ESD-TR-67-182

Technical Report 431

The Lincoln Experimental Terminal

J. W. Craig
W. R. Crowther
P. R. Drouilhet, Jr.
J. N. Harris
F. E. Heart
B. H. Hutchinson
I. L. Lebow
P. G. McHugh
B. E. Nichols
P. Rosen
S. B. Russell
J. Tierney

21 March 1967

Prepared under Electronic Systems Division Contract AF 19(628)-5167 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, with the support of the U.S. Air Force under Contract AF 19(628)-5167.

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
LINCOLN LABORATORY

THE LINCOLN EXPERIMENTAL TERMINAL

J. W. CRAIG
W. R. CROWTHER
P. R. DROUILHET, JR.
J. N. HARRIS
F. E. HEART
B. H. HUTCHINSON
I. L. LEBOW
P. G. McHUGH
B. E. NICHOLS
P. ROSEN
S. B. RUSSELL
J. TIERNEY

TECHNICAL REPORT 431

21 MARCH 1967

LEXINGTON MASSACHUSETTS

FOREWORD

Lincoln Laboratory has designed and constructed three satellite communications ground terminals (Lincoln Experimental Terminals) as part of its program in space communications. The first of these (LET-1) is a self-contained transportable, X-band terminal in which the RF equipment, antenna and mount are on one vehicle, and the signal processing equipment, prime power and operating console are in a second van. The other two systems (LET-2 and LET-3) consist of signal processors identical to that of LET-1. They are similarly housed in vans and are used in conjunction with antennas and RF equipment associated with other terminals. The three systems were completed in 1965 and tested extensively since that time, using active satellite repeaters and the moon as a passive reflector.

The function of the terminals is to demonstrate satellite communication in which multiple-access and anti-jam performance are emphasized. The use of vocoders for speech compression and an efficient modulation and coding system for signaling result in a low required satellite power per access. Bandspreading by pseudorandom frequency hopping provides the desired multiple-access and anti-jam capabilities.

This report is a description of selected portions of the system emphasizing the most novel features. The first chapter gives a general description of LET-1 as a whole, and the second an over-all description of the signal processing system. Chapter 3 discusses the equipmental realization of the signal processing system and Chapter 4 the utilization of a general-purpose computer as an element of the signal processing system. Chapters 5 and 6 treat two specific subsystems: the sequential decoder and the vocoder.

Accepted for the Air Force
Franklin C. Hudson
Chief, Lincoln Laboratory Office

CONTENTS

THE LINCOLN EXPERIMENTAL TERMINAL

CHAPTER 1
LINCOLN EXPERIMENTAL TERMINAL

B. E. Nichols and P. Rosen

ABSTRACT

The Lincoln Experimental Terminal (LET) is a complete, self-contained air-transportable ground terminal for testing and demonstrating evolving space communications techniques in a realistic environment. Its present equipment complement permits efficient, highly reliable, multiplexed digital communication of voice and record traffic with a variety of channels, including the moon and active satellites. Its modulation system, using a 16-symbol alphabet frequency-hopped over a 20-MHz band, together with efficient coding, provides multiple access to a wide-band satellite.

A. INTRODUCTION*

For a number of years the M.I.T. Lincoln Laboratory has worked on various techniques applicable to the solution of space communications problems. In addition to microwave technology and components such as cooled X-band parametric amplifiers and rapidly switchable frequency synthesizers, these techniques have included: modulation and demodulation for dispersive channels such as the troposphere, Moon and the West Ford belt; digitized narrowband speech processing with emphasis on speaker recognizability; and practical realization of coding and decoding schemes which are economically competitive with the more conventional means of achieving greater information rates on a given channel.

Some results of this work have been combined to produce an experimental air transportable terminal, called the Lincoln Experimental Terminal (LET), which has a number of desirable and unique features, particularly from a military communications viewpoint. The terminal will work efficiently on both coherent and time-varying dispersive channels; it provides good quality digital speech with speaker recognizability in a reasonably narrow band; and it permits multiple access of a broad-band satellite by spectrum spreading, without the severe synchronization problems commonly associated with the use of a pseudonoise carrier for this purpose.

* Numbered references for Chapter 1 are given on page 17.

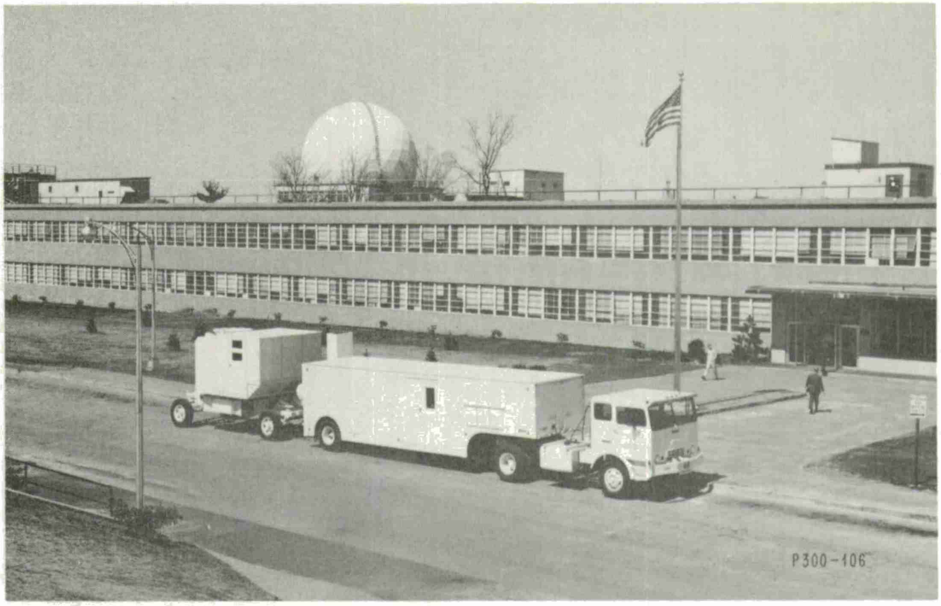


Fig. 1-1. LET ready for travel.

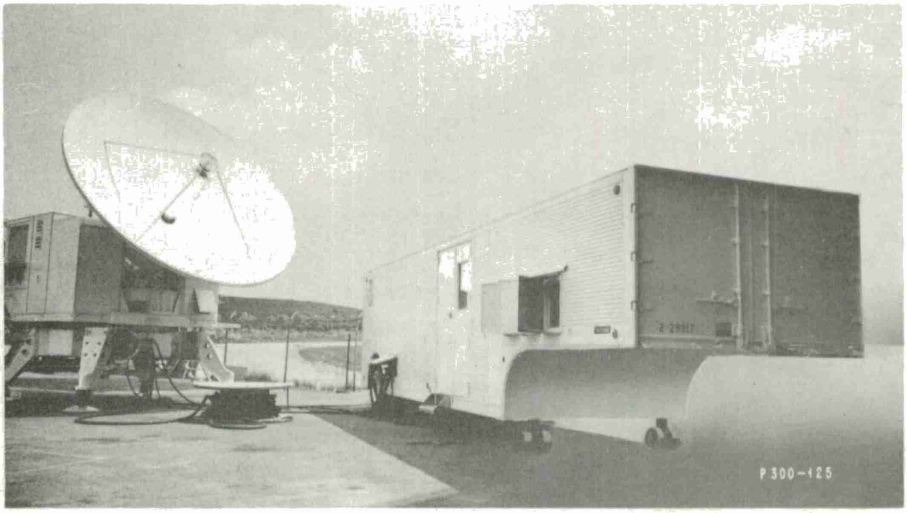


Fig. 1-2. LET in operation.

