurements without track identification fail to locate targets in the above described scenarios. The cells in which the targets are located never obtain corroborations from more than one or two sensors in each surveillance time.

If one assumes that the sensor processor, in addition to calculating azimuth measurements, can combine most sequential azimuth measurements so as to form target tracks, one can then interpolate between pairs of azimuth measurements to always provide an azimuth range at times predicted by the cell-sensor geometry in the search algorithms.

The 2D-search simulation with 100-meter cells was modified to allow linear interpolation between pairs of sequential azimuth measurements and the results are compared to those from lower resolution noninterpolation algorithms in Figs. IV-2 and -3. It is clear that the simulation with higher resolution more accurately defines target locations, but at an obvious computational

cost. There are 2500 search cells (compared to 100) through which to search in the higher-resolution simulation. This higher-resolution situation with interpolation between actual azimuth measurements is essentially a general multiple-site version of the two site location algorithms discussed in the previous SATS.

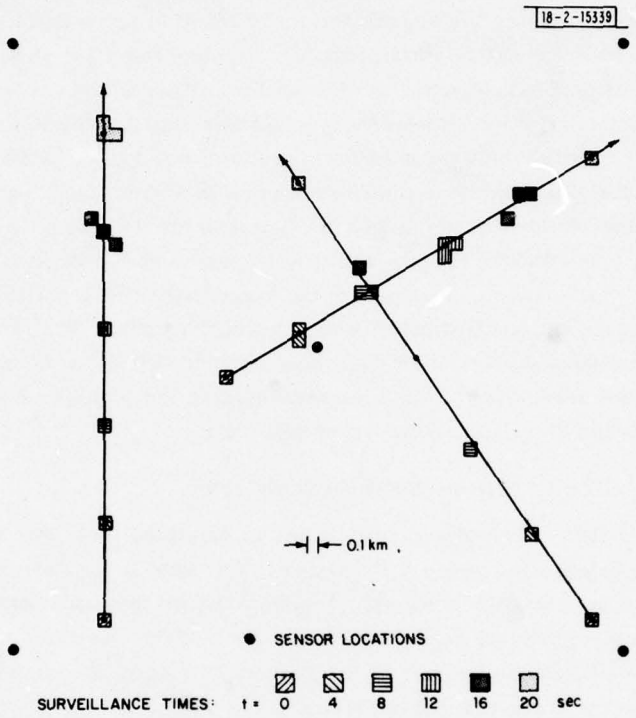


Fig. IV-3. Detection and location of three moving targets using finer search cell grid and linear interpolation between sequential azimuth measurements. Target cells are those cells with corroborations from all five sensors. The locations of targets at surveillance times are indicated by x's; target trajectories by solid lines.

V. ACOUSTIC MODELS, SIMULATION, AND SIGNAL PROCESSING SOFTWARE

A model for an acoustic node in a DSN has been formulated and corresponding simulation software has been written. The model, information about the software, and the results of a few simulations are presented in this Chapter. The model is moderately complete. Future work may require both simpler versions to reduce computation load in total multiple-node system simulations and more complete versions to investigate even more detailed behavior of an acoustic node and associated processing. The software package also includes routines that can be used to process experimental data. The software elements were conceived and realized as the need arose in developing a simulator for a single acoustic node, the maximum likelihood method (MLM) processing routines for wavenumber estimation, and programs to interface experimental data with the analysis tools. The computer system environment is Unix and the computer language is C.

A. ACOUSTIC ARRAY SENSOR MODEL

In developing a sensor model one must make a choice regarding which point in the data acquisition and processing chain to begin the simulation. It was decided to avoid simulating the actual temporal stochastic processes representing the pressure fields at each sensor, and to work from a theoretical ensemble statistic basis. The point of entry is then the spectral-covariance function which is a matrix function of frequency. For a given frequency it is the power-spectral-density matrix whose elements are the auto- and cross-power spectral densities between the elements of the small acoustic array located at a DSN node. It is modelled by assuming ensemble statistics of the targets and their geometrical relationship with the array, along with the ensemble statistics of some typical noise fields. This approach has the disadvantages of bypassing the detailed effects of transduction, electronic analog amplification, anti-alias filtering, digitization, channel imbalance, and spectral-covariance estimation. The advantages of beginning at this point are decreased execution and program development time, and the ability to include noise fields that can not effectively be simulated in temporal form, i.e., that of propagating noise from distributed sources.

1. Target and Propagation Model

Targets are assumed to be point sources of sufficient distance from the sensor arrays such that the pressure fields at the sensors consist of plane waves. The temporal power spectral density at the target is specified by 15 parameters, a_i, σ_i^2, m_i; $i = 1, \ldots, 5$. The spectrum is modelled as:

$$S(f) = \sum_{i=1}^{5} a_i \exp \frac{(f-m_i)^2}{2\sigma_i^2} \quad .$$

The representation as a weighted sum of shifted gaussians permits arbitrarily wide or narrow peaks to be specified easily.

Propagation is modelled by spherical spreading through a homogeneous medium with no absorptive attenuation, thus the power decreases 20 log(r) decibels for propagation over a distance r. Because the separation between sensors is very small compared to the total propagation path, all sensors are considered to receive the same signal power level. The atmospheric

homogeneity assumption degrades the model very little in terms of predicting azimuthal estimation precision since the atmosphere and local environment should be approximately symmetric as a function of azimuth. Time-varying horizontal stratifications would effect only the elevation-angle estimates. The model omits accuracy problems due to steady winds causing horizontal ray curvature.

The model does not presently include excess attenuation above that associated with spherical spreading. A short discussion of this issue is included here. Typical atmospheric attenuation for a 100-Hz tone at 20°C and 70-percent humidity is 0.018 dB/100 m. Thus a 1.8-dB loss is encountered over a 10-km propagation path. At 200 Hz, the figure rises to 8 dB for the same 10-km path. These figures indicate that the absorption loss can be disregarded in comparison to the 80-dB loss due to spherical spreading over the same distance, and it is not necessary to include these losses in a first-order model. However, when a substantial portion of the propagation path is through foliage surrounding the array, the model may need to be altered. Additionally, all channels are assumed to be matched in gain so that the total signal power output obtained from each of the channels is identical.

For modelling purposes, the state of the target is defined completely by its position and velocity, and we make the additional assumption that geometrical considerations allow for identical doppler shifts to be perceived at all sensors. We also consider only targets with speed less than the speed of sound. Under these constraints, and assuming that the time series generated by the source is a locally stationary random process, we may write the power- and cross-spectral densities at the sensors for a given time as a function of the target-power-spectral density, the target state, and the sensor locations. The cross-spectral density for the i^{th} and j^{th} sensors is:

$$P_{ij}(f) = \frac{\gamma S(\gamma f)}{r_i r_j} \exp\left[j2\pi f\gamma \frac{r_i - r_j}{c}\right]$$

where

$\gamma = [1 - v_r/c]$

v_r = the radial velocity of the target with respect to <u>both</u> sensors (m/sec)

c = velocity of sound in air (m/sec)

$S(f)$ = target-power-spectral density at 1 m

r_i = distance from target to the i^{th} sensor (meters)

f = temporal frequency (Hz).

