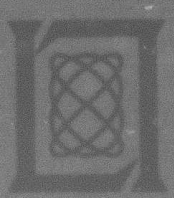
LSD-TR-82-04

Semiannual Technical Summary

Distributed Sensor Networks

31 March 1982

Prepared for the Defense Advanced Research Projects Agency
under Electronic Systems Division Contract F19628-80-C-0002 by



Lincoln Laboratory
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
LEXINGTON, MASSACHUSETTS

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
LINCOLN LABORATORY

DISTRIBUTED SENSOR NETWORKS

SEMIANNUAL TECHNICAL SUMMARY REPORT
TO THE
DEFENSE ADVANCED RESEARCH PROJECTS AGENCY

1 OCTOBER 1961 — 31 MARCH 1962

ISSUED 9 JULY 1962

LEXINGTON MASSACHUSETTS

ABSTRACT

This report desctibes the work performed on the DARPA Distributed Sensor Networks Program at Lincoln Laboratory during the period 1 October 1981 through 31 March 1982.

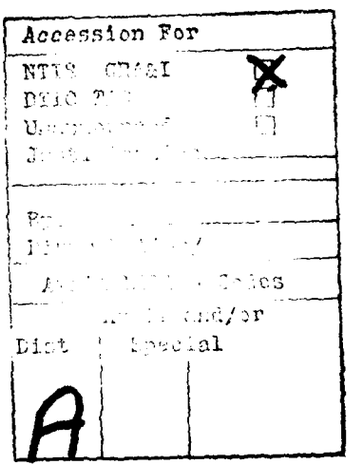


TABLE OF CONTENTS

DISTRIBUTED SENSOR NETWORKS

I. INTRODUCTION AND SUMMARY

The Distributed Sensor Networks (DSN) program is aimed at developing and extending target surveillance and tracking technology in systems that employ multiple spatially distributed sensors and processing resources. Such a system would be comprised of sensors, data bases, and processors distributed throughout an area and interconnected by an appropriate digital-data communication system. Surveillance and tracking of low-flying aircraft have been selected to develop and evaluate DSN concepts in the light of a specific system problem. A DSN test bed that will make use of multiple small acoustic arrays as sensors for low-flying aircraft surveillance is being developed and will be used to test and demonstrate DSN techniques and technology. This Semiannual Technical Summary (SATS) reports results for the period 1 October 1981 through 31 March 1982.

In November of 1981, a data-collection field experiment was conducted using a UH-1 helicopter and four test-bed array nodes. Section II describes this experiment and the azimuth- and position-tracking results obtained. Recent modifications and improvements to the tracking software are described. The algorithms were able to generate location tracks up to 3 km in length for individual pairs of nodes. The azimuth tracks were initiated at ranges up to 5 km. Azimuth tracks were maintained for up to 7 km of aircraft motion. These results were obtained using signal-processing and tracking algorithms that have been developed for initial distributed real-time use in the test bed. Section II also presents a detailed spectral analysis of the UH-1 data. Those results will be used to develop improvements to our signal-processing algorithms.

A DSN workshop was held at Lincoln Laboratory on 6-7 January 1982. We contributed four papers and gave several demonstrations of current test-bed capabilities. The demonstrations were designed to show progress in the areas of data-acquisition, signal-processing, and tracking capabilities. Real-time

signal processing was demonstrated using both prerecorded acoustic data and data read directly from the analog-to-digital conversion system. Real-time azimuth tracking was demonstrated using two nodes equipped with Motorola 68000 microprocessors. Real-time position tracking was demonstrated using data from two and three nodes, but using our PDP-11/70 software development computer rather than nodal hardware. A mobile data-acquisition node, installed and operating in a truck, was demonstrated. Details are given in Sec.III.

As work on the test bed has progressed, we have modified our detailed configuration plans. Section IV presents the improved test-bed configuration plans that have been developed during this reporting period.

Progress in the test-bed hardware is reviewed in Sec.V. Motorola Versamodule systems were first installed in two test-bed nodes and used for the real-time azimuth-tracking demonstrations during the January workshop. Five additional units have been procured and are being used to complete a three-node test-bed configuration capable of distributed real-time operation, with internodal message switching performed by one of the 68000 systems and with the others integrated into the nodes to provide tracking and communication functions. In addition, an acoustically quieted engine/generator has been installed on our mobile node for self-powered operation.

Test-bed software items are covered in Sec.VI. Implementation of the initial real-time signal-processing software was completed. The overall software structure was defined for initial distributed test-bed operation, with message passing as the fundamental communication system. Some elements of the message-passing system and associated software have been completed. The nodal 68000 executive software package has been extended to provide the multitasking and interrupt driven I/O that are required to implement the initial message-passing stage of the test bed.

Section VII summarizes two preliminary test-bed communication system working documents drafted during this reporting period. It also indicates how we will use the Motorola 68000-based Stanford University Network board to develop the Digital Communication Unit.

II. DATA COLLECTION AND ANALYSIS

A. UH-1 HELICOPTER EXPERIMENT

In November 1981, four test-bed nodes were used to record the signals from a UH-1 helicopter as it followed the flight path shown in Fig.II-1. The figure also shows the locations of the recording nodes, designated by the letters F, H, J, and L. The circles on the flight path indicate checkpoints which the pilot called out as he flew over them. The UH-1 flew both west-to-east and east-to-west at about 65 knots. The nodes were arrayed along the length of the flight path to simulate the penetration of a deep barrier. The data were collected to aid with the development and testing of algorithms.

Figures II-2(a) through (d) show the results obtained using our signal-processing and azimuth-tracking algorithms for a west-to-east UH-1 run. The status of real-time implementation is reviewed in the context of demonstrations described in Sec.III.

The figures show azimuth-time-intensity plots of peaks of acoustic power. The front-end signal-processing algorithms that produced these results are described in previous reports.* Only peaks within 10 dB of the local maximum are shown. The curves overlaying the peaks are the output of the azimuth tracker. The incoming UH-1 track is at about 280°, passing to the north of the node and receding toward 90° as it flies away from the node. As shown, all nodes tracked the UH-1 for a substantial time as it approached. However, azimuth track was lost shortly after the closest point of approach (CPA). Nodes J and L acquired the UH-1 at a range of 5 km and tracked it for a distance of approximately 7 km. Node F acquired the target at the start of the data run. Node H experienced difficulties in initial acquisition, probably due to local siting conditions.

---

*Distributed Sensor Networks Semiannual Technical Summaries, Lincoln Laboratory, M.I.T. (30 September 1979), DDC AD-A086800/0, and (30 September 1980), DTIC AD-A103045/1.

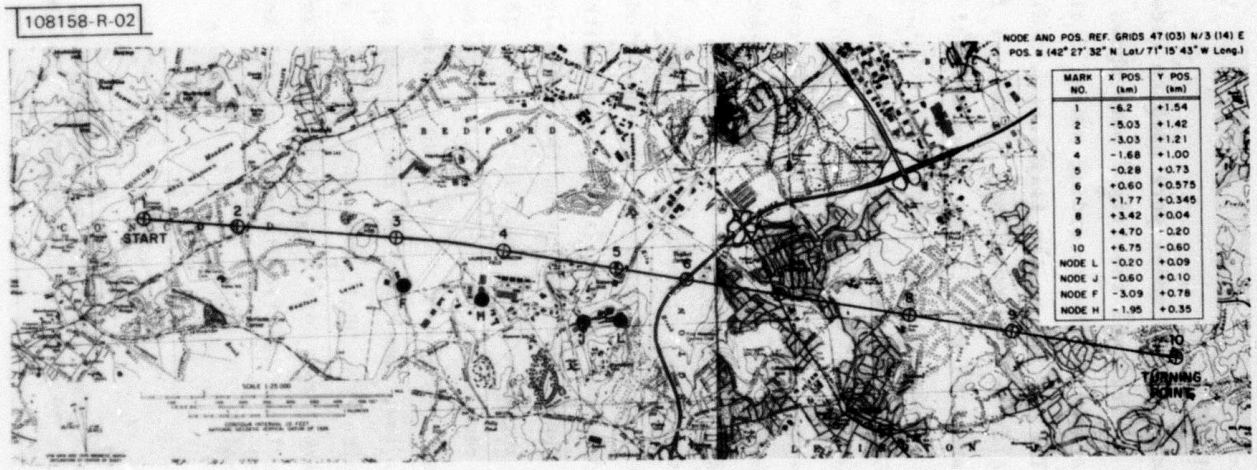


Fig. II-1. Test-bed node locations and UH-1 helicopter flight path for UH-1 experiment.

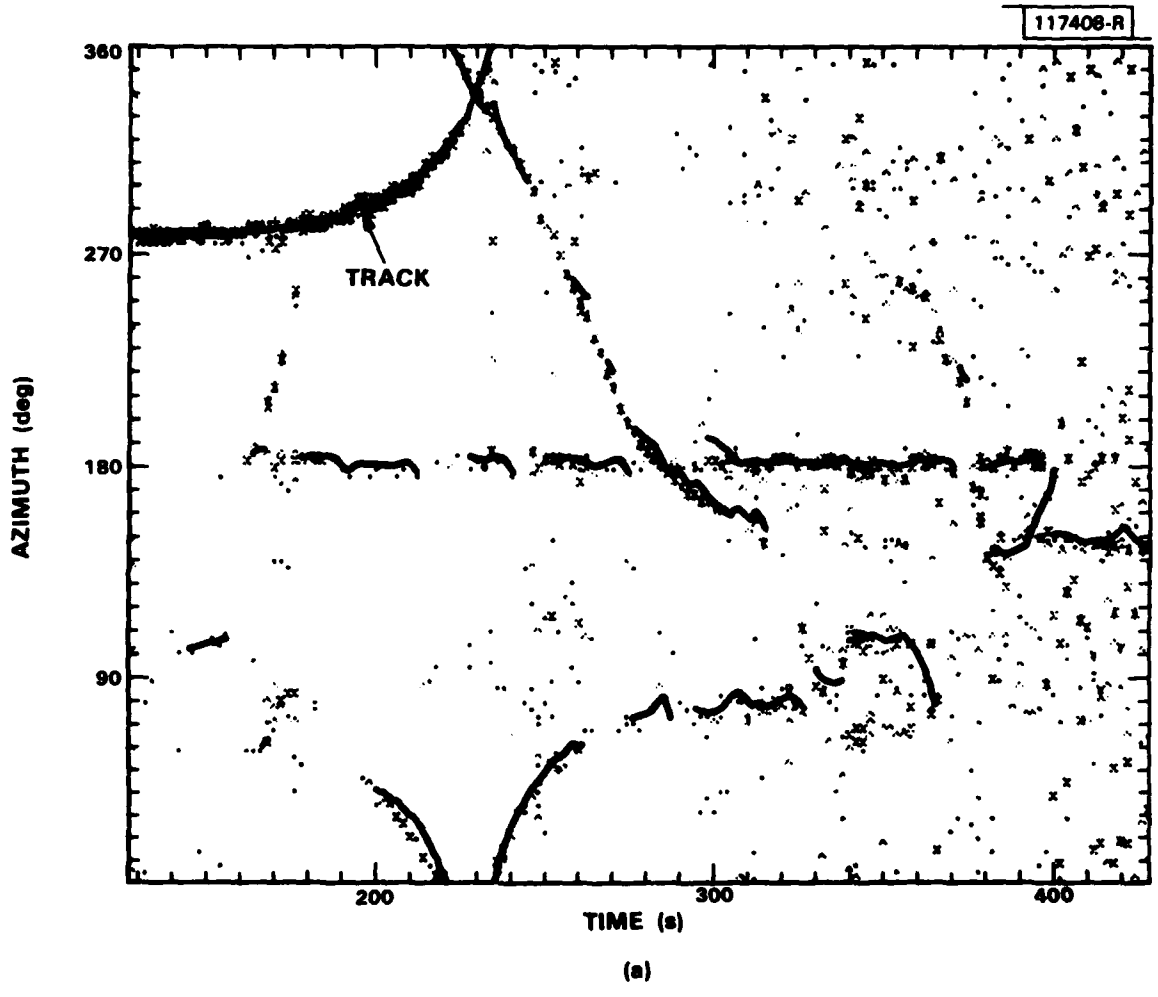


Fig. II-2. Azimuth-time-intensity plots and overlaid azimuth tracks for eastbound UH-1 flight path for (a) node F, (b) node H, (c) node J, and (d) node L.