B. GENERAL DESCRIPTION OF TERMINAL

Three terminals have been constructed in the LET program. The first of these is self-contained in two trailers and is shown in its road configuration in Fig. 1-1. One trailer, the so-called electronics vehicle, is a modified low-bed commercial van which contains the signal processing equipment, a communications and antenna control console, a prime power generator and its fuel, an air conditioner, and storage for the antenna panels. The second trailer, which we call the antenna vehicle, contains the transmitter and its heat exchanger, a refrigerated parametric-amplifier receiver, low-level microwave equipment, the antenna backup structure, feeds and servo-mechanism equipment. Figure 1-2 shows the terminal in operation.

The terminal has the following gross characteristics:

Transmitter frequency	~8000 MHz
Receiver frequency	~8000 MHz
Transmitter power	10 kW CW
System bandwidth	20 MHz
System noise temperature	100°K
Antenna diameter	15 ft
Antenna pointing	Computer-aided autotracking
Information rates	Up to 9600 bits/s
Information types	Multiplexed vocoded voice, teletype and data

Two additional sets of signal processing equipments have also been constructed and housed in vans identical with that associated with the first terminal. These have been designed to operate with antennas and RF equipments associated with existing terminals. One of these has been used with the 60-ft X-band West Ford system at Westford, Massachusetts, and the other, which was funded in part by the U.S. Army Satellite Communications Agency, has been used with the Mark 1A terminal at Camp Roberts, California and, later, at Ft. Monmouth, New Jersey.

C. SYSTEM CONSIDERATIONS IN SIGNAL PROCESSING DESIGN

The specific implementation of the techniques mentioned previously stems from a number of self-imposed specifications. These may be listed roughly as:

(1) The terminal should be able to operate on almost any channel, passive or active, coherent or dispersive.

(2) The terminal should continue to operate with high efficiency in the presence of fortuitous or deliberate interference.

(3) The terminal should operate with very high efficiency, i.e., it should provide very reliable, highly accurate output with low input signal-to-noise ratio.

(4) The terminal should provide digitized, good quality (speaker recognizable) vocoded voice.

(5) The terminal should be usable with a variety of inputs ranging from record traffic to voice, easily and flexibly multiplexed.

(6) The operation of the terminal should be close to automatic.

A conventional way of meeting the requirement that a terminal continue to operate under severe interfering conditions is to use a pseudonoise carrier in a broad-band system. Unfortunately, the ability to do so is in conflict with the desire to operate on dispersive channels.

However, a frequency-hopped modulation scheme which minimizes intersymbol interference, and is therefore well suited for high-rate modulation on dispersive channels, can also have excellent resistance to interference if used with efficient coding.

The signal processing system in LET is described in detail by Drouilhet.[1] The signal structure is sketched in Fig. 1-3. The elementary channel symbol used is a sinusoidal pulse 200 μs in duration on one of 16 frequencies. The received pulse is detected in 16 matched filters followed by envelope detectors.[2] This 16-ary orthogonal alphabet is used in conjunction with sequential coding[1,2,3,4] for the information transmission as shown in Fig. 1-3. Information rates of approximately 5000 and 10,000 bits/s are achieved corresponding to 1 or 2 information bits, respectively, per pulse.

The system thus formed operates with an energy-to-noise ratio, E/N_o, per bit of 6 dB at either rate on a Gaussian channel. This signal-to-noise ratio corresponds roughly to error probabilities per pulse before decoding of 30 and 10 percent for the two rates. At the output of the decoder, the error probability is very low ($\sim 10^{-5}$) for E/N_o values a fraction of a decibel above the threshold value.

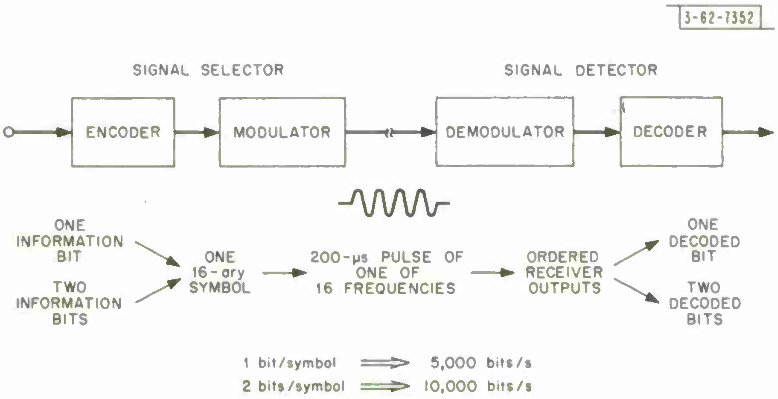


Fig. 1-3. Signal processing system.

The set of 16 frequencies used for every pulse is selected in a pseudorandom way from a total of 4096 frequencies. The band spreading thus obtained gives multiple access or anti-interference operation comparable with that achievable by pseudonoise techniques.[5] Since the frequency hopping takes place every 200 μs, the synchronization requirement on the system is modest compared with that necessary for a pseudonoise system of equivalent bandwidth. Signals are automatically acquired using time and frequency predictions obtained from a station clock and the satellite ephemeris. The desired accuracy is achieved by tracking a synchronization pulse.

Digitized speech (multiplexed with two 100-wpm teletype channels) may be transmitted at either the 4800- or 9600-bit/s rate by using an experimental vocoder designed as part of the terminal.[6] The vocoder, whose design reflects emphasis on speaker recognition, operates in a pitch-excited mode with high quality input when used at the lower rate. At the higher rate, the vocoder is used in a voice-excited mode, allowing the use of degraded input, including a "phone patch" connection to the commercial telephone plant.

A small, general-purpose digital computer (UNIVAC 1218) is used as an integral part of the communication terminal, and performs several simultaneous functions.[7] Given the orbital parameters of the satellite to be used, it computes pointing commands for the antenna during the satellite acquisition phase; it also simultaneously computes and delivers Doppler and range information to the communication system control. Concurrent with its orbital computations, the computer is also used to multiplex and demultiplex the terminal input data which may consist of digital voice, teletype and high-speed data. Although the computer is programmed at present to multiplex (and demultiplex) one voice channel, two teletype channels, and data up to a total of 9600 bits/s, the mixture of inputs may be changed with relatively minor program changes rather than by extensive equipment retrofit.