Since air is assumed to be a linear medium at the sound pressure levels we encounter, and the sources are assumed to be independent, the cross-spectral density for multiple targets is just the sum of the individual contributions.

2. Noise Models

The model takes into account three types of noise that might be present at the array. The first, and most important, is spatially independent and homogeneous random noise introduced by wind turbulence around the microphone and thermal noise in the channel electronics. The second is isotropic noise that plays an important role in underwater acoustics, but is much less

important in atmospheric acoustics. Finally, we include ring noise, so named because it is the field emanating from a ring of distributed sources in the plane of the array.

a. Spatially Uncorrelated Noise

Spatially "white" noise is generated primarily by local turbulence caused by wind passing the individual microphones. System thermal noise will typically be insignificant relative to real acoustic noise. One-third octave-band measurements for 40- and 30-km winds on microphones with and without windscreens have been normalized to 1-Hz bandwidths and are graphed

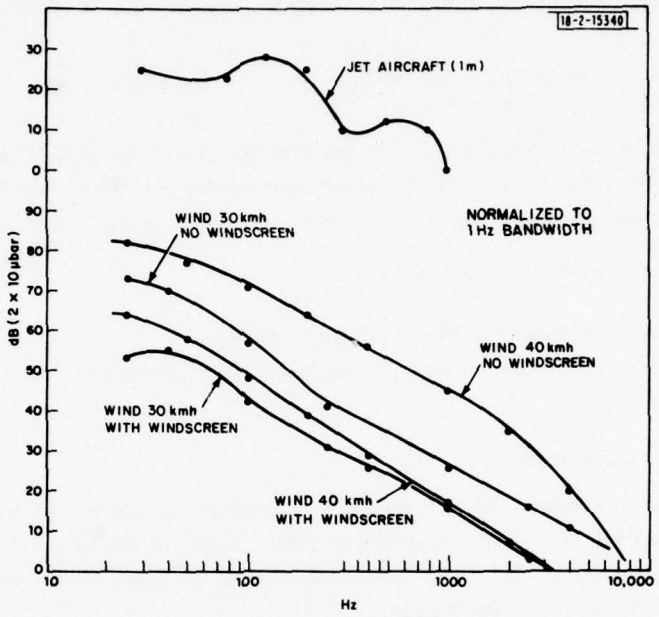


Fig. V-1. Measured wind noise and jet aircraft spectra. Measurements made with 1/3 active smoothing. Aircraft data adapted from "Acquisition, Reduction, and Analysis of Acoustical Data," NADC Report No. AWG-SU Dept. of the Navy, Naval Development Center, Warminster, PA (1974). Noise data adapted from L. L. Beranek, Noise and Vibration Control, (McGraw-Hill, New York, 1971).

in Fig. V-1. As one can see from the graph, the data are flat at low frequency and roll off at 25-30 dB/decade at high frequency. The model generates the theoretical noise-spectral-covariance function as:

$$P_{ij}^{w}(f) = \sigma_w^2 \frac{f_o^2}{f_o^2 + f^2} \delta_{ij}$$

where f_o is the nominal cutoff frequency, σ_w^2 is the power in the noise below the cutoff frequency, i and j are the sensor indices, and δ_{ij} is the Kronecker delta function. The roll-off is thus modelled as 20 dB/decade. The data indicate that f_o should be 30 Hz and σ_w^2 is chosen according to wind conditions and microphone characteristics. The windscreens used for the measurements (Fig. V-1) are optimized for intelligibility and speaker artifact reduction. It is probable that

better windscreens for true wind noise reduction over the frequency band important to aircraft detection will be developed, so that we may obtain even better noise reduction in the future.

b. Isotropic Noise

Isotropic noise is defined as a superposition of planewaves propagating from all directions with a uniform statistical level. Since a temporal simulation would require generating an infinite sum of infinitesimal time series, we can clearly only include this noise field in a simulation starting at the ensemble statistical level.

It can be shown that the frequency-wavenumber spectrum, which can be measured by a planar array in a three-dimensional isotropic noise field, is given by:

$$P_n(f, \hat{k}) = \frac{S_n(f)\lambda^2}{[1 - (\lambda|\hat{k}|)^2]^{1/2}} \qquad |\hat{k}| < 1/\lambda$$

where $S_n(f)$ is the power/unit bandwidth distribution in frequency, $\hat{k}$, is the two-dimensional vector wavenumber, and λ is the wavelength of the sound at frequency, f.

Taking the inverse Fourier transform in space, leads to a spectral-covariance function:

$$P_{ij}^n(f) = S_n(f) \text{ sinc } (2\frac{\pi}{\lambda} r_{ij})$$

where r_{ij} is the distance between sensor i and j.

There are currently no data available on sound pressure levels of this type of noise in the atmosphere.

c. Ring Noise

One would expect a greater amount of propagating noise with wavenumber magnitudes close to $1/\lambda$ emanating from surface sources rather than a uniform level from the entire hemisphere above. The limiting case is a ring of sources in the plane of the sensors. For this model, the frequency-wavenumber function is:

$$P_R(f, \hat{k}) = \frac{S_R(f)\,\delta(|\hat{k}| - 1/\lambda)}{2\pi/\lambda}$$

which leads to the spectral-covariance function:

$$P_{ij}^R(f) = S_R(f)\, J_o \frac{2\pi\, r_{ij}}{\lambda}$$

where J_o is the zeroth-order Bessel function and all the other variables are as defined above.

There are currently no data available on sound pressure levels of this type of noise in the atmosphere.

3. Discussion of Signal-to-Noise Ratios

Figure V-1 shows a noise spectrum for a jet aircraft. Actual data were measured using a 1/3-octave-band spectrum analyzer on measurements made at a distance of 10,000 feet. For the figure, these were corrected for spherical spreading back to a one-meter distance and no allowance was made for absorption losses. The corrected spectrum shows a 130 dB peak at 125 Hz. Since these data were taken with a 1/3-octave-band analyzer, the fine structure of the spectrum is lost. It is probable that the actual spectrum has more of a line nature, and the

third-octave plot is indicative of the average of a 1-Hz resolution spectral estimate. An array with windscreened sensors in a 30-kmh wind might have 40 dB of spatially uncorrelated sensor noise at 125 Hz. Thus, the SNR at the sensors, disregarding absorption losses, is approximately 10 dB for a 10-km target-to-array distance (disregarding other directional sources). One should realize that the figures stated are all averages, and the instantaneous SNR may vary wildly due to propagation-path disturbances and local wind conditions.