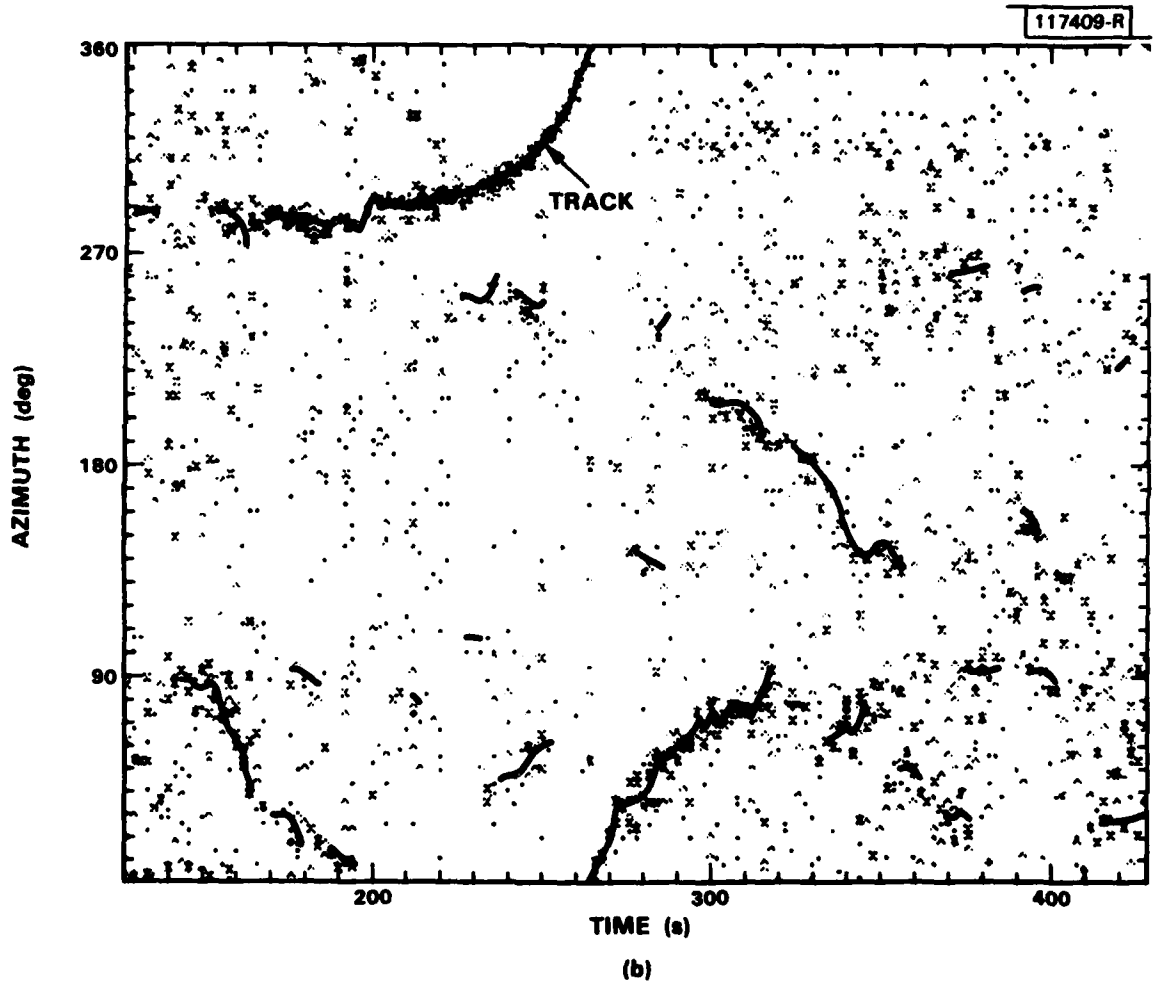
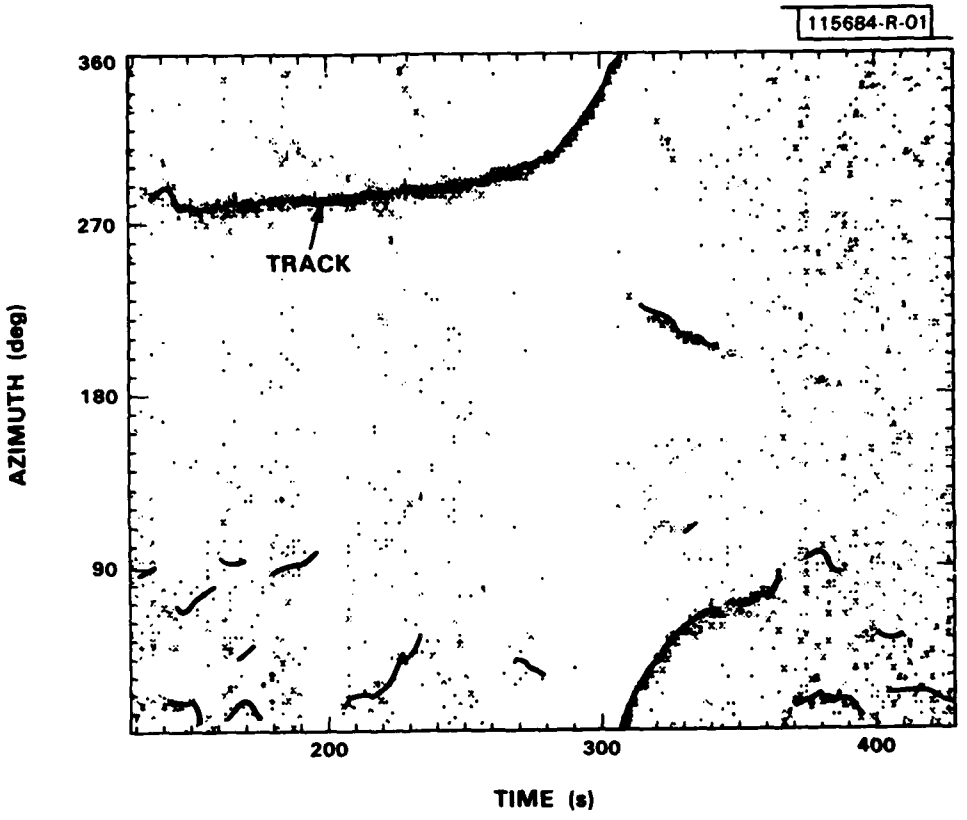


Fig. II-2. Continued.



(c)

Fig. II-2. Continued.

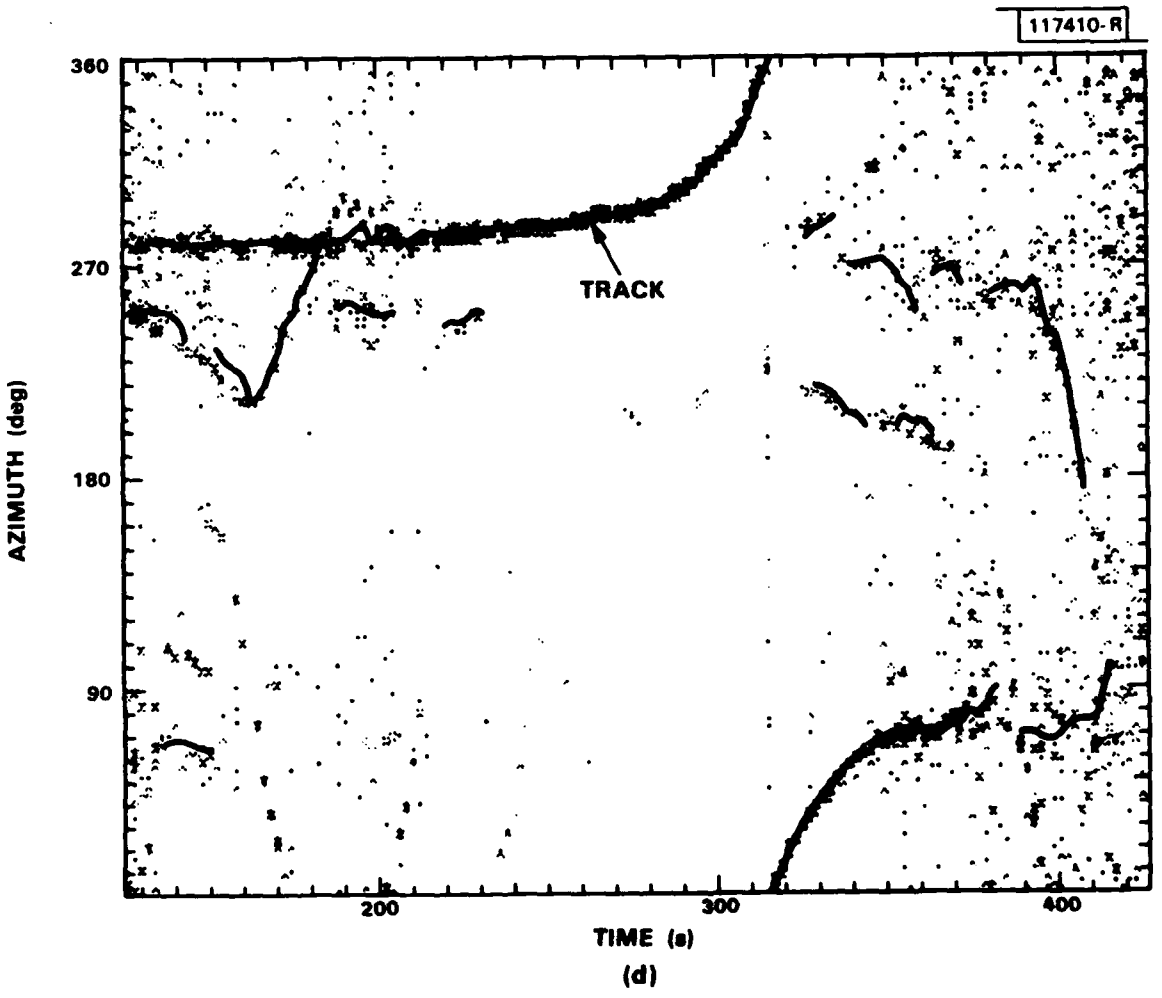


Fig. II-2. Continued.

To obtain these results, spectral analysis was performed with 4-Hz resolution, and spatial analysis was done at the nine largest spectral peaks. Work is now in progress to determine how the signal-processing algorithms could be changed to improve performance. As a first step, we calculated spectragrams for single microphones from all four nodes. Figure II-3 shows one of these spectragrams, which is typical of all the others. The spectragram is a plot of power as a function of frequency and time. The spectra shown were calculated with 1-Hz resolution. Spectra are shown for 1-s intervals of data with 0.5-s overlap between adjacent intervals. Each spectrum is normalized to its peak value. The spectra have been displayed with high gain and thresholding. This causes the light tracks to appear, corresponding to lines in the acoustic spectra.