A block diagram of the system is shown in Fig. 1-4. Digital signals multiplexed by the computer are encoded and sent to the frequency synthesizer where the selected one of 16 frequencies is translated under control of the pseudorandom sequence generator to the wide bandwidth with appropriate timing and Doppler changes. Another translation of this wide band of signals to microwave frequencies is followed by amplification to the kilowatt level and fed to the antenna. The signals received by the antenna (in another frequency band) enter the low-noise parametric amplifier and are then translated in frequency back to IF for further amplification. After detection, the signals are decoded, separated and delivered to the vocoder and teletype machine.

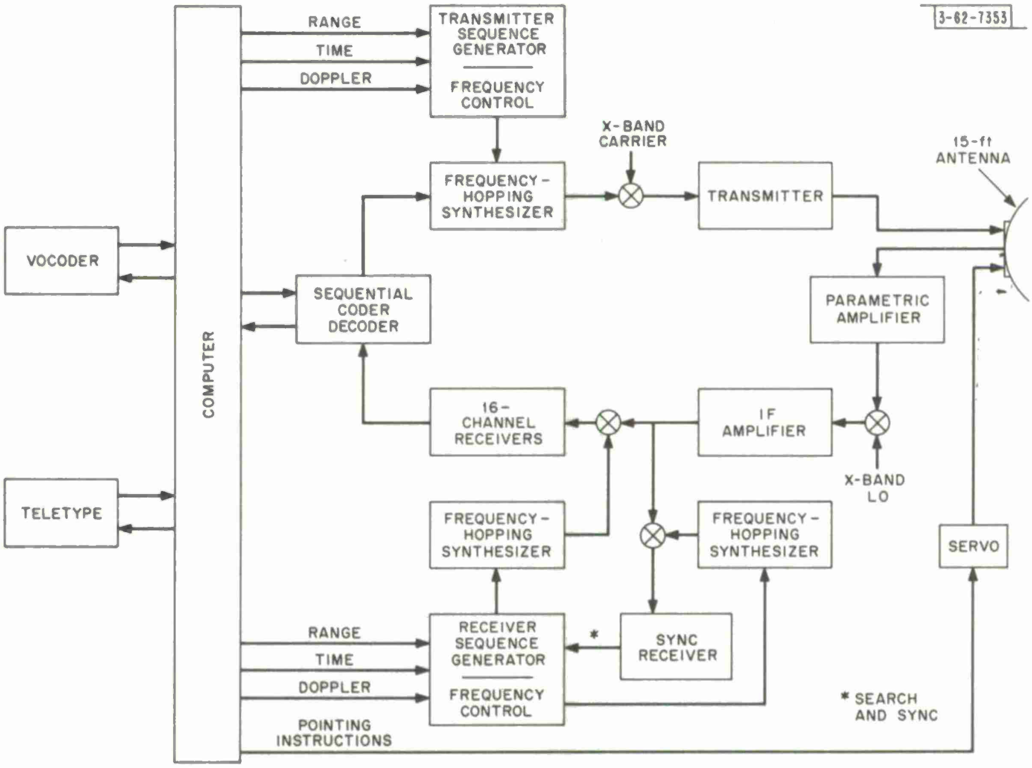


Fig. 1-4. LET system.

An example of LET performance, when operating in an enviornment requiring multiple access to a satellite, may be useful. The following satellite characteristics are assumed:

Altitude	23,000 mi
Frequency	8,000 MHz
Receiver noise figure	10 dB
Power output	2.0 W
Antenna gain (receive)	0 dB
Antenna gain (transmit)	6 dB
Bandwidth	20 MHz

TABLE 1-1

GROUND-TO-SATELLITE LINK

Transmitted power	+70 dBm (10 kW)
Ground antenna gain	48 dB
Satellite antenna gain	0 dB
Space attenuation	−202 dB
Received carrier power	−84 dBm
Noise power density	−199 dBm/°K/Hz
Receiver noise temperature (2700°K)	34 dB
Receiver bandwidth (20 MHz)	73 dB
Receiver noise power	−92 dBm
Carrier-to-noise ratio	8 dB

TABLE 1-2

SATELLITE-TO-GROUND LINK

Satellite transmitter power	33 dBm (2 W)	Carrier-to-noise ratio	64 dB/Hz
Satellite antenna gain	6 dB	Rate (5 kbits/s)	37 dB
Effective radiated power (ERP)	39 dBm (8 W)	E_b/N_o	27 dB
Space attenuation	−202 dB	Threshold E_b/N_o	6 dB
Ground antenna gain	48 dB		
Received carrier power	−115 dBm	Margin available for link degradation and multiple access capacity	21 dB
Noise power density	−199 dBm/°K/Hz	Margin for link degradation	5 dB
Receiving system noise temperature (100°K)	20 dB	Margin available for multiple access (40 users)	16 dB
Receiver noise power density	−179 dBm/Hz		

The signal and noise levels are computed in Tables 1-1 and 1-2 for both the up and down links; then the number of LET terminals that could use the same satellite is computed. The assumption is made that all terminals radiate the same power, and that they are equidistant from the satellite.

From Table 1-2, it may be seen that LET needs only about 200 mW of the 8 watts ERP available from the satellite allowing 5 dB of margin or about 65 mW with no margin. The implication is that even under degraded link conditions (5 dB) as many as 40 users (20 duplex circuits) could communicate simultaneously, provided that they used the satellite power intelligently by spreading their signals across the 20-MHz satellite band. One may then ask how much the effective receiving system noise temperature rises under the circumstances postulated above. The worst case, i.e., where all the satellite power appears at the receiver as interfering noise, is computed as:

Received carrier power	− 115 dBm
Satellite bandwidth	73 dB
Receiving system noise power density	− 179 dBm/Hz
Interference to receiver noise ratio	− 9 dB
Increase in receiving system noise	12.7°K

where it is seen that the effective receiving system noise temperature is increased by only 12.7°K.

D. ANTENNA AND FEED SYSTEM

The LET antenna is a 15-ft-diameter paraboloid employing a Cassegrainian feed system.[8] Some of the more common antenna and feed parameters are given below.

Antenna gain (50-percent efficiency) including losses	48 dB
Half-power beamwidth	0.58°
First side lobes	⩾20 dB
Transmitting polarization	RHCP
Receiving polarization	LHCP
Isolation between transmitting and receiving modes	>20 dB
Axial ratio	<2 dB
Operating frequency band	7200 to 8400 MHz

E. ANTENNA VEHICLE

The antenna vehicle is basically an elevation-azimuth pedestal carrying an equipment shelter as well as the antenna. The shelter contains the antenna drive system, the RF receiver, the transmitter and its power supply, receiver and transmitter cooling, microwave excitation equipment and test equipment. Road transportability is achieved by attaching a fixed wheel and axle assembly to one end of the pedestal base and a steerable wheel and axle assembly to the opposite end. During over-the-road travel, the antenna is disassembled and stowed inside the forward compartment of the electronics van. The equipment shelter mounted on the pedestal is about 8 × 10 ft and rotates with the antenna in azimuth. Prime power and signal frequencies up to the IF (60 MHz) are brought into and taken out of the equipment shelter through slip rings.