4. Model Limitations

The major weakness of the model just described consists of its omission of the imperfections of estimating the spectral matrices. Using even the simplest method of covariance estimation, the dyad of a vector consisting of one complex frequency sample from the discrete Fourier transform of a block of data from each channel, one can encounter gross nonstationarity within the interval required for one time-series data block to be acquired (typically 0.25 to 1 sec). The nonstationarity arises as a result of the nonzero velocity and accleration of the target. Within one block length, the position of the aircraft and its doppler shift seen at the array may change dramatically. As an example, if we consider a target travelling at mach 0.5, with a closest point of approach of 250 m and a time block length of 0.25 sec centered on data generated at the time of closest approach, a received spectral line would change 8 percent in frequency during the block. The azimuth of the target would also change about 8 degrees during the same period. The problems with nonstationarity of this sort also precludes the use of large amounts of spatial-covariance matrix averaging to generate more stable estimates for some target scenarios. However, note that these examples are very extreme and the model does retain its validity for targets at distances greater than 1 km, and for those that travel more slowly.

B. SOFTWARE

The first step in the development of a software package for modelling, simulation, and processing of experimental data was to begin to standardize the functions and structures needed, and to consolidate them into one flexible and (hopefully) easy-to-use tool kit. This approach will allow fast software development, easy communications between members of the research team, and prevent unnecessary duplication of effort. To this end, an effort has been made to develop in the language "C", under the Unix operating system, a comprehensive numerical processing library, a set of data structures for recurrent data types encountered in the types of processing foreseen, and the basic graphics and I/O routines needed for operator interaction with the data and processing results. Such software is described in this Section. It constitutes a first version of part of the software that will eventually be used for both single-site and total-system simulations as well as actual software for future experimental efforts. This software will be undergoing continuing modification and expansion as we proceed to more simulation and experimental work in the future. Also outlined are some interactive analysis and simulation packages that have been built out of the blocks.

1. Building Blocks

There are several elements of the basic software:

(a) Data structures

(b) Numerical routines, particularly for matrix operations

(c) I/O routines
(d) Graphics routines
(e) Memory management and error handling.

Following is an overview of the important data structures and major matrix operations. More details, including information about I/O, graphics, and memory management will be available in a separate working paper under preparation.

a. Data Structures

Various data structures whose subject-to-change description constitute part of our software package and evolving standards, are described in this Section.

The matrix (including the vector as a special kind of matrix) is a very common data type involved in signal processing, detection, and estimation algorithms. A convenient construct for passing arbitrarily typed and dimensioned matrices is required, and the explicit passing of additional parameters such as size and data type must be eliminated. To achieve this, instead of passing a pointer to the first element of an array, passing explicit dimensions, and using the data type declared in the subroutine to infer a structure, matrices are passed via a single pointer to a data structure. This data structure called "ma-struct" includes information specifying the dimensions of the matrix, byte requirements for each element, and the starting address of the actual data. The current structure is adequate for up to three dimensions, respectively denoted, planes, rows, and columns, and has an additional parameter for singling out one "plane" so that 2-dimensional operators can be passed 3-dimensional arrays without ambiguity. The structure also includes the data type of the elements of the matrix so that no a priori information about the matrix is needed by a routine operating on the matrix.

There also exists a clear need for easy manipulation of complex quantities in general numerical procedures but the C language does not include a complex data type. To provide this, we have defined a complex structure that is treated as one element. In subroutine links, scalar complex quantities are passed via pointers to the address of the first byte of the real part. Of course, some knowledge by the subroutine of the data precision is required to interpret correctly the data passed. The present standard is double precision, thus one complex number requires 16 bytes of storage.

There are also structures designed for more specific needs. These include the "out" structure, which is usually defined as one element of a vector and is used to store the indices and value of one array element that has been singled out for some reason. The "info" structure is used to hold additional information pertaining to a matrix, typically, information regarding the information needed to make plots of matrices in some form. This may eventually be coalesced into the matrix descriptor structure "ma-struct" described above. The "loc-struct" data structure is typically one element of a vector. It consists of three elements, x, y, and z, and is useful for passing an array of 3-space coordinates. The "etab" structure is used as one element of a matrix and can hold the wavenumber, angular coordinates, and power estimate at the stored wavenumber. The "desc" structure is used as a header to data files containing complex frequency estimates. It specifies the indexing information associated with the list of values to follow, and the array that was used to obtain the data.

b. Numerical Operators

Following is a summary of the main general matrix operations that have been implemented. Existing routines for manipulating matrices were written to manipulate complex matrices since that was the immediate need.

DECOMP	reduces one plane of a 3-dimensional array to upper triangular form and saves pivot information. Numerical stability is enhanced by partial pivoting in the gaussian elimination algorithm used.
SOLVE	uses the triangularized matrix "A" from DECOMP and a vector argument "b" to solve Ax = b for the vector "x".
CINV	computes the inverse of a planar matrix using DECOMP and SOLVE.
DOTPROD	computes the dot product of two vectors: $z = x^H y$, where H denotes conjugate transpose.
NORMAL	normalizes the diagonal elements of a planar conjugate symmetric matrix to 1, and divides the off-diagonal elements by the square root of their corresponding row and column diagonal elements.
OUTPROD	computes the outer product of two vectors: $R = xy^H$.
OUTPRUD	is the same as OUTPROD, but adds the result of the outer product to the previous contents of the R matrix.
PKLOC	finds all local maxima (excluding boundary points) of a planar matrix or vector.
QUADR	computes the quadratic form of the vector "x" with the matrix "A" and returns the double precision result, "y": $y = x^H Ax$.
STABIL	computes A = A + Ia where "A" is a plane matrix, "I" is the identity matrix, and "a" is a real scalar.
TRANSFO	computes y = Ax where "A" is a plan matrix and "x" and "y" are vectors.
MULTM	forms the general matrix product AB where "A" and "B" can be selected planes from higher-dimensional matrices.
SUMC	computes a general linear combination of matrices: A = aA + bB where "a" and "b" are complex. SUM is a version with "a" and "b" real.
GRAMS	Orthonormalizes a set of vectors that are the rows of a planar matrix. The rows linearly dependent on the previous rows are returned as zeros.

2. Higher-Level Analysis and Simulation Packages

The building blocks outlined above were brought together in several routines to synthesize some useful functions. The following is not a complete list of routines, but is an outline of most of the functions available.

GETDAT (GET DATa)

This is the main driver routine from which all simulation and processing routines can be called. The functions available from GETDAT include:

- constructing sensor location files with the proper format
- reviewing sensor files
- generating estimates of the covariance matrices at selected frequencies, and the ability to print them out.
- magnitude, phase, real part and imaginary part plots of the frequency data to be processed.
- calling the MLM frequency-wavenumber estimator to operate on power-spectral-density matrices.
- calling the source simulator (SOURCE)
- generation of simulated spectral-density matrices via the estimation procedure used on real data.

MLM (Maximum Likelihood Method)

This routine performs MLM wavenumber estimation on selected planes of the three-dimensional matrix supplied. The main functions provided are:

- normalize and invert the spectral matrix with optional numerical stabilization, and perform power estimates over a specified grid size and density in wavenumber space.
- estimate the power of a single location in wavenumber space.
- plot a contour map of the wavenumber spectrum.
- locate all local maxima in the wavenumber spectrum and store their energy and location parameters for later output.
- plot a slice through the estimator output matrix.