The many harmonic lines visible before CPA are associated with UH-1 main and tail rotor-blade slap. As CPA is approached and passed, the aspect of the helicopter to the array becomes side-on and from the tail, and the higher-order harmonic lines due to the main rotor degrade. This effect is attributable to the complex dynamics of helicopter-blade rotation. The

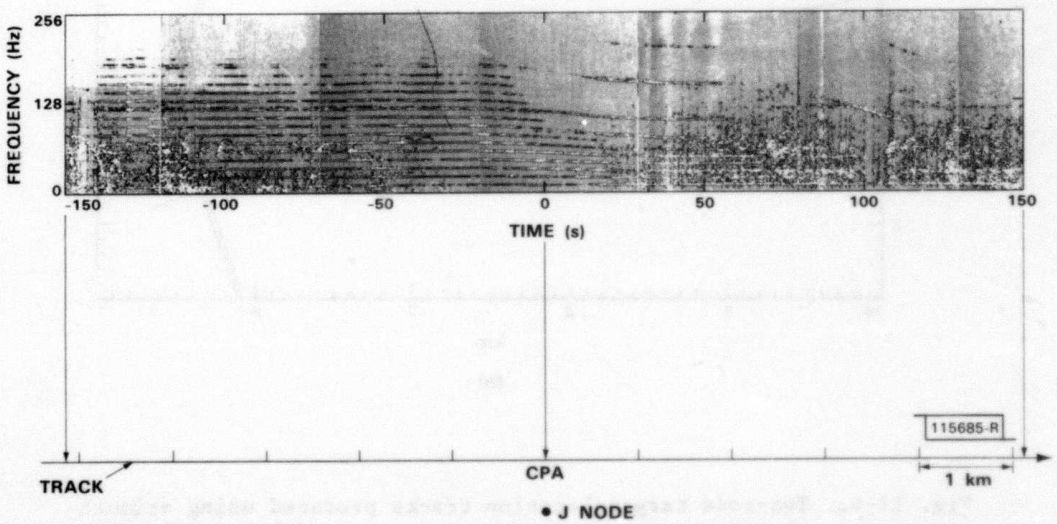


Fig. II-3. Spectragram of UH-1 flyby showing spectral content of received signals as a function of time.

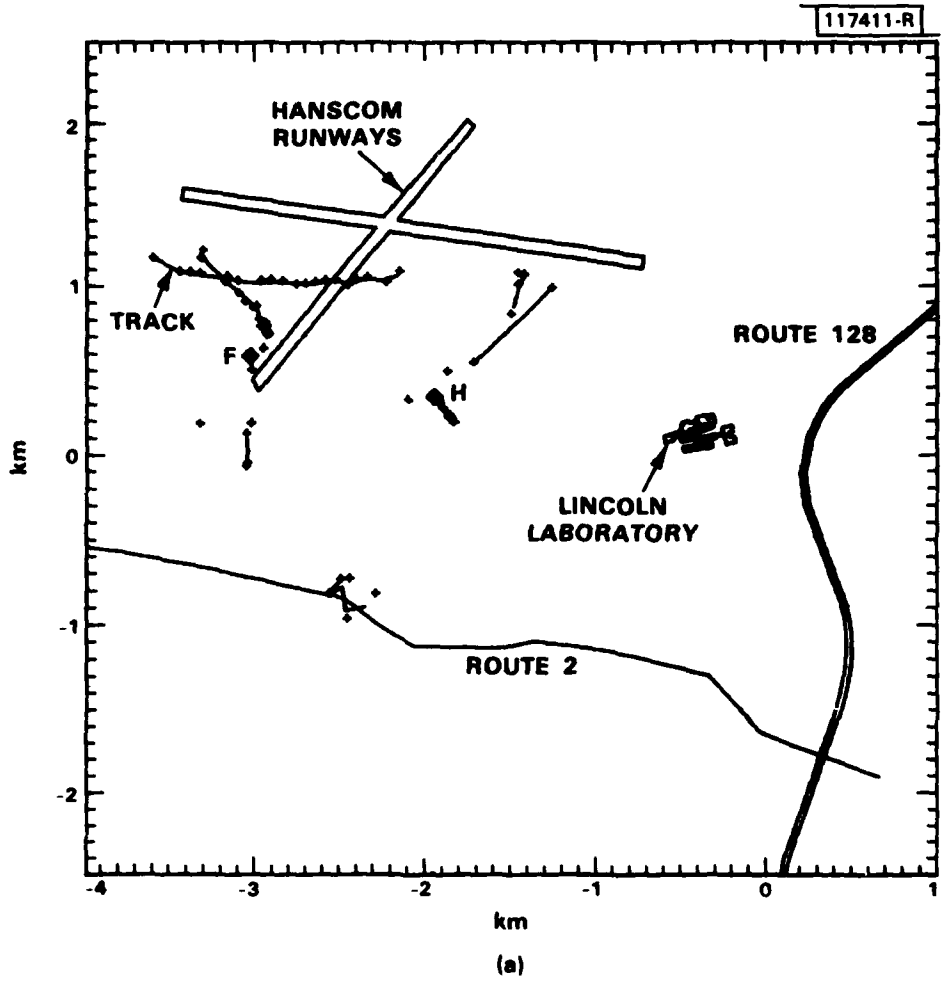


Fig. II-4. Two-node target-location tracks produced using azimuth tracks shown in Fig.II-2 from (a) F and H nodes, (b) H and J nodes, (c) J and L nodes.

result is that the UH-1 signature at the side and tail aspects is only composed of tail rotor harmonics and lower-order main-blade lines. Since tail rotor-blade slap is less powerful and the lower-order main-blade lines are in the region of high background noise, the general impression from the spectragram is that the UH-1 is detectable at larger ranges inbound than outbound. This feature is consistent with our tracking results for these data as discussed above. Closer inspection suggests that the low-order main-blade lines persist for a considerable amount of time. We are now determining how low-frequency noise suppression, increased spectral resolution, and spectral line tracking can be used to improve our current signal-processing capabilities.

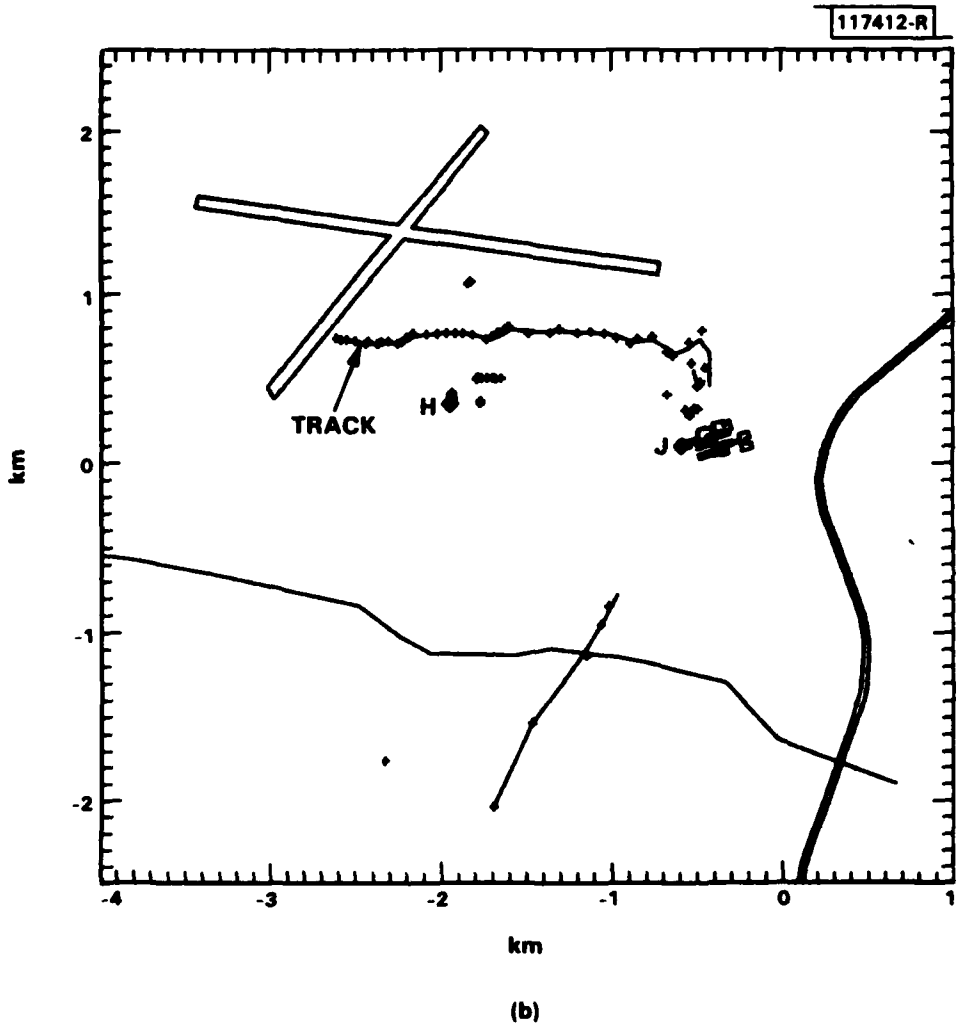
Figures II-4(a) through (c) show our position tracker output for azimuth-track inputs from pairs of nodes. Overlapping segments of the UH-1 flight path were generated for different node pairs. Other track segments, some spurious and some corresponding to other targets in the area, were also generated. Similar results were obtained from the pairs FJ, FL, and HL.

B. IMPROVEMENTS TO TRACKING ALGORITHMS

A number of changes were made to the azimuth- and position-tracking algorithms to obtain the results described in the previous section. These changes and the reasons they were implemented are described here.

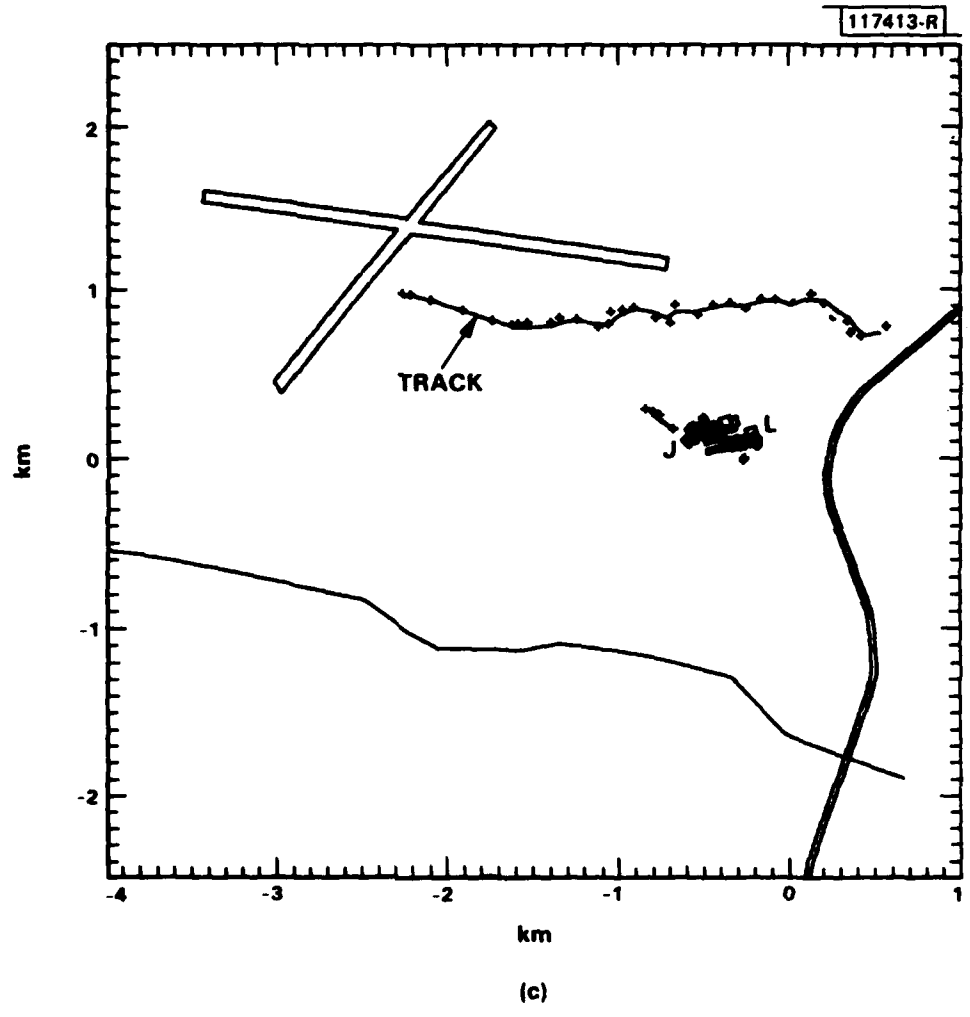
Azimuth-tracking algorithms were modified to favor targets with slowly changing azimuths at the time of track initiation. This improves performance for targets approaching from a distance, while discriminating against local targets that would have more rapidly changing azimuths when first detected. This has proven to be an effective strategy, although there may be difficulties with approaching targets that are detected late or with targets for which track is momentarily lost.