F. RF TRANSMITTER

The LET transmitter, shown in Fig.1-5, designed to operate at X-band with an instantaneous bandwidth of 20 MHz, develops 10 kW of CW power. Its tube, a Varian Type 885B klystron, has a tuning range of 7700 to 8400 MHz. Modulating signals from the electronics van are sent via coaxial cable at the 60-MHz IF to the antenna vehicle, where these signals are translated to X-band in the transmitter driver. The transmitter power supply has an output capacity of 38 kW at output voltage of 16 kV, and various taps permit transmitter outputs of 10, 5, 2.50 and 1.25 kW, respectively. A Varian Type 849 klystron operating at a fixed frequency can also be used in this transmitter with no change in power supply. Control of the functions involved in adjusting or tuning the transmitter are carried out in the antenna trailer shelter. RF drive and monitoring of transmitter operation are accomplished in the electronics van. Hence, an operator is needed in the antenna vehicle only for a short time at the beginning of an operational period. A photograph of the exciter, receiver IF amplifiers and test equipment is shown in Fig.1-6.

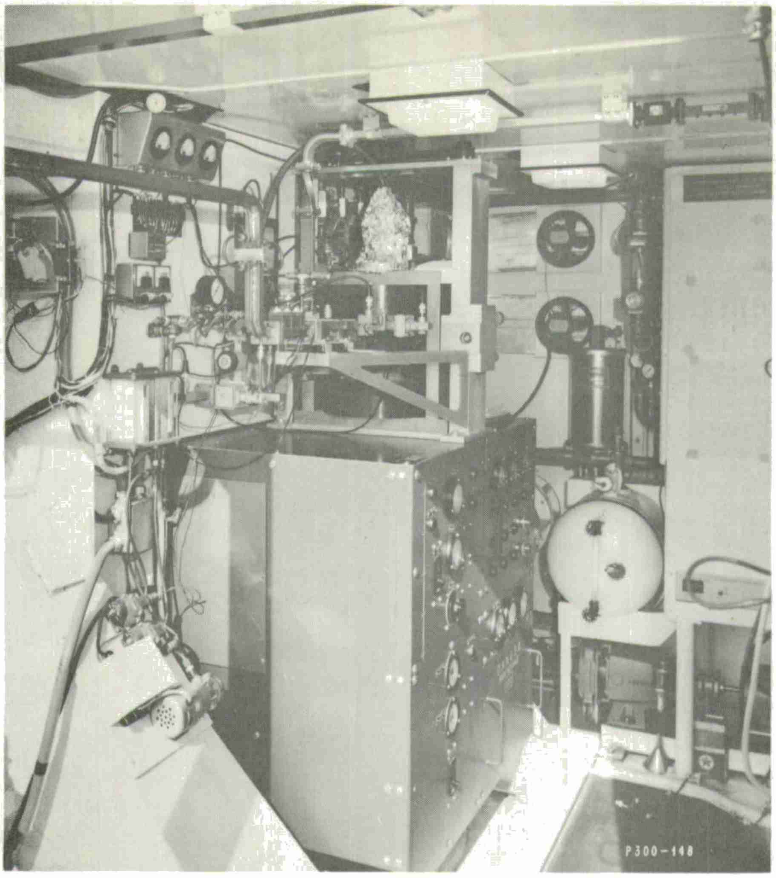


Fig. 1-5. Transmitter.

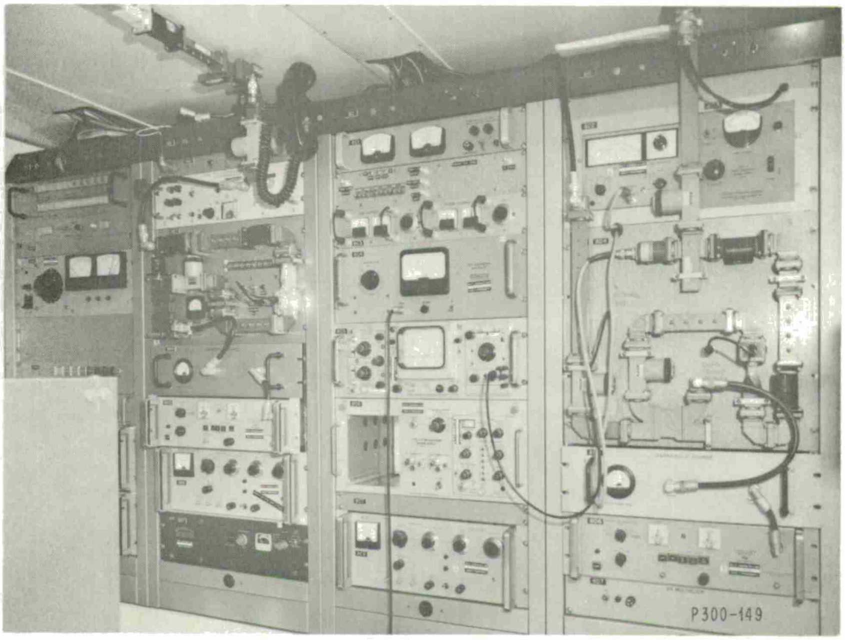


Fig. 1-6. Exciter-receiver.

G. RF RECEIVER

The Lincoln Laboratory designed receiving system uses a refrigerator-cooled, tunable parametric amplifier[9] as a front end, followed by a mixer for translation to the 60-MHz IF. The frequency range and the bandwidth of the parametric amplifier are 7.2 to 8.4 GHz and 20 MHz, respectively.

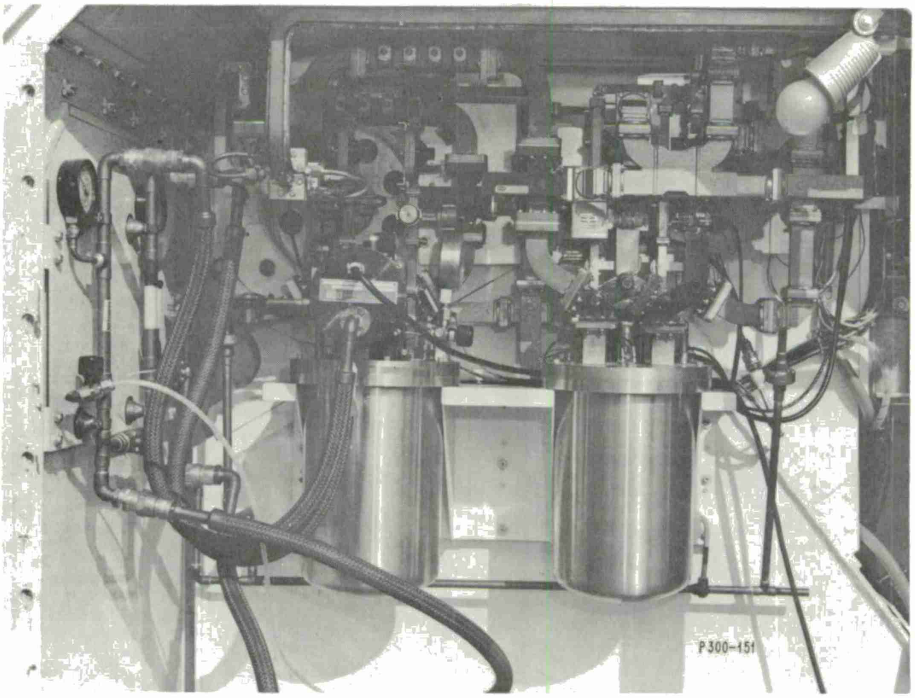


Fig. 1-7. Parametric amplifiers.