SOURCE is the simulator for generating the theoretical spectral matrices for arbitrary target scenarios and noise fields. The functions available are:

- define, store and recall files containing the geometry of sensor arrays.
- define, store and recall particular target scenarios and spectral statistics.
- calculate and display beam patterns for arbitrary planar untapered arrays.
- move targets according to stored velocities and positions.
- generate the theoretical spectral-covariance function for the target scenario and array geometry.

- generate the theoretical spectral-covariance function for frequency-dependent, spatially uncorrelated noise, isotropic noise with a flat temporal spectrum, and ring noise with a flat temporal spectrum.
- call the MLM routine to process the synthetic data.
- display all model parameters and plots of target point spectral densities.

C. RESULTS

The software described above can be used to simulate part of the functions of an acoustic node and to process actual acoustic data. Both have been done and some results are described here. First we show an example of a simulation result based on the theoretical spectral-covariance matrix for the case of wind noise and two aircraft. In the future it will be possible to use more extensive runs of the same kind to generate moderately realistic single-acoustic node inputs for multiple-site DSN simulations. Second, the results of simulation of a single-dyad spectral-covariance-matrix estimator and frequency-wavenumber analysis using that matrix are presented. This simulation indicates the degenerate multiple-target results that can be obtained using the single-dyad estimator. Finally, the results of analysis of real multiple target acoustic data are presented. The analysis of real data also confirms the limitation of single-dyad spectral-matrix estimation, confirms the improvement that can be obtained using multiple-dyad estimates, and demonstrates some of the rich alternatives for processing and interpretation of acoustic data.

Figures V-2 and V-3 show wavenumber processing results obtained using theoretically predicted power-spectral-density matrices. The model software was used to construct power-spectral matrices for situations with two targets located 5 km from the array and for microphones with windscreens operating in a 30-kmh wind. The aircraft were modeled as independent sources radiating at 100 Hz. For Fig. V-2 both sources were 130 dB and for Fig. V-3 the level of the source to the northeast was reduced to 120 dB. Using the data from Fig. V-1 for microphones with windscreens, a 30-kmh wind, and 130-dB sources at 100 Hz, we predict an 11-dB SNR at the array for sources at 5 km. In Fig. V-2, both sources are at 130 dB and the wavenumber processing output shows two sharply defined peaks at the correct locations with the same magnitude. The two real peaks are about 17 dB above all secondary peaks and perhaps 20 dB above the average background in the figure. For an input SNR of 11 dB this situation must roughly correspond to an array gain of $20 - 11 = 9$ dB. This compares favorably with the theoretical beamforming gain for the simulated array which is 9.5 dB. Figure V-3 shows the simulated wavenumber analysis obtained with one of the sources reduced to 120 dB. For that lower level source, the predicted input SNR is only 1 dB. Both peaks are still sharply defined and the lower is still 8 to 10 dB above the noise floor, approximately the processing gain.

The same wavenumber analysis for the same situations as Figs. V-2 and V-3 was repeated, but using a theoretical single-dyad estimator of the aircraft-signal-spectral matrix rather than using the theoretical spectral matrix. The theoretical estimator roughly models the estimation of an element of the spectral matrix as the product of the Fourier transform of the channels corresponding to the row and column indexes of the matrix. From a matrix point of view this is the same as forming a column vector with the i^{th} element corresponding to the Fourier transform of the i^{th} channel in the array and estimating the spectral matrix as the outer product of this vector with itself. Hence the term, single-dyad estimator. For the simulation, an identity

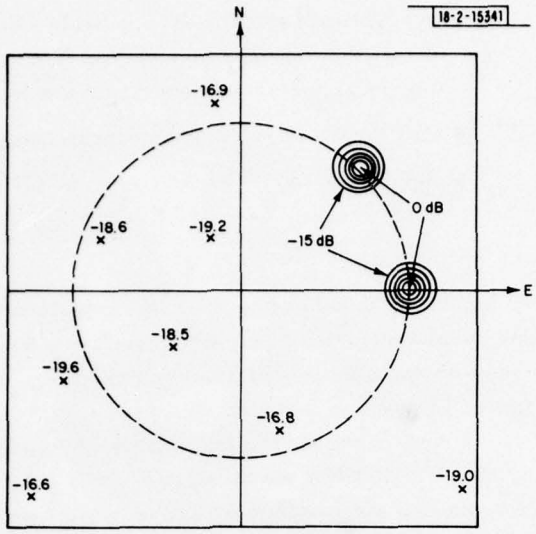


Fig. V-2. Results of maximum likelihood wavenumber analysis of theoretical spectral covariance matrix generated by model. Dashed circle indicates locus of points corresponding to a horizontal speed equal to the speed of sound. Situation modelled was two 130-dB sources located 5-km away. Source directions are east and northeast. Noise conditions for microphones with windscreens in a 30-kmh wind were used. The frequency of the analyses and modelling is 100 Hz. Contours shown are at 3-dB intervals. Below the lowest contour the x's denote the locations and dB values for all local maxima.

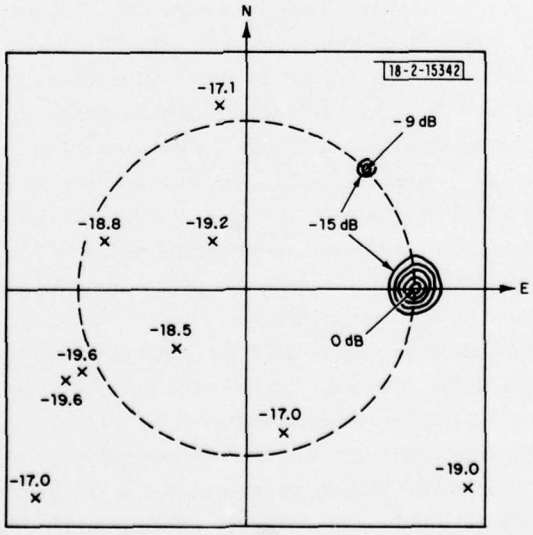


Fig. V-3. Results of maximum likelihood wavenumber analysis of theoretical spectral covariance matrix. Situation same as for Fig. V-2 except that power level of source to the northeast is reduced by 10 dB.

with enough power to represent the wind noise was added to the outer product of the theoretical signal vectors. The single-dyad estimator has been used in the past to obtain results we have reported in previous SATS reports. Here we indicate its serious shortcoming in the presence of multiple targets.

The difficulty with the single dyad is that data consisting of the sum of two phase vectors from two independent point sources will generate extraneous cross terms in the outer product. In order to illustrate this, let $X_1(f)$ and $X_2(f)$ be spectral vectors from two sources, i.e., if $x_i(t)$ is the vector of acoustic time-series observations due to source i, and T_n is an observation interval of length T, then let

$$X_i^n(f) = \frac{1}{T} \int_{T_n} x_i(t) \exp[-j2\pi ft]dt \quad .$$

With this definition, the theoretical single-dyad spectral-matrix estimator for the sum of the two signals is:

$$P(f_o) = [X_1(f_o) + X_2(f_o)]\,[X_1(f_o) + X_2(f_o)]^H$$

$$= X_1(f_o)X_1^H(f_o) + X_2(f_o)X_2^H(f_o)$$

$$+ 2\mathrm{Re}\,[X_1(f_o)X_2^H(f_o)]$$

where the first two terms are the theoretical spectral-covariance matrix. We would like the third term to be zero, but in general, that is not even approximately true.