The inaccuracy of the position locations obtained with our two-node location procedure is directly related to the errors in the azimuth tracks. Some tuning of azimuth-tracking filter parameters was done to reduce the effect of new data points on well-established tracks. This resulted in smoother and more accurate azimuth tracks.



(b)

Fig. II-4. Continued.



(c)

Fig. II-4. Continued.

A first-order model is used for azimuth tracking, as this minimizes the amount of CPU time required for filter processing. This model is a good approximation for slowly maneuvering targets except when near the CPA. When the target is passing by the node, its angular acceleration becomes significant and the first-order model is a poor approximation. As a result, the actual target azimuth falls outside the angle gate around the projected target azimuth and the track is dropped due to data points not being associated with the existing track. To overcome this problem, the ratio of the azimuth-track variance to the input-cluster variance was monitored and the effect of the new data points was increased as the ratio increased. Whenever the target is near CPA, or maneuvering, the azimuth-track variance increases due to model errors and the filter relies strongly on new data cluster input values. Away from CPA, the azimuth-track variance is significantly less than the input-cluster variance and a highly filtered and smoothed azimuth-track results.

The next improvement was to provide azimuth-track-variance estimates to the position tracker. This enables the position tracker to estimate the accuracy of each target location obtained. If the error in determining a target location is outside a specified bound, then that point is not used as input to the position tracker. A location is obtained by determining the corresponding arrival angles of sound rays from two nodes and then triangulating. The error in position is determined by computing the position errors that would result due to a one-standard-deviation error in the azimuth rays. If the worst-case error deviation in position is outside some bound, then the point is not used. This takes into account the geometry of the target relative to the pair of nodes in determining the effect of errors. It should be noted that this procedure causes the lengths of location tracks in Figs.II-4(a) through (c) to be shorter than one might expect from the length of the contributing azimuth tracks. This would not be the case for other geometries.

The final major improvement is related to the times associated with target locations used as input to the position tracker. In a conventional

tracking situation, the tracking filters are updated at a regular time interval. When we determine target locations by triangulation from a pair of nodes, we also compute the time at which the target was at that location. This time has errors induced by azimuth-track errors. It is beneficial to produce a smoothed estimate of this time, otherwise situations occur in which successive target locations appear to move backwards in time. To smooth the time estimate, we make use of the knowledge that the time intervals between successive locations will be slowly changing. As such, we use the computed times to produce an exponentially smoothed estimate of the current update interval and use this, in turn, to compute the times which are input to the tracking filters. This results in reasonable estimates for the rate of change of target position with time and, hence, in effective operation of the position filters to reduce errors in the target tracks.

C. TRACKING-ACCURACY EVALUATION

As part of our data-analysis activity, we are developing computer programs to compare target tracks obtained acoustically with those obtained from a radar system located at Lincoln Laboratory. The radar system is the DABS* developed for air-traffic-control purposes for the FAA. It produces data tapes containing the positions of all transponder-equipped aircraft in the vicinity of Hanscom AFB, where we conduct our experiments. In particular, the tape includes tracks of our primary target aircraft.

We have written a computer program to extract from the DABS tapes the successive locations of our target aircraft and the times associated with these locations. This program plots the target track and produces a corresponding data file. A track-comparison program is now being written that will determine peak and rms track errors and will plot instantaneous position errors.

*V.A. Orlando and P.R. Drouilhet, "Discrete Address Beacon Systems Functional Description," Project Report ATC-42, Rev. A, Lincoln Laboratory, M.I.T. (11 April 1980), DTIC AD-A085169/1.

III. DSN WORKSHOP AND DEMONSTRATIONS

Lincoln Laboratory organized and hosted a DARPA DSN contractors workshop which was held on 6-7 January 1982. As part of our participation, we prepared and presented papers and demonstrations as summarized below. We have also organized the written contributions from the participants into a workshop proceedings that is now being reproduced for distribution to all participants.

We presented four technical papers,[1-4] written versions of which have been prepared and are included in the workshop proceedings.

Reference 1 is an overview of the DSN program at Lincoln Laboratory. The problem of battlefield surveillance using seismic and acoustic sensors as a possible DSN application area is identified in Ref.2.

The other two papers presented more detail on topics covered in the overview. In Ref.3, the acoustic-signal processing and tracking algorithms used in the test bed were addressed. These include the Maximum Likelihood Method of array processing to provide measurements of sound levels as a function of direction and frequency, clustering and data-association algorithms, and azimuth and location tracking using alpha-beta filters to update tracks. Details of the test-bed communication system were covered in Ref.4.

As part of the workshop, a tour of the DSN test bed was organized. Five demonstrations were given:

(1) Signal processing and peak detection from recorded data.
(2) Combined azimuth and position tracking by two nodes.
(3) Position tracking with input data from three nodes.
(4) Signal processing and peak detection using live data.
(5) Data acquisition using a truck-mounted mobile node.

The input data for demonstrations (1) and (2) were prerecorded microphone data from the UH-1 experiment described in Sec.II. Demonstration (3) also involved UH-1 data, but the input was recorded peak data that had been

previously produced by applying signal-processing algorithms to the microphone data from each array. Demonstrations (4) and (5) involved ambient noise at the time of the demonstrations, as well as signals generated by loudspeakers which are used for calibration purposes. The demonstrations were designed to show the status of our real-time acquisition, processing, and tracking capabilities.

Figure III-1 shows the data-flow paths between those parts of the test bed that were directly involved in demonstrations (1) through (4). The dashed line encloses those elements located at node J. There was another identical set of equipment at node L, except for the graphics display. The two nodes are each connected to our PDP-11/70 computer over 9600-baud land lines. The graphics display on node J was used to show azimuth-time-intensity plots and azimuth tracks during demonstrations. The PDP-11/70

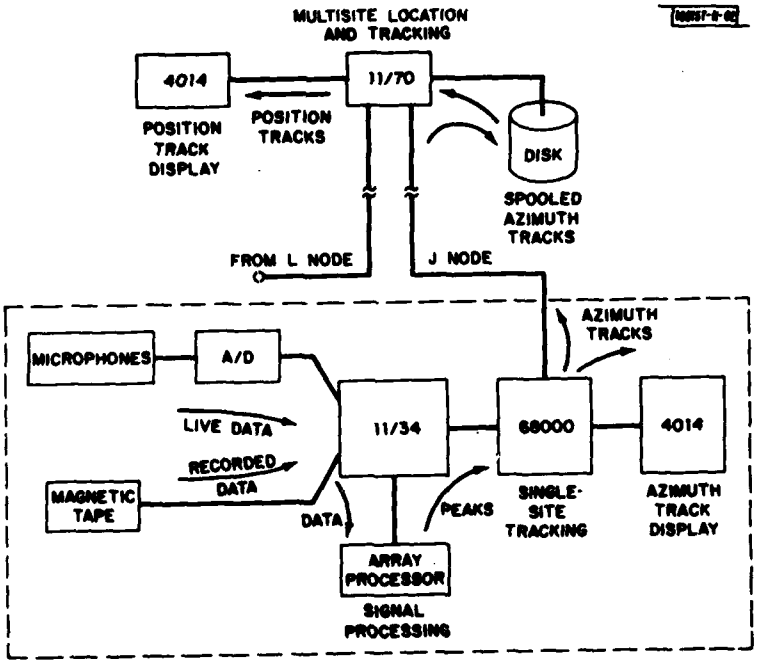


Fig. III-1. Test-bed elements used for real-time signal-processing and tracking demonstrations.

display was used for position tracks. Both displays were located in the same room as the J-node electronic hardware and could be viewed simultaneously.

In the first demonstration, digitized microphone data, recorded at J node during the UH-1 experiment, were input to the signal-processing algorithms running in the array processor. The data were processed at a real-time rate. The resultant peaks were transmitted to the tracking processor which displayed azimuth-time-intensity plots on its graphics display terminal. This demonstrated our ability to perform signal processing and peak detection in real time.

In the second demonstration, signal processing was again performed in real time simultaneously in J and L nodes using prerecorded data. The resultant peaks were processed by the respective azimuth-tracking processors. The J-node processor also plotted the azimuth tracks on the graphics display terminal. Both nodes sent their azimuth tracks to the 11/70 where they were converted into target-position tracks which, in turn, were displayed on the graphics terminal attached to the 11/70. This demonstrated our ability to perform acoustic target tracking in a real-time environment with two nodes.

For the third demonstration, the position-tracking algorithm running on the 11/70 produced target-position tracks using precomputed azimuth-track files from a set of three nodes as input. This demonstrated that the position-tracking algorithm was able to process data from a variable number of nodes and that the processing load is sufficiently small so the tracking function can be accommodated by a 68K nodal processor.

For the fourth demonstration, J node was operated with live input from its microphone array. The microphone signals were processed in real time and the peaks of sound were displayed on an azimuth-time-intensity plot by the tracking processor. Each microphone array is equipped with a loudspeaker for calibration. When this speaker was turned on, the calibration peaks became clearly visible on the plot. This demonstrated our ability to perform the signal processing and peak selection in real time from live data. Rain and a low ceiling prevented aircraft from flying low over the airfield, so that real-time azimuth tracking of live targets was not possible although we have

accomplished this on other days with targets of opportunity around the airfield.

The fifth demonstration was the mobile node operating in data-acquisition mode. Visitors were shown this node in operation, collecting data from a microphone array located near the vehicle and producing data displays on the operator's terminal.

REFERENCES

1. R.T. Lacoss, "Overview of the Distributed Sensor Networks Program at Lincoln Laboratory," presented at the DSN contractors workshop held at Lincoln Laboratory, 6-7 January 1982.

2. T.E. Landers, "Passive Battlefield Surveillance: A DSN Application," loc. cit.

3. P.E. Green, "Distributed Acoustic Surveillance and Tracking," loc. cit.

4. D.P. White, "DSN Communication System Development," loc. cit.

IV. REVISED TEST-BED CONFIGURATION PLANS

As work on the test bed has progressed, our plans have evolved to incorporate new ideas and also to take account of equipment availability. This section describes present plans for the next four stages of test-bed evolution, namely:

(a) A three-node configuration which will be used to perform real-time DSN experiments with all processing functions accomplished in the nodes.
(b) A six-node modification and expansion of the three-node configuration.
(c) A six-node configuration with communications by means of the nodal Radio Units.
(d) The development of an advanced nodal architecture which will provide sufficient processing power to accommodate processor-intensive distributed algorithms.

Stages (a) and (b) will use land lines from the nodes to the Experiment Control Computer (ECC) for internodal communications. Stage (c) will use the Radio Units, and stage (d) will take place in parallel with the other stages.

All stages will use message-passing communications. This is a major and essential advance on our previous experiments where we have used character-by-character communications on fixed physical circuits. Initial plans call for static message-routing tables to be compiled into the message-handling software in the test bed.