An Arthur D. Little Model 340 gaseous helium refrigerator is used with a very slow speed (75 strokes per minute) regenerator. The refrigerator operates at about 17°K when loaded with a two-stage parametric amplifier (Fig. 1-7). Under these conditions, with the parametric amplifier's output connections properly terminated, the noise temperature at the terminating flanges is about 55°K. The overall receiving system noise temperature is about 100°K. A second receiving channel is also included for the autotrack error signal, but it uses an uncooled parametric amplifier similar to the one previously described, running at about 300°K. Control, calibration and measurement of receiver performance is accomplished by remote control in the electronics vehicle at the operations console.

H. PRIME POWER

Prime power is furnished at 400 Hz, 3-phase, 120/208 volts from one of several alternate sources. Distribution of this power is controlled from a power distribution panel in the electronics van, where the power is distributed to four main trunks. One design objective of this power distribution system is to isolate, as much as is practical with one generator, the loads with large transient currents from those with relatively steady current requirements. When LET is used as a self-sufficient terminal in the field, prime power is furnished by a gas turbine driving a 100-kW, 3-phase, 400-Hz alternator. JP4 fuel for the turbine is stored in a 350-gallon

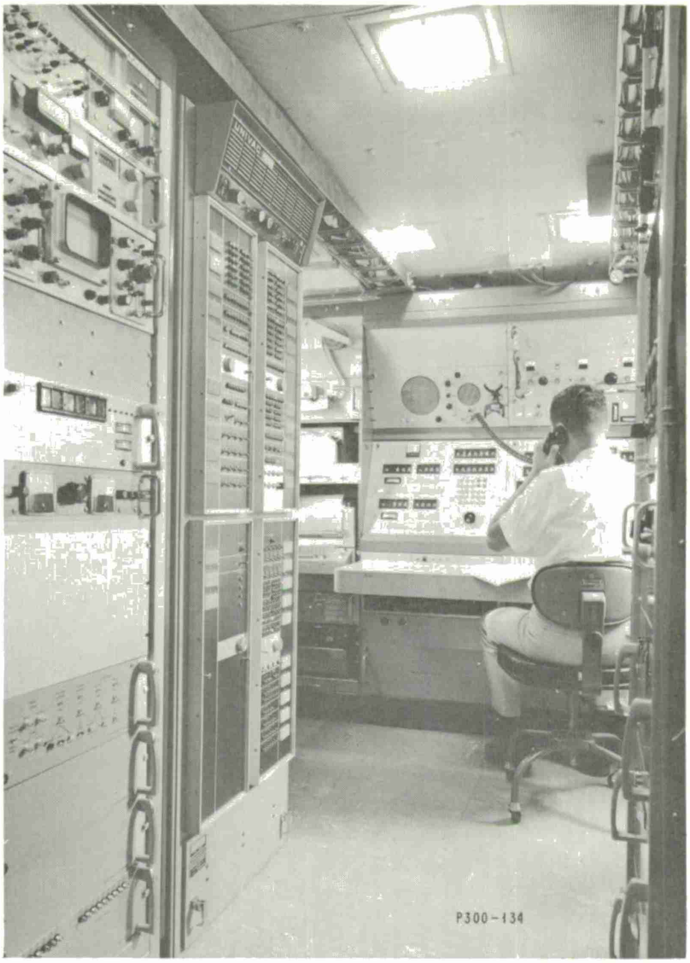


Fig. 1-8. Electronics vehicle in operation.

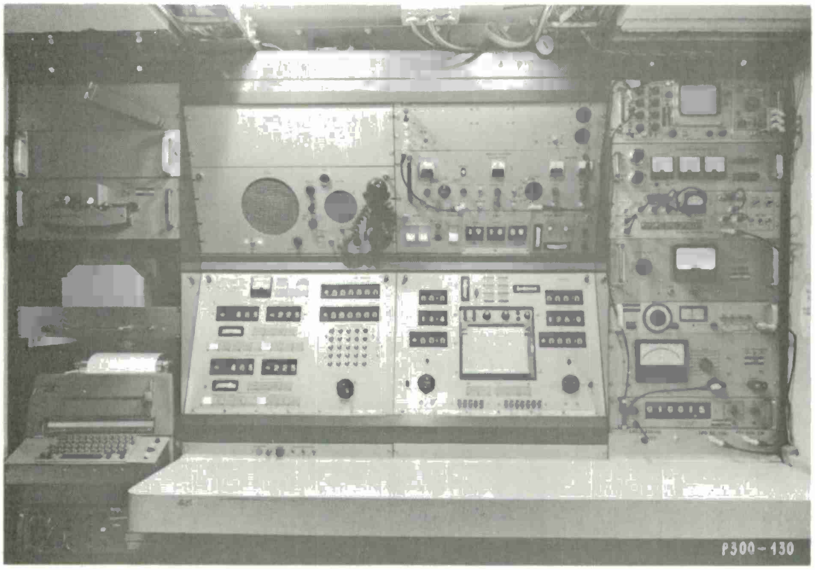


Fig. 1-9. Console.

fuel tank which forms the floor of the prime mover compartment in the rear of the electronics van. Fuel consumption at rated load is about 230 lb/hour, providing about 15 hours of operation on one tank filling. The following turbine-alternator specifications may be of interest:

Turbine alternator weight	1,100 lb
Fuel weight	2,450 lb
Turbine shaft speed	40,000 rpm
Alternator speed	6,000 rpm

For those occasions when full prime power is not needed, a 5-kW gasoline-engine-driven 400-Hz, 3-phase alternator is provided.

I. ELECTRONICS VEHICLE

The electronics vehicle is a commercial 30-ft 6-inch semitrailer with dimensions that allow it to be loaded into a C-130E aircraft. The total inside length is 30 ft, and full-width opening doors are provided at both ends. Two lateral partitions divide the internal body into three compartments:

(1) The prime power compartment in the rearward 8 ft of the body contains the turbine-driven alternator, the auxiliary gasoline-engine-driven alternator and fuel for both turbine and gasoline engine.

(2) The electronics compartment in the central 15 ft contains all the operating controls for LET plus all the electronics for that part of the terminal operating below 60 MHz.

(3) The air conditioning and storage compartment in the forward 7 ft contains the electronics compartment air conditioner plus storage racks for the panels and secondary reflector of the 15-ft antenna.

One possible problem with this configuration is control of noise generated by the gas-turbine alternator. To keep this noise at a reasonable level, the turbine exhaust is directed upward into a 6-ft chimney with sound-absorbing walls. The turbine inlet manifold is lined with sound-absorbing material and is baffled to eliminate direct line sound paths from the turbine compressor inlet to the outside. The electronics trailer is insulated on all sides with about three inches of sound-absorbing material. The partition and the hatch between the prime power and the electronics compartments have six inches of sound-absorbing material.