Fortunately, an estimator constructed by averaging simple estimates of the above form over time is asymptotically unbiased and consistent. That is, the third term averages out to be zero if enough independent terms are averaged. An intuitive way to see that this might be true is to consider x_1 and x_2 as independent random processes with nonzero bandwidth. If we do a complex spectral analysis of x_1 at time, t, the component at f_o will have some magnitude and phase, $X_1(f_o)|t_1$. The same will be true for $X_2(f_o)|t_1$. Since the Fourier transform is linear, the output will be $X_1(t_o) + X_2(f_o)|t_1$. If we then do another spectral decomposition at $t = t_2$, the nonzero bandwidth of the processes will cause the phase relations of processes $x_1(t_2)$ and $x_2(t_2)$ at frequency f_o to have changed. Thus, the real parts of $X_1(f_o)\,X_2^H(f_o)$ will assume another random value, with mean value zero, whereas the $X_iX_i^H$ terms will, of course, still be coherent with themselves, and the estimator should approach the ensemble value as more terms are averaged (within the limits of stationarity). The practical difficulties of obtaining a valid estimate in this fashion are a function of the bandwidth and the stationarity of the processes, $x_1(t)$ and $x_2(t)$.

The results of using a theoretical single-dyad spectral-matrix estimator as described above are shown in Figs. V-4 and V-5. In Fig. V-4, for sources of equal strength the wavenumber processor failed to locate the peaks correctly, and the peaks that did occur were barely above the noise floor. In Fig. V-5, with the source from the northeast at 10 dB lower than the source from the east, the processor located the stronger peak correctly, but with not nearly the SNR output of the Fig. V-3 case, and failed to locate the weaker source correctly. In both cases, the noise floor was not as low as in the previous results with the true covariance matrix.

Finally, a series of experiments using real data have been completed that confirm the predicted performance of acoustic arrays using single- and multiple-dyad spectral estimates and

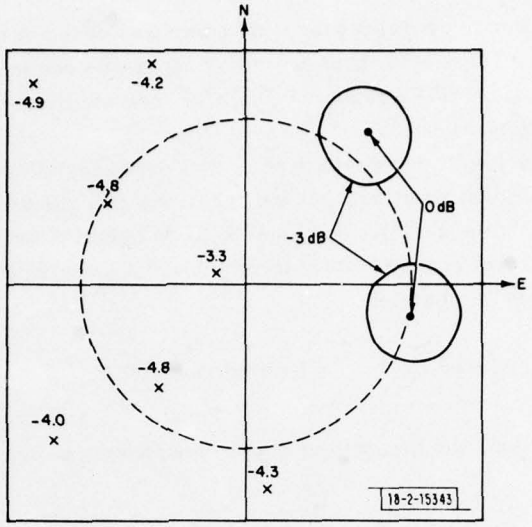


Fig. V-4. Results of maximum likelihood wavenumber analysis of theoretical single-dyad estimator of the spectral covariance matrix. Situation identical to Fig. V-1 except that a theoretical single-dyad estimator is used in place of the actual theoretical spectral covariance matrix. Only −3 dB power contours are shown.

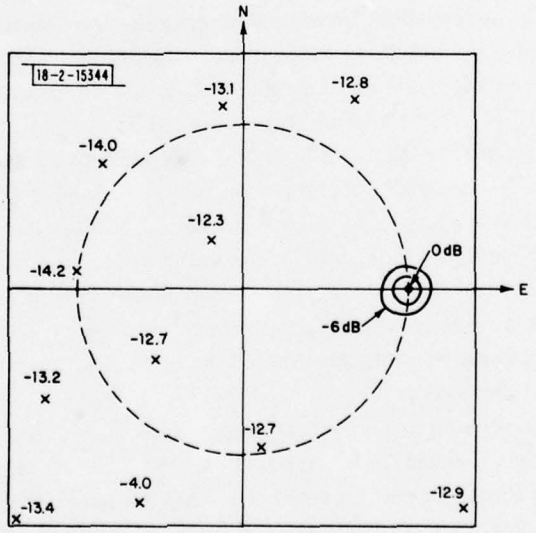


Fig. V-5. Results of maximum likelihood wavenumber analysis of theoretical estimator of the spectral covariance matrix. Situation identical to Fig. V-2 except that a theoretical single-dyad estimator is used in place of the actual theoretical spectral covariance matrix. Only −3 and −6 dB contours are shown.

also further demonstrate the richness of the acoustic array data source. Figures V-6, V-7, and V-8 show typical single-channel spectra for a U-8 propeller aircraft and an A-7 jet aircraft. The U-8 is approximately 6.7 km due west of the sensor site and the A-7 is approximately 8.6 km due east. These data were added together to obtain time series containing both aircraft as well as noise. The spectrum for the sum is shown in Fig. V-8. In addition to the targets there are spectral contributions present from a generator located northwest of the array. There is also a great deal of system noise at 60 and 180 Hz, probably due to grounding problems. The U-8 data by itself, the A-7 data by itself, and the summed data were all used in array processing experiments reported below.

Figures V-9, V-10, and V-11 show the results of wavenumber processing of the summed time series at frequencies selected so that either the U-8 or A-7 source dominated. In Fig. V-9, the single-dyad estimator is adequate to make a good detection of the A-7 at 105 Hz where its power is very large compared to the U-8. Figure V-10 shows that the single-dyad estimator is able to detect the U-8 at 117 Hz where its energy is approximately 8 dB greater than the A-7. Note that the A-7 is not located at all. The results in Fig. V-11 for a 5-dyad average at 117 Hz show that the A-7 can be detected in the presence of the more powerful U-8 with the better estimator, and the processor output noise floor is decreased by about 3 dB overall.

Figures V-12 through V-15 document further the improvements obtained by time-averaging covariance matrices but now for the case of targets of approximately equal strength. Figure V-12 shows a single-dyad detection obtained using the A-7 data corresponding to the spectrum shown in Fig. V-7. The frequency of analysis was 100 Hz. Figure V-13 gives the results of using a single-dyad spectral-matrix estimator and wavenumber analysis at 100 Hz for the U-8 data corresponding to the spectra shown in Fig. V-6. At 100 Hz the U-8 noise is not significant and the most important noise source is a motor generator set located to the northwest of the sensor array. The peak corresponding to this is clear in the figure. The poor results of trying to simultaneously locate the generator and A-7 in the sum of their time series via a single-dyad spectral-matrix estimator can be seen in Fig. V-14. Note that no correct detections were made. The results of averaging eight covariance dyads to yield the final estimate are shown in Fig. V-15 for the same general time period as was used for Fig. V-14. In this figure, both targets are detected very distinctly with an overall SNR better than the single sources alone with the simpler estimator.