A. THREE-NODE CONFIGURATION

We have decided to conduct our initial distributed operation experiments using three nodes, with the ECC acting as a communications emulator as shown

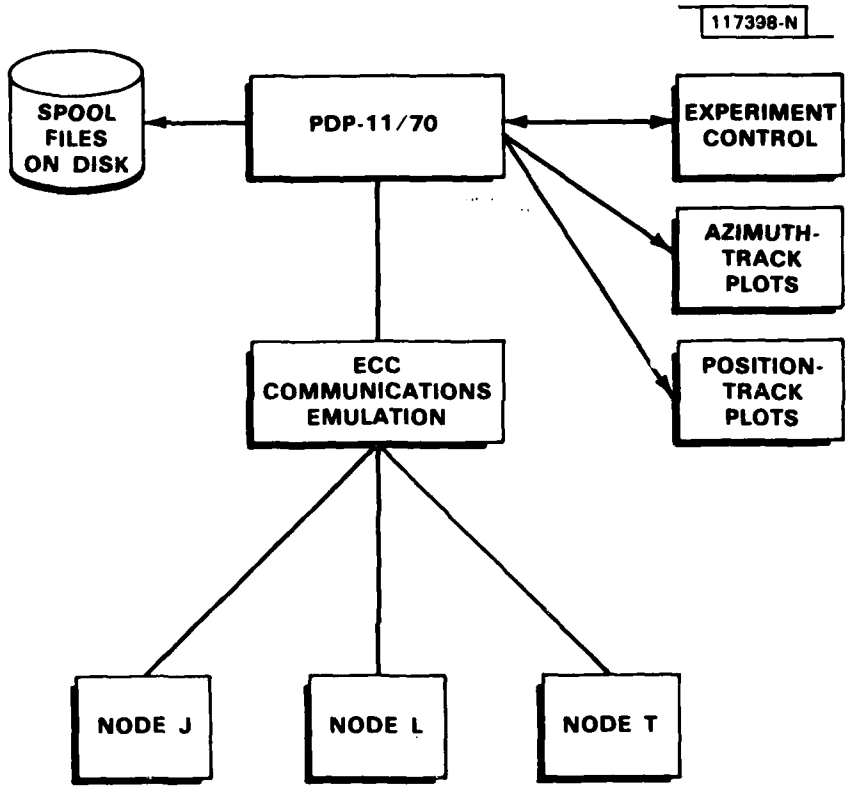


Fig. IV-1. Three-node configuration for completely distributed real-time experimental DSN operation.

in Fig.IV-1. Also shown in the figure is the PDP-11/70 computer which will serve as the user interface for monitoring and control of the experiment.

Each of the nodes will sense the environment and generate its own azimuth tracks. These will be sent as ASCII messages to the ECC which will relay the messages to the other nodes. Each node will use its own azimuth tracks and those it receives to generate position tracks based on the current pairwise location algorithm. These tracks will be sent to the ECC for display.

DSN experimenters will operate from an 11/70 terminal from which they can download the ECC and the nodes, set up experiment parameters, and start and stop experiments. They will also be able to obtain, store, and display azimuth or position tracks passing through the ECC. The ECC is being implemented using Motorola 68000 Versamodules.

The configuration for each of the three nodes is shown in Fig.IV-2. Each node will contain two Motorola 68000 Versamodule systems and a signal-processing subsystem. One 68000 will be used for azimuth tracking and one

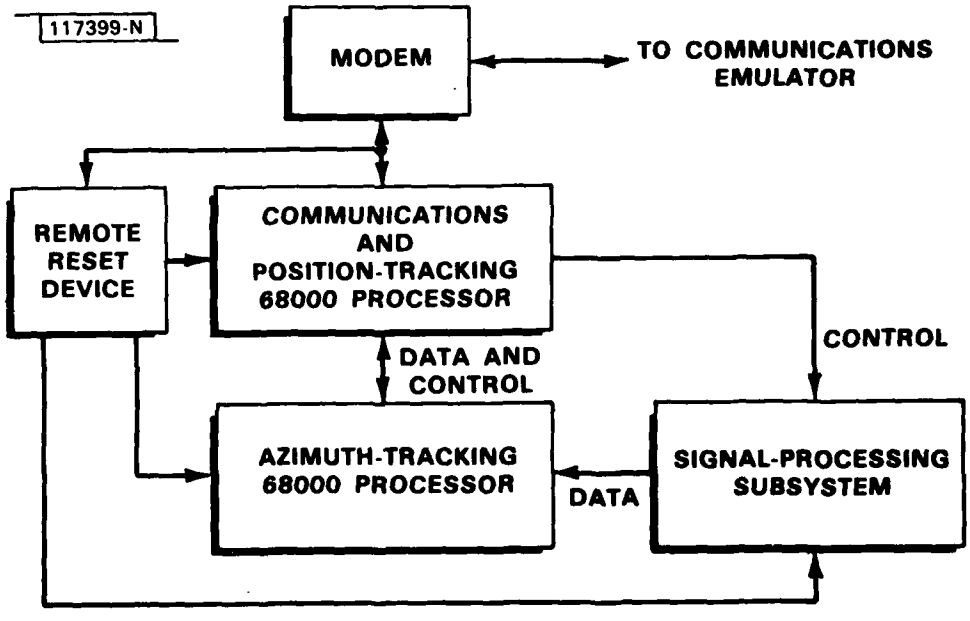


Fig. IV-2. Nodal configuration for three-node real-time experiments.

will be used for both position tracking and internodal message communications. Communications between the processors within a node will be by means of 9600-baud serial links except for the data link from the signal-processing subsystem to the tracking processor, which uses a parallel data path. A remote reset device that monitors the line from the ECC is included in each node. This device resets the individual processors when special codes are received from the ECC. The signal-processing subsystem consists of a PDP-11/34, an A/D converter to which the microphone array is connected, an array processor which is used for signal processing, a 9-track tape drive, and a satellite time clock.

The nodes that will be used in this configuration are J, L, and T. J and L are fixed nodes, and node T will be located in a truck and will serve as our first mobile node. In all cases, land lines and modems will be used for communications between the nodes and the ECC and from the ECC to the 11/70.

The initial three-node configuration will be available for distributed experiments starting during the next quarter.

B. EXPANSION TO SIX-NODE DISTRIBUTED OPERATION

In expanding from a three- to a six-node distributed capability, during the next two quarters we will introduce Stanford University Network (SUN) workstation processor boards into the test bed. We have evaluated this board during this report period, and it appears to be better suited than the Motorola Versamodule for present and future test-bed applications. Its small size, availability from multiple sources (Pacific Microcomputers, Inc. and Forward Technology, Inc.), and its ability to interface to the proposed IEEE-796 bus initially popularized by Intel as the Multibus make it particularly attractive for our advanced node architecture. The Multibus is a mature technology with many sources of memory boards and I/O interfaces.

The six-node configuration shown in Fig.IV-3 will utilize two SUN processor boards on a Multibus as separate communication and position-tracking processors. The extra 512-kbyte memory on the Multibus is available on a single off-the-shelf card and will serve as a buffer for interprocessor communications. Connections to the ECC, to the Versamodule that will continue to

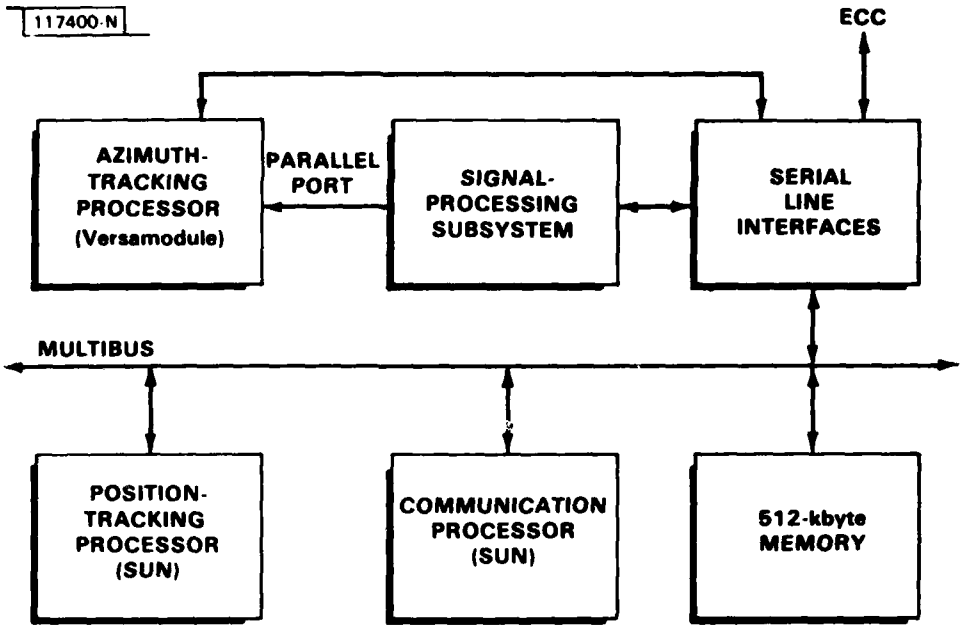


Fig. IV-3. Nodal configuration for real-time six-node operation without radios.

perform azimuth tracking, and to the signal-processing subsystem will use an off-the-shelf serial interface card. As an intermediate stage, some nodes will operate with a single SUN processor for communication and tracking. The use of separate processors anticipates the addition of radios to the test bed.

Software conversion required to make use of the SUN processor boards will be limited to a few low-level library routines that deal with I/O devices and interrupt handling.

C. SIX-NODE CONFIGURATION WITH RADIO UNITS

The next stage of development is depicted in Fig.IV-4. There are two primary changes. First, a DMA interface will be added to connect a Radio Unit to the Multibus. Second, the processor performing the azimuth-tracking function will be changed to a SUN processor on the same Multibus as the

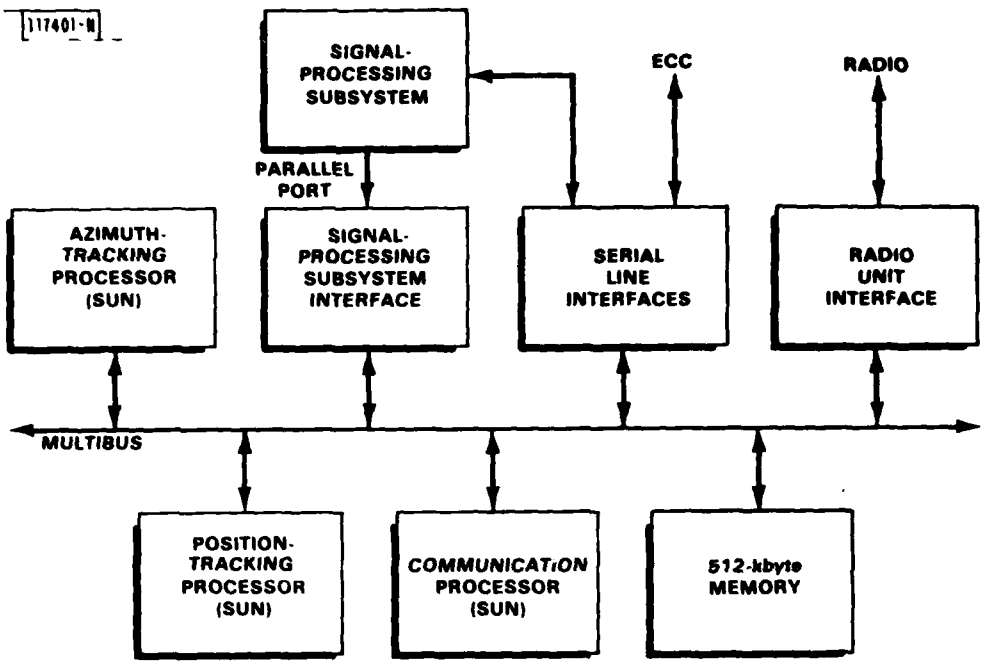


Fig. IV-4. Nodal configuration for operation with radios.

communication and position-tracking processors. With all processors now located on the same Multibus, all local interprocessor communication will be accomplished through the shared 512-kbyte memory. This regular architecture will allow us to easily add processors to the node as needed.

The Radio Unit interface, one SUN processor, part of the memory common board, and serial I/O ports constitute the Digital Communications Unit for each test-bed node. The common memory will include the DMA buffers required for interacting with the Radio Unit. The Radio Unit interface cards should be completed within six months for integration with the Radio Units in early FY 83. Conversion of the azimuth-tracking processor to the SUN processor will also take place during this period.