Figure 1-8 shows the inside of the electronics vehicle, looking toward the control console. Figure 1-9 displays the control console with teletype equipment to the left, communications and computer control in the center left, antenna control in the center right and microwave receiver control and test equipment at the far right. The signal processing equipment is shown in Fig. 1-10, with the computer at the far right. An operator is adjusting the tape recorder located just above the four drawers that constitute the vocoder. Below the vocoder is the encoder-decoder plus its memory, located at the bottom of the rack. In the rear rack are located the channel receivers and the frequency synthesizers.

Figure 1-11 is a photograph of a frequency synthesizer, showing the construction techniques used for analog circuits. Figures 1-12 and 1-13 depict the encoder-decoder as an example of digital circuit construction techniques which use integrated circuits. Packaging in this manner allowed the entire system, including test equipment, to fit into four racks.

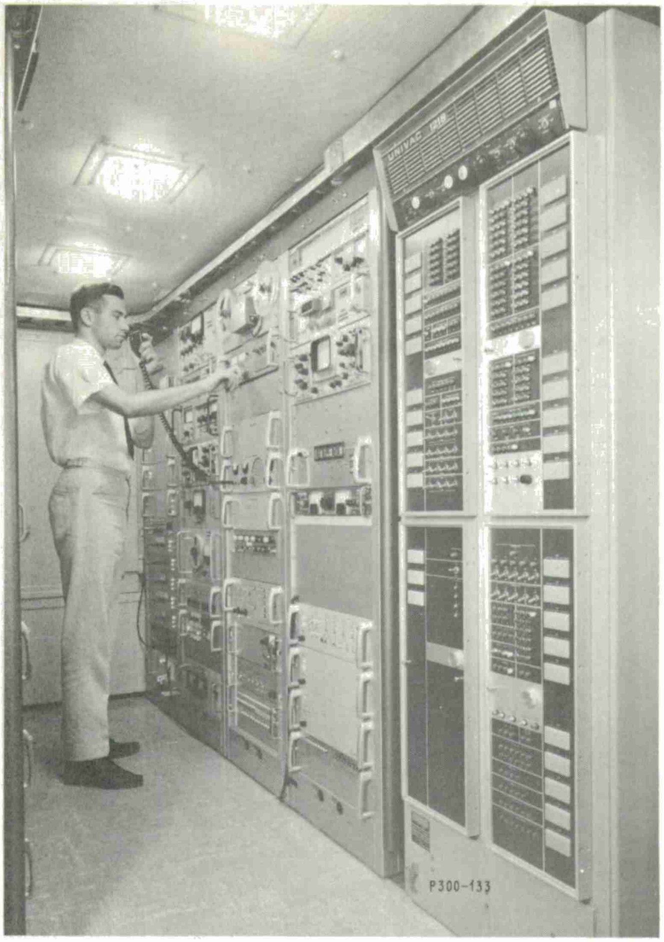


Fig. 1-10. Signal processing equipment.

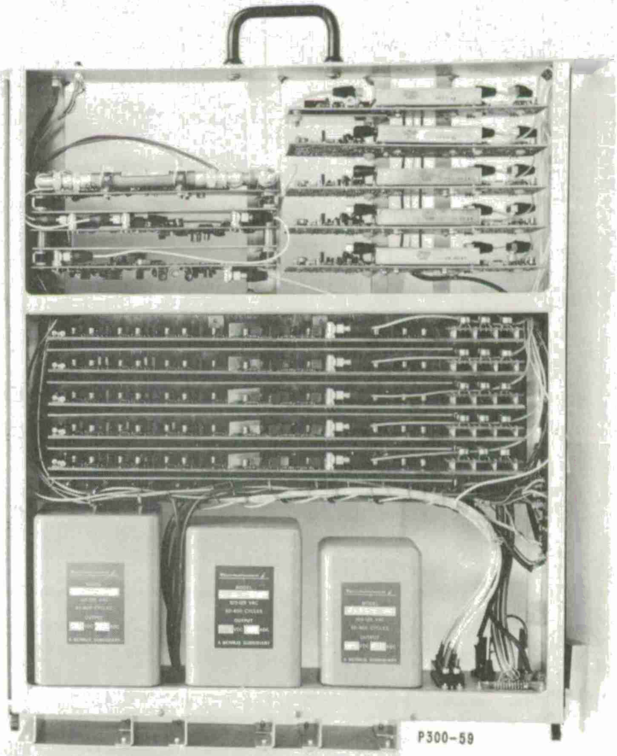
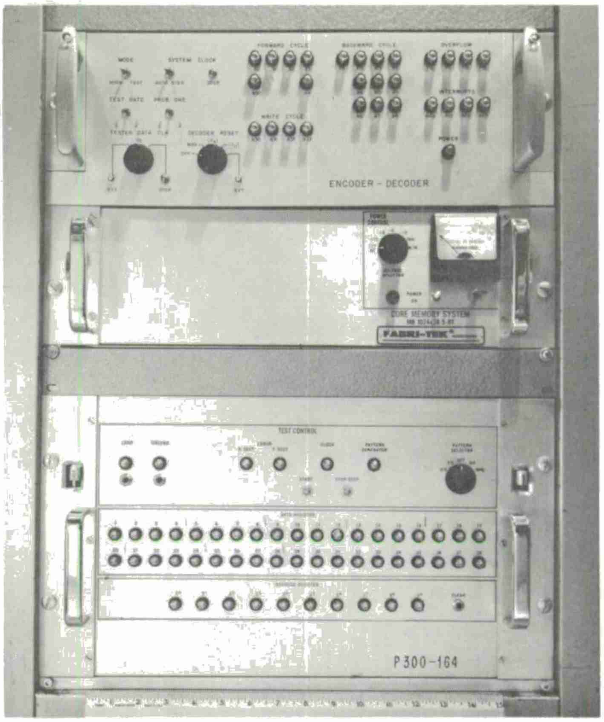


Fig. 1-11. Frequency synthesizer.

Fig. 1-12. Sequential encoder/decoder front view.