It is informative to review these last processing results in the context of our theoretical model for a node to see if real data are consistent with the model. From Fig. V-15 the processing output noise floor relative to the A-7 is at about -15 dB. Our wavenumber analysis using a theoretical spectral matrix showed an array gain of 9 dB. If we assume that 9 dB was achieved for the experimental data then the input SNR for the A-7 (exclusive of the generator noise) on the summed data was about $15 - 9 = 6$ dB. A range of 8.6 km gives about 79-dB loss due to geometrical spreading and a microphone with windscreen in a 30-kmh wind would have about 42 dB of wind noise at 100 Hz (Fig. V-1). For these conditions and the observed SNR the implied source sound level is $42 + 79 + 6 = 127$ dB. The correct value for this aircraft is not known but the value of 127 dB is certainly reasonable in the light of the smoothed, but typical data shown in Fig. V-1.

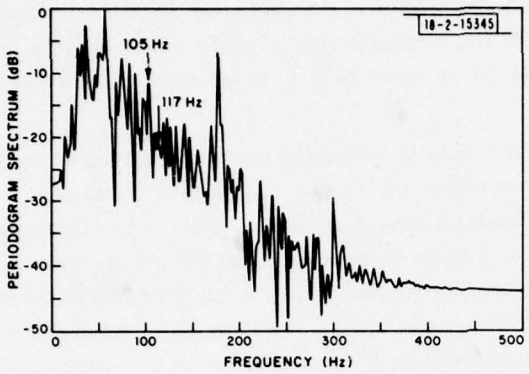


Fig. V-6. Periodogram spectrum from a single microphone recording of a U-8 aircraft approximately due west of the array. This unsmoothed spectrum was obtained from 1024 data samples uniformly taken over a period of approximately one second. Spectrum normalized to peak value.

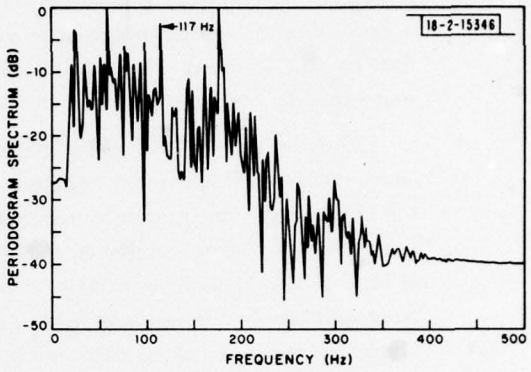


Fig. V-7. Periodogram spectrum from a single microphone recording of a A-7 jet approximately due east of the array.

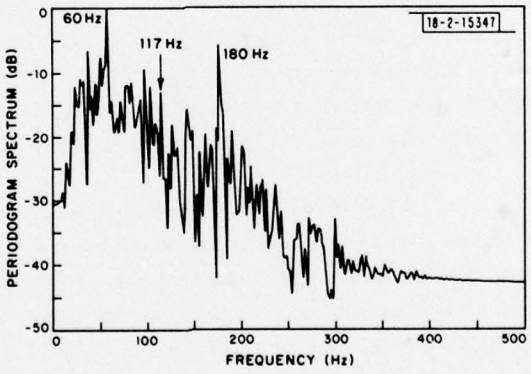


Fig. V-8. Periodogram spectrum from the sum of the U-8 and A-7 data for which Figs. V-6 and V-7 are the spectra.

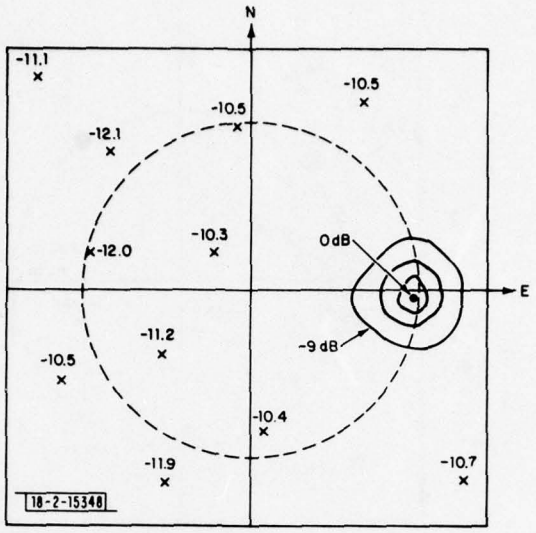


Fig. V-9. Results of wavenumber analysis of combined (added) U-8 and A-7 data. Analysis is at a frequency of 105 Hz where A-7 noise power dominates. Spectral matrix estimated using a single dyad obtained by discrete Fourier transformation of 1024 samples of data over a one-second interval. Power contours are at 3-dB intervals. The x's indicate small local maxima that do not correspond to known sources. Dashed circle is locus of points corresponding to sound traversing the array horizontally at the speed of sound. The A-7 acoustic range is approximately 8.6 km and the U-8 acoustic range is approximately 6.7 km at the time of analysis. Note the clear peak corresponding to the A-7.

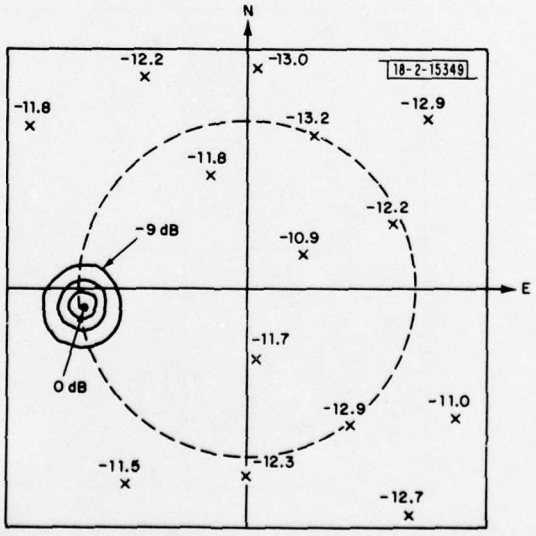


Fig. V-10. Results of wavenumber analysis similar to Fig. V-9 except that the analysis is at 117 Hz where the U-8 noise power dominates. Note the clear peak corresponding to the U-8.

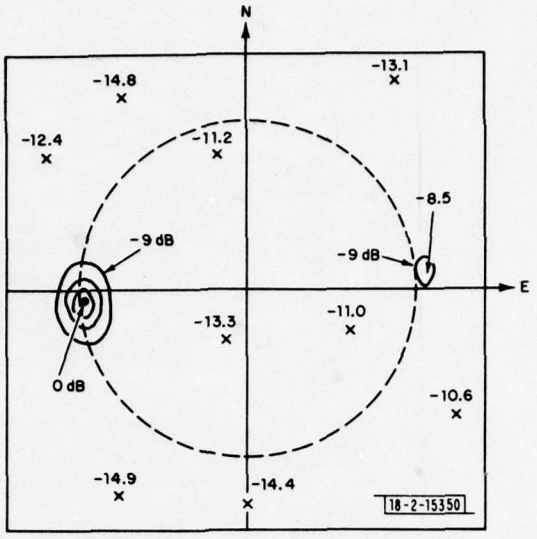


Fig. V-11. Results of wavenumber analysis similar to Fig. V-10 except that five dyads (at 117 Hz) are averaged to obtain spectral matrixes. Dyads obtained using 50-percent overlapping-data intervals so total data interval was 2.5 sec. Note appearance of U-7 peak although it is just barely above background noise peaks.