Many of the operational objectives we have previously discussed* for advanced node configurations are accommodated by the Multibus-SUN processor

*Distributed Sensor Networks Semiannual Technical Summary, Lincoln Laboratory, M.I.T. (31 March 1981), DTIC AD-A108275/9.

configuration. The architecture is regular and expandable, as long as loading on and contention for the Multibus do not become too excessive. Further examination of the enhancements which might be made to the above six-node configuration will be deferred until we gain experience with the shared-bus architecture. If the shared-bus architecture proves to be adequate for interprocessor communications and there are no major operational or maintenance difficulties, then the basic DSN nodal architecture will probably remain unchanged for the foreseeable future.

V. TEST-BED HARDWARE

All three custom vehicles (heavy-duty trucks with special body housings) that will be used as mobile nodes have now been delivered. As previously reported, installation of the data-acquisition and signal-processing hardware in the first of these vehicles was initiated in September 1981. In November 1981, this unit was connected to an acoustic array set up on a grassed area next to Hanscom hangar Bldg. 1718, and was used as one of the nodes in a four-node tracking experiment with a UH-1 helicopter. An FPS-120B array processor has since been added, and this node is now capable of performing real-time signal-processing functions.

With the exception of one PDP-11/34 computer and an FPS array processor which are on order, all signal-processing subsystem hardware for the other two mobile nodes is now on hand. Installation is being deferred while three existing nodes (L, J, and T) are being upgraded to support distributed real-time target-acquisition and tracking experiments.

An acoustically quieted engine/generator power system (E/G system), purchased from J & A Enterprises of Swampscott, Massachusetts was delivered to Lincoln Laboratory in January 1982. The E/G system was installed on the rear platform of mobile unit 1, and acceptance tests of the power system were conducted successfully on 2 February 1982. A photograph of the installation is shown in Fig. V-1. The overall E/G system consists of an E/G set that was modified to obtain quieter operation and a sound-deadening enclosure that is the dark box at the rear of the truck.

The E/G system meets all its performance specifications. We are now preparing to use it as the power source for a mobile node to measure performance under quiet ambient conditions. These tests will confirm the acceptability of the E/G system for operational use, and, subsequent to this evaluation, two more systems will be procured for the second and third mobile nodes.

As noted above, we have been concentrating on completion of a three-node test-bed configuration that will allow the demonstration of distributed performance with all signal processing and tracking performed in the nodes.

Fig. V-1. Mobile node with acoustically quieted engine/generator mounted at rear.

To achieve this upgrade, we are installing Motorola 68000 Versamodule microcomputer systems containing a single-board 68000-based microcomputer, 128-kbyte extended memory board, and a chassis with integrated power supply in three DSN nodes.

The Versamodule system which will function as the ECC for communications emulation and experiment monitoring has been attached to the PDP-11/70 as shown in Fig. IV-1. It is fully tested, and the communication links from the ECC to the remote DSN nodes are being installed at this time. Once installation is complete, and the communications software finished, the ECC will emulate a radio network in which various communications schemes can be tested.

Two multichannel communications modules (mccm) have been purchased and installed in the ECC system. Each mccm is a board-level Motorola product containing two full-duplex serial channels. These will be used to connect

the test-bed nodes to the ECC. The mccm will format serial data, and transfer that data using direct memory access to the ECC. This will provide very efficient handling of messages by the ECC, and increase throughput of message transfers within the network.

VI. TEST-BED SOFTWARE

A. REAL-TIME ACOUSTIC-SIGNAL PROCESSING

An acoustic-signal-processing program that operates in real time in the test-bed nodes has been completed. It processes 2 s of digitized microphone data from the 9-element test-bed arrays to produce a list of power peaks in frequency, azimuth, and elevation. The program computes the power estimates for a variable set of 8 frequencies, 120 fixed azimuths, and 8 fixed elevations. For each analysis frequency, power estimates are obtained using the high-resolution Maximum-Likelihood Method of array processing. For each frequency, the power values in azimuth-elevation space are searched to select peaks that are passed to azimuth-tracking software. The frequencies at which the spatial processing is done are selected as the frequencies corresponding to peaks in the power spectral density of single acoustic channels in the preceding 2 s. This software was demonstrated in January 1982 as described in Sec.III.

The software requires 2.43 s to completely process a 2-s block of data. Therefore, for real-time operation it does not process contiguous 2-s blocks but skips about 0.43 s between blocks. Of the 2.43 s, 2.13 s are spent waiting for the signal-processor programs to execute in the nodal FPS-120B array processor; 0.23 s is spent on PDP-11/34 signal-processing management programs (the greater part of this copying 74 kbytes of data and 16 kbytes of program into the signal processor); and 0.07 s is spent managing input/output. If the array processor could be coded to run at 50 percent of its maximum rate, then we estimate that the array processor time would be reduced to only 0.75 s.

The fact that the current implementation skips over about 18 percent of the incoming data is not viewed as a serious problem. We are dealing with relatively slow targets in the test bed, with correspondingly modest angular rates of change even at their closest point of approach. If necessary, we can recode part of the array processor or modify some of the processing

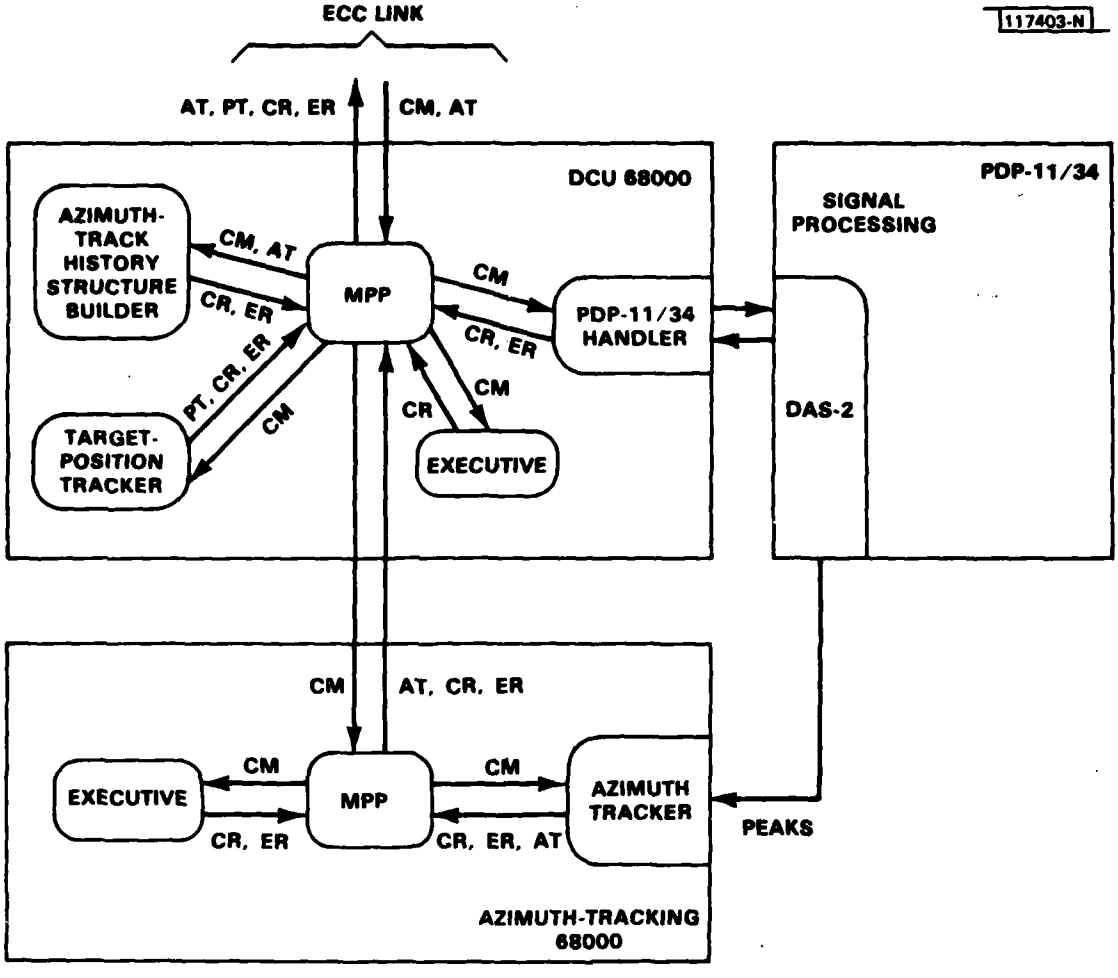


Fig. VI-1. Message flow and tasks in a node.

LEGEND:

MPP - Message-Passing Package
CM - Command message
CR - Reply to a command message
AT - Azimuth-track message
PT - Position-track message
ER - Error message

parameters to avoid skipping some of the data. Additional work on the real-time signal processing will be directed toward algorithm improvement rather than nonessential speed increases.

One possible improvement area is frequency resolution. Due largely to memory limitations in the array processors, the current algorithms deal with 0.25-s blocks of data and have an intrinsic 4-Hz frequency resolution. One change we are considering is to modify the software so that it can coherently combine the transforms from four 0.25-s blocks to provide 1-Hz resolution, which could improve signal-to-noise ratio and detection ranges for narrow-band targets.

We also plan to investigate the possibility of converting the signal-processing-system control processor from a PDP-11/34 to a Motorola 68000 processor, the processor now being used for all other functions in the test-bed nodes. This would allow for easier experimentation with algorithm improvements, primarily as a result of the increased virtual-address space and increased physical memory that is possible with the 68000.

B. MESSAGE-PASSING SOFTWARE

As noted in Sec.IV, the DSN test bed is transitioning to a message-based communication system - the generic type of communication system that appears to be appropriate for a DSN. The following describes the message-passing and related software that we are now implementing and how it will initially be used to support completely distributed real-time experiments. The binding of nodal tasks to processors is specific to the three-node configuration now nearing completion, but the general idea and the software will carry over to subsequent configurations in which communication and target-tracking tasks will be in separate processors.

Figure VI-1 depicts the tasks that will run in a node and the classes of message traffic between them. The azimuth-tracking 68000 will support an executive task and an azimuth-tracking task. The executive task will control the azimuth-tracking task. The azimuth-tracking history-structure building task and position-locator task in the DCU 68000 together constitute the