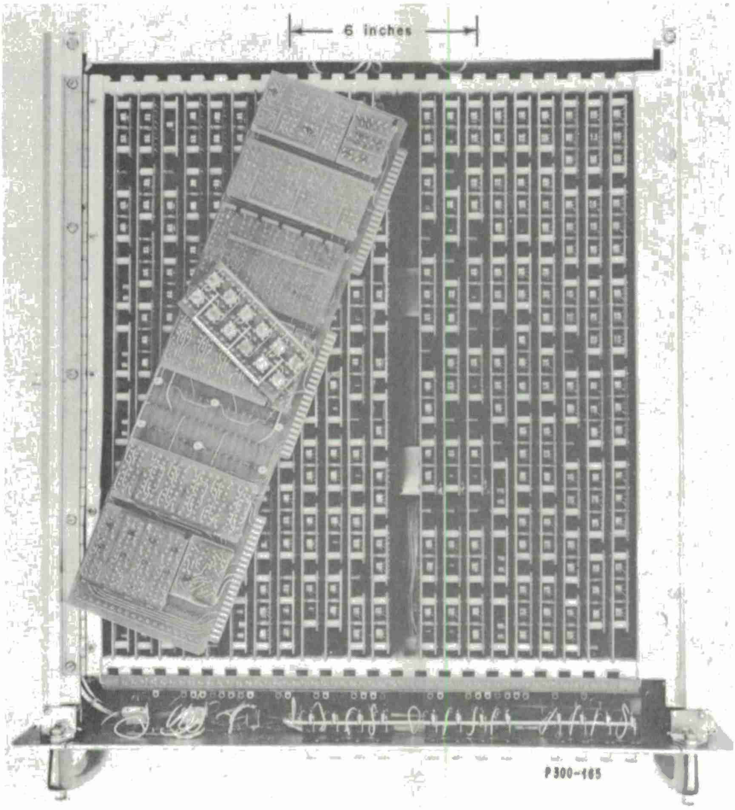


Fig. 1-13. Typical encoder/decoder circuitry.

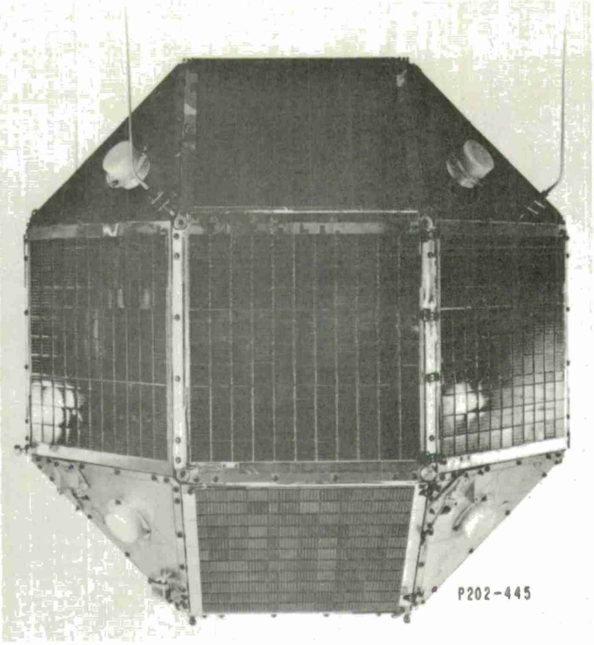


Fig. 1-14. LES-2.

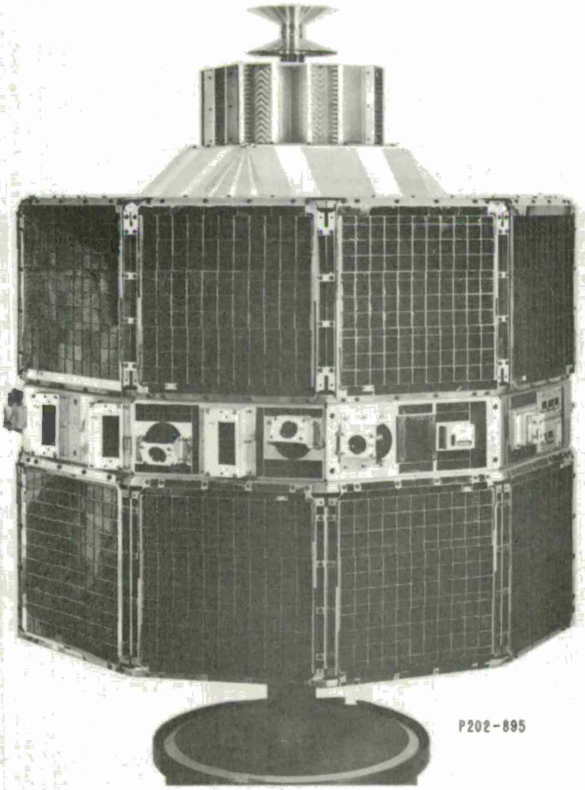


Fig. 1-15. LES-4.

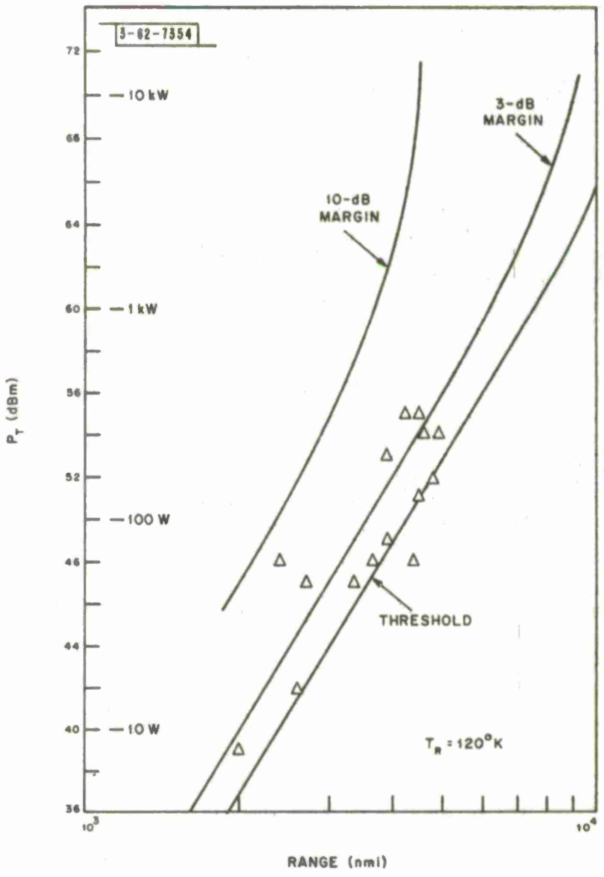


Fig. 1-16. LET up-down performance with LES-2 (10 kbits/s).

J. TEST OPERATIONS

Since the completion of the terminal in May 1965, a great many tests[10,11] have been run with the terminal (1) by itself on a "back-to-back" basis, (2) to a satellite and back to itself, and (3) via a satellite, the moon, or tropospheric scatter mode to another station with one of the identical signal processing vans. With the terminal tested "back-to-back," threshold signals corresponding to a 9-dB signal-to-noise ratio in the 5-kHz bandwidth for the 10-kbit rate and to a 6-dB signal-to-noise ratio for the 5-kbit rate were achieved. This corresponds to the predicted threshold E/N_o ratio per bit of 6 dB for the system.

Test operations have also been run from the terminal to the Lincoln Experimental Satellites[12] LES-2 and LES-4 which are shown in Figs. 1-14 and 1-15, respectively. The LES-2 satellite is in a highly elliptical orbit with a perigee of about 1500 nmi and an apogee of about 10,000 nmi. When apogee is near overhead, visibility times can be as long as four hours.