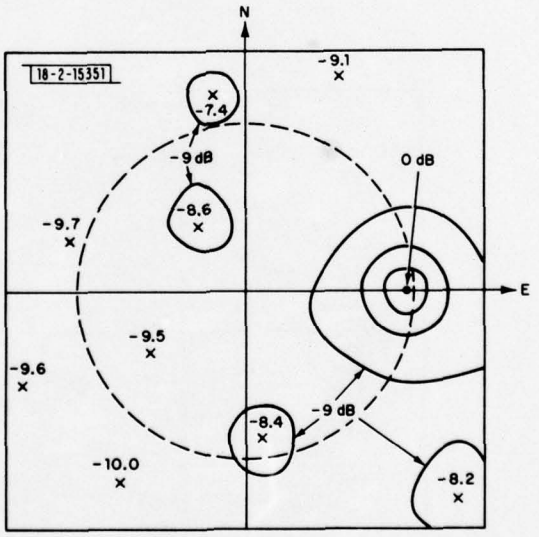


Fig. V-12. Results of maximum likelihood wavenumber analysis of A-7 jet noise. Acoustic range at analysis time is approximately 8.6 km and frequency of analysis is 100 Hz. Single-dyad spectral-matrix estimate used. Jet noise peak to the east is clearly seen.

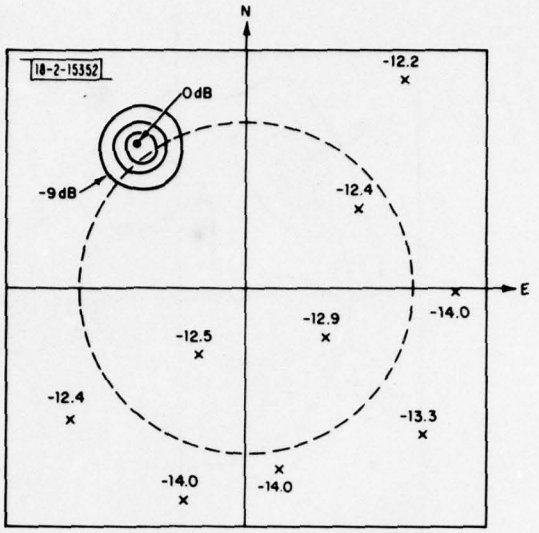


Fig. V-13. Results of maximum likelihood wavenumber analysis at 100 Hz. There was a U-8 approximately 6.7-km west of the array but it produced negligible noise at 100 Hz. The clear peak to the northwest corresponds to a nearby electric power generator. A single-dyad spectral-matrix estimate is used.

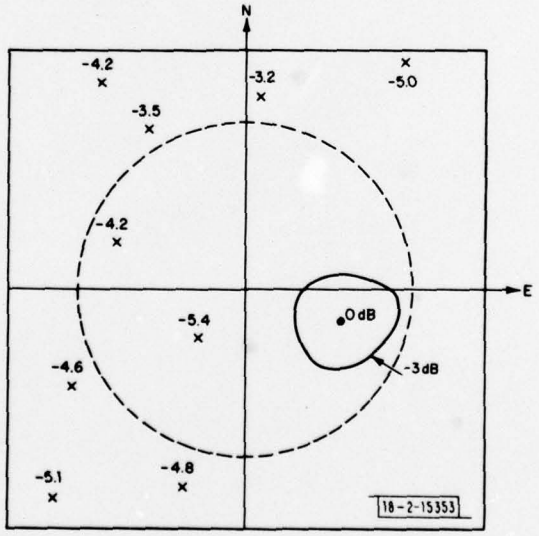


Fig. V-14. Results of maximum likelihood wavenumber analysis of the sum of the A-7 data used for Fig. V-12 and the generator noise data used for Fig. V-13. A single-dyad spectral estimate is used at 1 Hz. The A-7 and generator power levels at 100 Hz are about equal. No clear peak obtained for either known source.

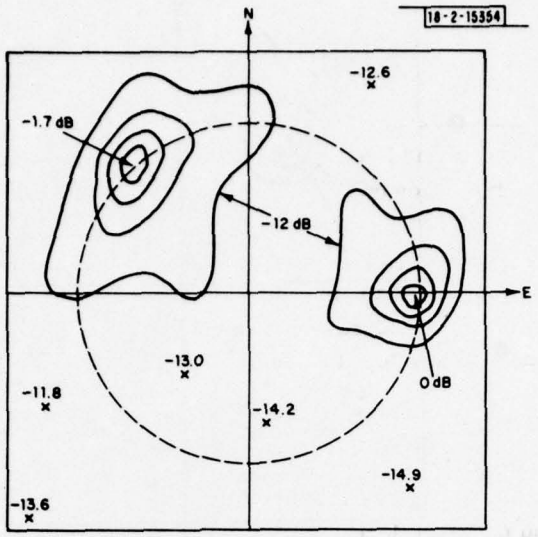


Fig. V-15. Results of maximum likelihood wavenumber analysis of same summed time series used for Fig. V-14. All conditions as for Fig. V-14 except that an 8-dyad estimate of the spectral matrix was used. Both sources known to be in the data have been clearly resolved.

In conclusion, processing the time-series data by averaging dyad-spectral-covariance estimates over time, and applying maximum likelihood, high-resolution, wavenumber processing continues to be very promising. Also, the model that has been developed appears to reasonably represent expected performance of an acoustic node in a simple multiple-target environment.

VI. EXPERIMENTAL FACILITIES

One goal of the DSN project for FY 79 is to record and analyze more acoustic array data. The acoustic data that were recorded in the Ft. Huachuca experiments were recorded with an analog recording system and later digitized and analyzed. A sophisticated digital data acquisition system can record the analog data with greater precision over the full dynamic range, allowing potentially more powerful signal processing. Digital signal processing is of significant interest since a large effort will be spent on investigating whether acoustic arrays can detect targets in the presence of high levels of ambient noise – population, automotive, etc. The purpose of this Section is to discuss design issues and present preliminary specifications for a digital data acquisition system to be developed and used in the coming year.

A. ACOUSTIC DATA ACQUISITION ISSUES

The sound sources of interest, i.e., aircraft, in the DSN environment emit sounds with sound pressure levels as high as 120 to 150 dB in a 1-Hz bandwidth when measured at a distance of 1 m. (All sound pressure levels are measured with reference to the threshold of human hearing, 20 micropascals, where 1 pascal represents a pressure of a force of 1 newton applied over an area of 1 m^2.) It is assumed that in the DSN environment, the sound will be measured in arrays that are between 100 m and 30 km from the source. Assuming attenuation due to spherical spreading and the atmosphere, acoustic signals in the 5- to 500-Hz frequency band of interest at the acoustic arrays will have sound pressure levels between 10 and 110 dB – a dynamic range of 100 dB. Ambient noise levels at these frequencies due to the environment range from 20 to 70 dB.