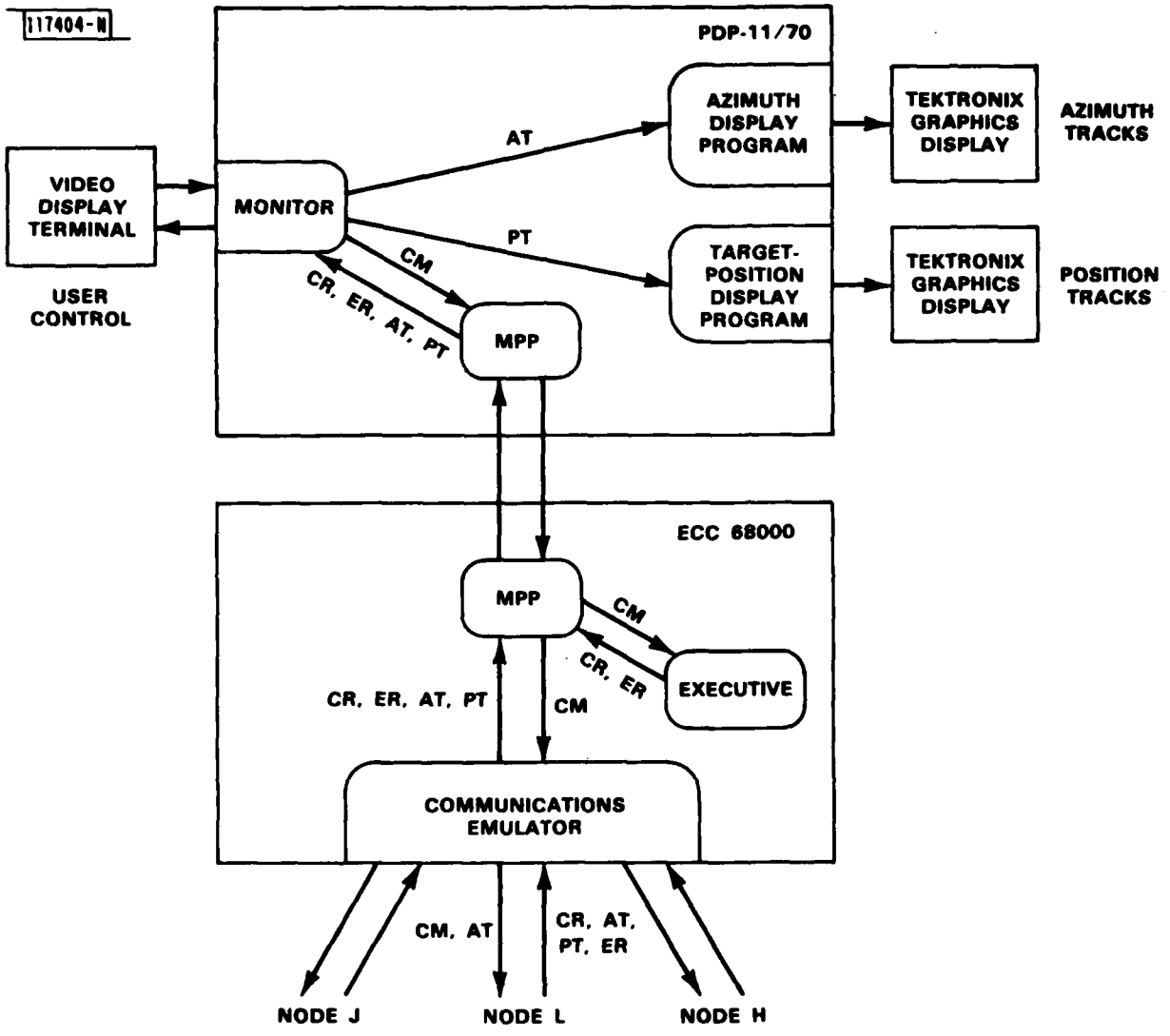


Fig. VI-2. Messages and tasks in ECC and PDP-11/70. Same legend as Fig. VI-1.

position-tracking algorithms. These tasks will run under the control of an executive routine.

All inter- and intra-node message passing will be done via a Message-Passing Package (MPP). This is a collection of routines that will enable tasks to open ports to receive messages, to form connections to ports for sending messages, and to send and receive messages. Command and reply messages will require connections to be opened explicitly. For the initial implementation, all routing tables will be linked with the MPP routines when the load module for a particular processor is formed.

User tasks will supply the MPP routines with a path name of the form "node/cpu/portname" when requesting a connection. If the node and CPU names correspond to the CPU in which the MPP is running, and if the port is found, then a connection ID will be returned to the caller to be used whenever a message is to be sent to that port. If the port name is not found then an error code will be returned to the user, indicating that the task associated with that port name is not operating. If the node/cpu/portname is not local, then the MPP looks up the requested connection in its route table to find which I/O port is to be used to send the message to its destination. Eventually, a connection ID or error message will be returned to the local MPP. Both local and remote connections will be established in a uniform manner.

As mentioned above, explicit connections to remote tasks will be required only for command messages and replies. The MPP itself will establish local port names to handle track and error messages. It will route these messages to one or more places dependent on the message class and the contents of its route table. This allows tasks such as the tracking tasks to be written without concern for which CPU they are to run in, which should simplify subsequent system expansion.

Figure VI-2 shows the corresponding messages and tasks in the Experiment Control Computer (ECC) and the PDP-11/70 computer. The communications emulator will repeat azimuth-track messages from any node to others as determined by its internal tables. It will also relay command, reply, and error messages. The communications emulator will send azimuth and position tracks

from any node to the monitor task in the PDP-11/70 for display and storage. The monitor task in the PDP-11/70 will interface experimenters to tasks operating in the nodes and to the tracking data.

At present, a single-processor version of the MPP software is operating and a multiple-processor version has been designed and is being implemented. An interprocess communication software package was obtained from Carnegie-Mellon University and adapted for use in developing the MPP. The azimuth tracker has been converted to use message passing and to operate in a multitasking environment. It is now being debugged in that environment. The executive for the azimuth tracker and a simple version of the monitor are now operational.

C. 68000 EXECUTIVE SOFTWARE

The 68000 executive software package has been extended to provide cooperative multitasking, and interrupt-driven I/O.

The multitasking features available in this new executive package include a task scheduler, and a set of functions available to the user tasks for timesharing the CPU. The scheduler does not control allotment of CPU time, but instead directs the transfer of control between user tasks. The system is intended to run a set of cooperative tasks, each using as much time as needed to accomplish an operation, and then to voluntarily release control of the CPU. Each task is written so that at specific times it relinquishes control of the CPU by one of three methods: (1) it can simply give up control and be rescheduled to run when the CPU is available; (2) it can give up control for a specific amount of time; or (3) it can give up control and wait for a flag to be set by another task. The flag feature provides a simple interprocess communication link between tasks.

The multitasking functions and the interrupt-driven I/O functions work together to provide the user tasks a powerful variety of I/O capabilities. The standard UNIX I/O utilities such as opening devices, transferring single characters, reading and writing buffers, and character string manipulation

are provided in two modes: the standard version, and the immediate return version. The latter allows the user task to determine whether sufficient data exist on input or output without having to give up control of the CPU. This feature allows for efficient utilization of the allotted processing time without having to wait for completion of an I/O operation.

The system software to control the mccm I/O ports of the ECC is 50-percent complete. It will contain the drivers for the communications emulation and experiment monitoring software that will run in the ECC. The mccm software will use queued buffers for passing messages between nodes.

D. OPERATING SYSTEMS SUPPORT

The need to remotely operate and control test-bed nodes from the PDP-11/70 has required modifications to be made to VERSABUG ROM-based monitors in the 68000's and to the UNIX operating system on the PDP-11/70.

The VERSABUG monitor has been changed to accept commands from either the console port or the host port of the processor. The port from which the commands are received is determined from the position of a local/remote switch. The monitor now also follows a transmission on/off protocol on both the console and host ports, and does so for both reception and transmission. On either port, reception of an XOFF character (DC3) will inhibit transmission on that port until an XON character (DC1) is received. In addition, an XON is sent out on a port when VERSABUG is ready to read on that port, and an XOFF is sent out when VERSABUG is finished reading on the port. Various bugs in the UNIX operating system have been fixed in order for the bidirectional XON/XOFF protocol to work correctly. These changes were largely in the drivers for various terminal devices.

Currently, VERSABUG is being changed to allow communication from either the console or host port of the DCU to the console ports of either the PDP-11/34 or the azimuth-tracking processor.

E. DATA CONVERSION

A program has been written to convert acoustic data recorded on DAS (Data Acquisition System) tapes into a set of WDU (Waveform Data Unit) files in UNIX. The DAS tapes are the acoustic data tapes produced by DSN test-bed nodes. The conversion program allows us to easily make use of an existing UNIX data-analysis-and-display package that requires the WDU files as input. The conversion program provides for preselection of data on the basis of data channel and time period. It demultiplexes the data, provides conversion of DAS voltage readings to sound pressure levels, creates a waveform file for each selected data channel, and creates the necessary index file for the waveform files. This program replaces a less-flexible multi-step procedure that we have used previously, with a single convenient operation. The spectragram shown in Fig.II-3 was produced using existing software to process acoustic data which were converted from DAS to WDU format using the new conversion program.

VII. COMMUNICATION SUBSYSTEM

A. SUBSYSTEM OVERVIEW

A communication subsystem working document has been prepared that provides an overview of the functions planned for implementation in the DSN test bed. It addresses all test-bed communication functions, including internodal communication by means of land lines as well as by means of radios. It identifies the major elements of the communication subsystem in the test bed as an experiment control computer (ECC) for the test bed and a nodal communication subsystem for each test-bed node. The nodal communication subsystem consists of a Digital Communication Unit (DCU) and a Radio Unit (RU).

Three stages of communication-system development are identified in the working paper: (1) a stage when all communications are by means of messages switched over telephone lines, (2) a stage when this is augmented by digital radios providing local broadcast services, and (3) a last stage when all communication functions are provided by radios.

Following are the functions of the nodal communication subsystem that were identified.

(1) Radio Communication:- Four experimental radio-communication services were identified for test-bed implementation. These are an unacknowledged local broadcast service, a service to support self-location by means of precise timing measurements, reliable point-to-point service, and a less-reliable real-time point-to-point service.

(2) Network Time Synchronization:- Time synchronization is an ancillary communication subsystem function that is required to support DSN surveillance and tracking and to support the normal communication modes of the system.

(3) Self-Location:- The determination of the location of the nodes in a DSN is a communication subsystem function. This will be done in a distributed manner using radio measurements of propagation time between pairs of nodes, and any additional information that users may provide.

(4) Control and Loading of Remotely Located DCUs:- The DCU is a digital computer in the nodal communication subsystem that controls the nodal communication subsystem. Provision must be made to remotely reset that computer, load it with new software, and perform remote debugging.

(5) Control and Operation of DSN Sensors and Processor Systems:- The DCU in each test-bed node is the remote interface to the nodal sensor and processor systems. The DCU must provide for remote control, downloading, and debugging of the attached DSN devices. These capabilities, and those mentioned under (3) above, will be provided first by means of phone lines connecting the nodes to the ECC and subsequently by means of radios.

(6) Internodal Broadcast Communication Simulation:- One test-bed mode of operation will be to provide for internodal broadcast of messages by means of phone lines interconnecting the nodes and the ECC in a star configuration.

B. RADIO COMMUNICATION SERVICES

An outline has been prepared for a second working paper dealing in more detail with the radio communication functions and the functional characteristics of the RU/DCU interface. Drafts have been completed for sections dealing with channel organization and basic channel-access protocols, with

broadcast and self-location support services, and with the exchange of status and command messages between the DCU and the RU. Simplified radio services, sufficient for initial test-bed experiments with radios, are now being defined for initial implementation. The following summarizes the completed sections.

The nodal communication subsystem radio will be a pseudonoise spread-spectrum system with bit-by-bit spreading-code changes throughout each packet. In addition, the code for the first bit of each packet (called the seed) will be changed as a function of time. It will be possible to change the seed as often as once every 10 ms; therefore, the radio channel has been broken into 10-ms code slots so that the changing of code seeds can be organized and coordinated.

Packet transmissions can start at any time within a seed slot and be of any length. They can extend over any number of seed slots. If a transmission is to start in a particular slot, the start time within the slot will be selected randomly. The normal mode of operation will be with the receiver active up until the selected transmission time. If a reception has started but is not completed by the selected time, the transmission will be aborted. It will be possible to restrict allowed starting times in seed slots to a beginning interval, and in that way implement slotted as well as unslotted communication services. Figure VII-1 shows the division of the channel into seed slots, and shows more details within a slot for the case when the start times have been restricted to a beginning interval. The seed slots are further collected into groups to provide a simple mechanism for the organization of slot usage as reviewed below.

The slots within a group of slots will be organized into a set of disjoint lists as indicated in Fig.VII-2. Each list will correspond to an allocation of slots to a communication service or task. The allocations will be quasi-static. A list defines the times during which the associated service can initiate a transmission or start a reception. Different services can be made noninterfering by restricting packet sizes and causing transmissions and receptions to not extend beyond the allocated slots.

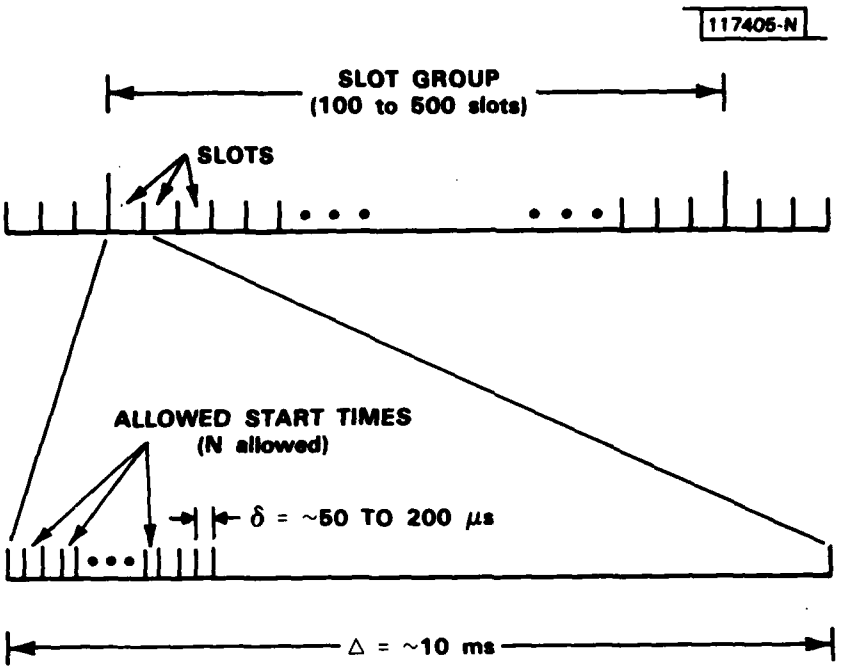


Fig. VII-1. Organization of radio channel into groups of slots with discrete packet start times within each slot.

Unacknowledged broadcast services will constitute the basic test-bed communication services. From the point of view of a user, a DSN broadcast service will behave as a packet queue with known deterministic and stochastic characteristics. Each broadcast service is allocated a set of slots that is specified by one of the lists mentioned above. Packets will be handled in the order they are presented. If a packet is at the head of the queue, a transmission will be scheduled with probability p for the next available allocated slot. A transmission attempt may be aborted by an incoming packet, in which case the outgoing packet remains at the head of the queue. The maximum time a packet can stay in the queue without being broadcast will be limited by specifying the maximum number of allocated slots that can pass by while the packet is in the queue. When the maximum is exceeded, the packet is discarded and the user is informed. The number of packets accepted by a broadcast queue during a moving time window will be restricted. This will provide for fair access to a common broadcast service by several nodes in an area.

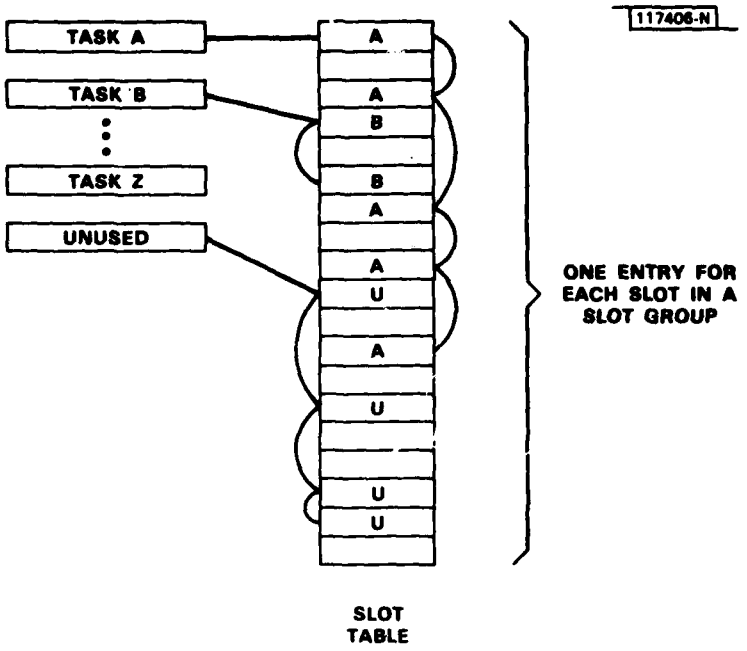


Fig. VII-2. Allocation of disjoint sets of code slots to distinct tasks. Allocated slots are those within which task can start a transmission.

Associated with each allocated slot will be a code-seed type and channel-access privileges. These can be different for different nodes, although the allocated slots may be the same. This will provide necessary flexibility for future implementations of services using receiver-directed techniques. The code-seed type will specify if the standard changing seed sequence is to be used, or if a fixed common seed is to be used, or if a user specific seed sequence is to be used. The channel-access privilege will specify if transmissions are allowed for a slot.

Not all features discussed above will be required for initial DSN experiments with radios in the test bed. A much-simplified broadcast capability, with features listed below, is being planned to support initial experiments. All seed slots will be allocated to a single service. All slots will allow transmission. Either a constant or changing common seed will be used, based on user determination, on a packet-by-packet basis. The broadcast queue

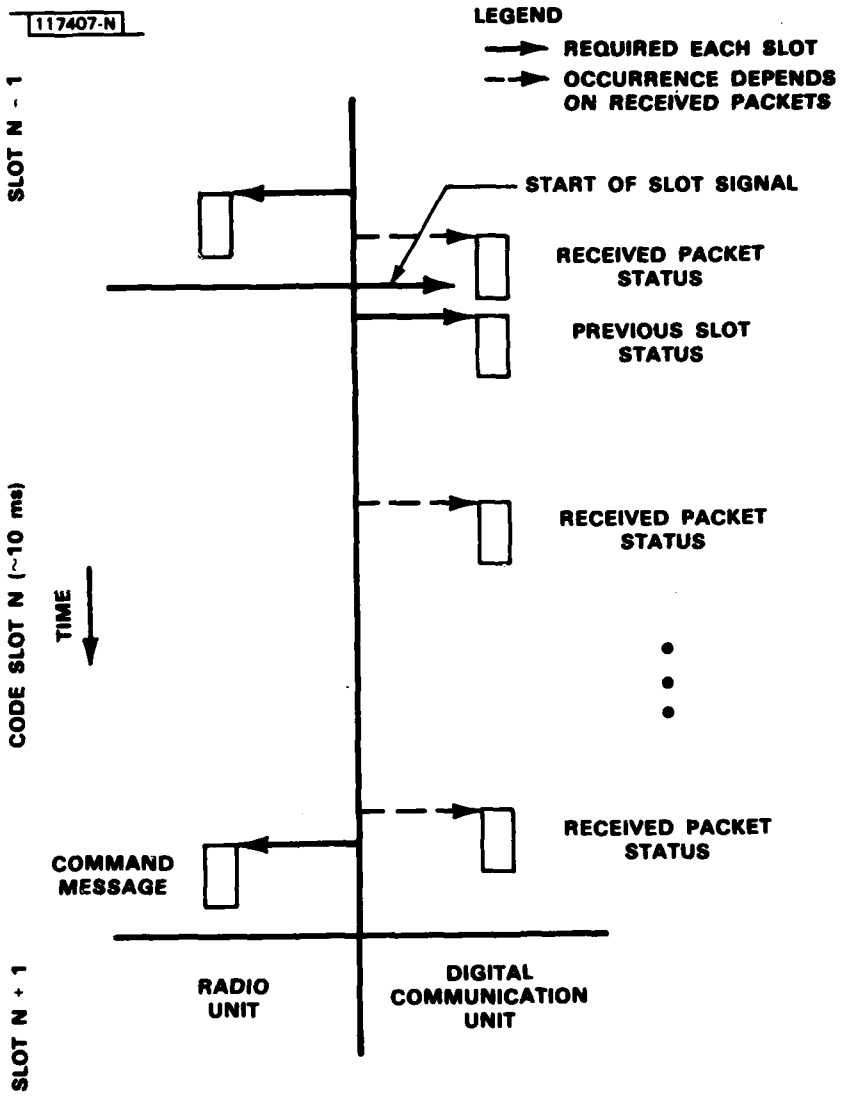


Fig. VII-3. Exchange of command and status messages between DCU and RU.

length will be limited to one packet. In addition, a round-trip timing service will be implemented to support self-location experimentation. It will randomly share the channel with the broadcast service.

The general scheme for the interactions between the DCU and RU is indicated in Fig.VII-3.

Once in each code slot, a command message will be sent from the DCU to the RU. This message controls the RU functions. A command packet can cause the transmission of, at most, one packet, so no more than one packet transmission can be started in a 10-ms interval. The command packet will specify the length of the next 10-ms interval in terms of counts of the RU hardware clock. This will be used to slew the clock and obtain network synchronization. As shown in Fig.VII-3, the command packet is transferred late in a slot and controls the next slot. We are also considering sending the command packet at the very start of each slot. The RU will provide a signal to the DCU when the count is reached.

Two kinds of status messages will pass from the RU to the DCU. One will be passed shortly after the end of each nominal 10-ms interval. It will contain general information about the operation of the RU in the previous slot as well as a report on the transmission activity of the RU during that slot. Also, an information message (in addition to the actual data) will be sent to the DCU as each packet reception is completed. This will contain packet specific information such as time-of-arrival information and error conditions. As shown, a packet specific-information message late in a slot may cause the general-information message to be delayed slightly. Note that several packets may be received within a single 10-ms interval or only a fraction of a packet may be received.

C. DIGITAL COMMUNICATION UNIT HARDWARE

After reviewing the options, we have selected the Motorola 68000-based SUN (Stanford University Network) board as the basic processor board for the DCU. The basic unit will also contain a memory board and a DMA board to interface with the RU. We will construct the DMA board using the 8089, which

is an intelligent DMA controller IC that is available from Intel. Each 8089 can handle two separate DMA channels, and two 8089's are designed to cooperatively share the support circuitry providing four DMA channels. Each controller can execute simple channel programs as well as perform DMA transfers. This will give the interface sufficient flexibility to deal with variable-length received messages using one of several methods. With the associated mature Multibus interface ICs, the RU interface can be implemented with minimal design effort.

The functional utilization of the DMA channels has not changed from that specified in the last report.* The detailed description of the hardware interface to the radio is in the process of being defined. Since multiple small packets may be received in a 10-ms code interval, there may be multiple status reports from the radio - one for each reception. These status reports are in addition to the overall status report for the 10-ms code interval. The intelligent DMA controller will be used to offload the processor from the time-critical task of re-initializing the DMA channel with new starting-address and byte-count parameters.

*Distributed Sensor Networks Semiannual Technical Summary, Lincoln Laboratory, M.I.T. (30 September 1981).

UNCLASSIFIED

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

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER ESD-TR-82-044	2. GOVT ACCESSION NO. AD-A118 868	3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)* Distributed Sensor Networks		5. TYPE OF REPORT & PERIOD COVERED Semiannual Technical Summary 1 October 1981 – 31 March 1982
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR*(s)* Richard T. Lacoss		8. CONTRACT OR GRANT NUMBER*(s)* F19628-80-C-0002
9. PERFORMING ORGANIZATION NAME AND ADDRESS Lincoln Laboratory, M.I.T. P.O. Box 73 Lexington, MA 02173-0073		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS ARPA Order 3345 Program Element Nos. 61101E and 62708E Project Nos. 2D30 and 2T10
11. CONTROLLING OFFICE NAME AND ADDRESS Defense Advanced Research Projects Agency 1400 Wilson Boulevard Arlington, VA 22209		12. REPORT DATE 31 March 1982
		13. NUMBER OF PAGES 56
14. MONITORING AGENCY NAME & ADDRESS *(if different from Controlling Office)* Electronic Systems Division Hanscom AFB, MA 01731		15. SECURITY CLASS. *(of this report)* Unclassified
		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
communication network
acoustic sensors
low-flying aircraft
acoustic array processing
digital radio

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

This report describes the work performed on the DARPA Distributed Sensor Networks Program at Lincoln Laboratory during the period 1 October 1981 through 31 March 1982.

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

UNCLASSIFIED

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