The high-quality microphones that will be used as acoustic transducers in the DSN sensor, typically, have sensitivities of −26 dB referenced to 1 V per pascal or 50 mV per pascal. Thus, 0- and 100-dB acoustic signals produce output voltages with peak-to-peak values of 1 μV and 100 mV, respectively. The frequency response of such microphones is flat from approximately 4 Hz to 10 kHz. Given the low level of outputs of the microphones, the preamplifier and anti-aliasing filter must be designed specifically with low-noise components.

An A/D system must have a dynamic range that covers at least 100 dB, which is equivalent to 16 bits plus a sign bit. Even the lowest level signals must be represented with significant resolution – at least 5 or 6 bits. The dynamic range combined with the resolution that must be maintained for all levels of signal would require a linear A/D converter with greater than 20 bits of resolution. Considering the current state of technology, a 20-bit converter with any useful degree of temperature stability is essentially impossible to realize. Two alternate approaches are being considered to realize the A/D requirements: a floating-point A/D conversion system and a logarithmic A/D converter.

One possible technique used to realize a floating-point linear A/D converter system is to have control logic automatically sense the signal level and set the gain of a programmable-gain amplifier (for example, in binary-related steps of 1, 8, and 64) such that the signal seen by a 14-bit A/D converter is maintained at a relatively large fraction of full scale. In this way, the dynamic range requirement is satisfied since the product of the two ranges, 64×2^{14} is in excess of 100 dB. The resolution requirement is also met since even at those gain-switching points where the A/D receives only one-eighth its full-scale input, the least significant 11 bits still operate. The programmable-gain amplifier can be controlled by logic that includes several

comparators or a microprocessor system in which each waveform to be digitized would be sampled twice: once with unity gain to determine the gain, and the second time with the appropriate gain. Other techniques of realizing a floating-point A/D converter system are currently being investigated.

In a logarithmic converter, each step, or binary increment, is made to vary successively in size to give the best fit to a true logarithmic curve. The resulting step size is a fixed proportion of the input voltage signal. Resolution is not a percentage of full scale, but is a percentage of the current input level. A 15-bit logarithmic converter exceeds the dynamic range and resolution performance of the above mentioned floating-point converter. Special-purpose logic and an inverter are required to sense the polarity of the input signal, invert it if it is negative, and include a sign bit in the result. The actual technique for digitizing the acoustic signals in the DSN system has not yet been chosen.

Recording data with more than a 100-dB dynamic range with only several bits of resolution is routinely done in low-data-rate seismic systems for earthquake monitoring and at rates approaching our requirements in high-frequency seismic systems used in oil and gas exploration. Thus constructing the digital data acquisition system is not a matter of extending technology as it is of using and adapting current technology.

B. DSN EXPERIMENTAL DATA ACQUISITION SYSTEM

In FY 79, acoustic array data will be recorded digitally onto magnetic tape and later analyzed off line. In the data acquisition system of an actual DSN node, the analog front end will present digital data to the nodal processor for real-time processing. In both systems, the analog front end will be essentially the same.

The DSN acoustic array may include up to 10 microphones. The frequencies of interest are between 5 and 500 Hz. To assure accurate digital representations of the signals, each of the 10 channels will be low-pass filtered (the anti-aliasing filters will have a 500-Hz cutoff frequency and a 60-dB-per-octave attenuation rate) and sampled at a 2-kHz rate – a total system throughput of 20 kHz.

Following is a summary description of what we believe will be developed in FY 79 as an appropriate DSN data acquisition facility. The experimental data acquisition system will either contain an analog front end with automatic gain-ranging A/D or with a programmable-gain A/D with a microprocessor to control the gain of the A/D. A minicomputer will be used to control the overall system and to transfer digital data words (16-bit encoded) to the minicomputer which will format the data for recording onto noncontinuously recording (standard record gap) magnetic tape. The minicomputer will also include appropriate time markers and other pertinent information with the data. Two tape drives will be required since one 2400-foot tape reel will hold only approximately 12 to 14 minutes of data and often one-half to one hour of continuous data may be required.

Initial operation of the digital acquisition equipment will depend on commercial 110-V AC power. However, it is expected that battery operation for field experiments may be required in the future. Continuously operating generators cannot be considered because of the large amount of acoustic noise they would generate. Plans will be formulated for future operations using batteries and inverters in conjunction with recharging using either generators or commercial power.

The specifications presented here briefly are rather severe: a 10-channel A/D system with 100 dB of dynamic range, more than 30 dB of resolution over this range, and a 20-kHz system

throughput. It should be emphasized that these specifications define an experimental system that will be used for research. The results of the investigations may indicate that some of the data acquisition system requirements can be relaxed. However, it is important that in studying the capabilities of acoustic sensors, the most complete and most accurate digital representations of acoustic signals be utilized. Thus one is assured that the performance of the detection and location algorithms depend on the characteristics of the acoustic signals and sensors. When these algorithms are more fully developed and their dependence on input data has been analyzed, perhaps, less demanding requirements for a DSN data acquisition system may be specified.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: ESD-TR-78-398	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE *(and Subtitle)*: Distributed Sensor Networks	5. TYPE OF REPORT & PERIOD COVERED: Semiannual Technical Summary, 1 Apr – 30 Sept 1978, no. 3
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): Richard T. Lacoss	8. CONTRACT OR GRANT NUMBER(s): F19628-78-C-0002, ARPA Order-3345
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Lincoln Laboratory, M.I.T., P.O. Box 73, Lexington, MA 02173	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: ARPA Order 3345, Project No. 8D30
11. CONTROLLING OFFICE NAME AND ADDRESS: Defense Advanced Research Projects Agency, 1400 Wilson Boulevard, Arlington, VA 22209	12. REPORT DATE: 30 Sept 1978
	13. NUMBER OF PAGES: 68
14. MONITORING AGENCY NAME & ADDRESS *(if different from Controlling Office)*: Electronic Systems Division, Hanscom AFB, Bedford, MA 01731	15. SECURITY CLASS. *(of this report)*: Unclassified
66p.	15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

None

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

multiple-sensor surveillance system
multisite detection
target surveillance and tracking
2-dimensional search-space cell
acoustic, seismic, radar sensors
low-flying aircraft
packet radio
high-resolution search-algorithms

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

This report summarizes Distributed Sensor Networks research conducted during the period 1 April through 30 September 1978. One specific model for function decomposition and distribution is presented. A top level design of a strawman DSN for detecting and tracking low flying aircraft is described. High resolution acoustic signal processing algorithms for the strawman are specified and sized. Additional features of a multisite target search algorithm are presented. A moderately detailed model for simulation of an acoustic node under various scenarios is developed. New model simulation and experimental data analysis software are described and demonstrated. Technical issues involved with and plans for a digital acoustic data acquisition system are presented.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*