Figure 1-16 shows the expected performance of the terminal with the satellite for various ranges. Ground transmitter powers are shown at the left for various theoretical operating margins based on a threshold P_r/N_o of 46 dB required for the 10-kbit rate. At low ranges and low up-link powers, the satellite transponder is operating as a linear amplifier. As the margin is increased, especially at the greater ranges, the curves bend upward reflecting the fact that the satellite transponder is approaching saturation. Typical measured operating threshold points are shown. These points are from data taken over several months, with the threshold criterion being acceptable voice quality. Because the modulation-coding scheme used is very efficient, this threshold is quite sharp and easily detected.[1] These curves are affected by a number of satellite and ground station parameters, many of which are variable and beyond the control of the operator. For example:

(1) Satellite power may vary with temperature, age, and inexact aiming and switching of the satellite antennas toward the earth.

(2) Terminal receiving system noise temperature may vary with weather conditions and antenna elevation angle; 120°K noise temperature was used for this chart instead of the 100°K "best case" condition.

(3) Path loss may vary with weather conditions and antenna elevation angle.

(4) Terminal calibrations may vary with age.

(5) Satellite and terminal bandpass characteristics may not be flat.

When another station transmitted an interfering signal to the satellite, either as a tone or wide-band noise, no interference in the UP/DOWN transmissions was detected until the interfering signal substantially exceeded the communications signal at the satellite input. When this "power jamming" of the hard-limited satellite transponder caused the communications down-link signal to drop below receiver threshold, only then did the communications fail.

Figure 1-17 shows the expected performance for a simultaneous two-way link through the satellite. In this case, the second terminal is a fixed station with a 60-ft antenna and a receiving system noise temperature of about 300°K. To maintain equal margin at the two receivers simultaneously, the up-link transmitters must be maintained with an 8-dB transmitter power ratio; that is, 10 kW for the LET and 1.5 kW for the 60-ft terminal. Theoretical operating margins based on a threshold P_r/N_o of 46 dB are shown as a function of range. A limited number of tests have been made using LES-2 between LET and the 60-ft terminal; signal-to-noise ratio measurements have essentially agreed with the curve.

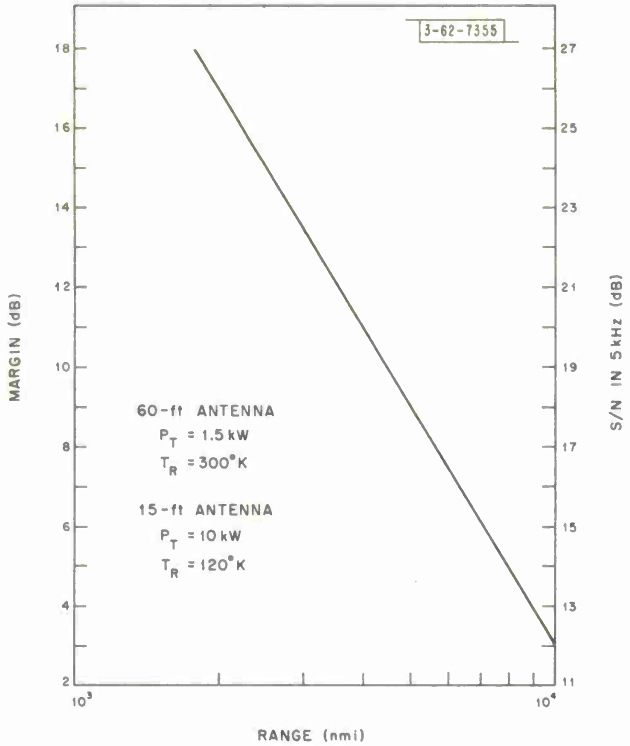


Fig. 1-17. Receiver margin, 2-way link with LES-2 (10 kbits/s).

Tests have been made varying the frequency spread while operating at the threshold of good voice performance. No difference in threshold was noted when using 2.5-, 5-, and 10-MHz hopping. When the 20-MHz spreading was used, the performance was slightly degraded because of bandpass limitations in the satellite and in the terminals.

Voice has been transmitted successfully through the terminal and satellite from standard telephone system 2-wire desk telephones within the Laboratory and via long-distance lines. Listeners to these calls have easily been able to recognize a familiar voice. Full duplex teletype has also been transmitted simultaneously with the telephone traffic.

In addition to the active satellite, the moon has been used as a passive reflector and carried simultaneous 2-way voice and teletype conversations between the LET and the 60-ft terminal, along with the minor frustrations of waiting 5 seconds for a response to a question or comment – a situation which would be quite tolerable if this were the only means of voice communication one had available.

The tropospheric scatter propagation mode also can easily support communications between terminals. Initial tests indicate that good communications can be achieved at the 10-kW power level for separation distances of over 100 miles, when the antennas are aimed at each other on the horizon. This feature, like the moon channel, might be a useful mode under certain emergency conditions.

REFERENCES

1. P.R. Drouilhet, Jr., "Signal Processing System," Chap. 2.
2. B.H. Hutchinson, S.B. Russell and J.W. Craig, "Modulation and Demodulation System," Chap. 3.
3. J.M. Wozencraft and B. Reiffen, "Sequential Decoding" (M.I.T. Press, Cambridge and John Wiley, New York, 1961).
4. I.L. Lebow and P.G. McHugh, "A Sequential Decoding Technique and Its Realization in the LET," Chap. 5.
5. I.G. Stiglitz, G. Blustein and F. Jelinek, private communications.
6. J. Tierney and J.N. Harris, "Channel Vocoder," Chap. 6.
7. F.E. Heart and W.R. Crowther, "Computer System," Chap. 4.
8. B.F. LaPage, "Lincoln Experimental Terminal Antenna System," Technical Report 404, Lincoln Laboratory, M.I.T. (4 October 1965), DDC 630702.
9. L.W. Bowles, "Parametric Amplifiers in the Lincoln Experimental Terminal," NEREM Record (November 1965).
10. K.L. Jordan, Jr., "The Performance of Sequential Decoding in Conjunction with Efficient Modulation," IEEE Trans. Commun. Tech. (June 1966).
11. B.E. Nichols and P. Rosen, "Results of Experiments with The Lincoln Experimental Terminal Using a Variety of Channel Types," NEREM Record (November 1965).
12. H. Sherman, D.C. MacLellan, R.M. Lerner and P. Waldron, "The Lincoln Experimental Satellite Program (LES-1, 2, 3, 4) A Progress Report," Conference Record of AIAA Communications Satellite Systems Conference, Washington, D.C. (2 – 4 May 1966).

CHAPTER 2
SIGNAL PROCESSING SYSTEM

P. R. Drouilhet, Jr.

ABSTRACT

A part of Lincoln Laboratory's program in satellite communications involves the development of signal processing techniques that will improve communications over satellite links; particular emphasis is placed on techniques suited to military command and control requirements. A transportable satellite communications terminal, called the Lincoln Experimental Terminal (LET), has been built to embody several recent developments; its purpose is to experiment with and demonstrate the utility of these techniques in a military communications environment.

The modulation and coding system in the LET permits operation at low received signal-to-noise ratio with both dispersive (e.g., the moon) and nondispersive (e.g., active repeater) satellite communications media. Frequency hopping of a 16-ary signaling alphabet plus sequential coding and decoding matched to this modulation format provide a high degree of interference rejection and an erasure probability of less than 10^{-3} at an E/N_0 of approximately 6 dB on a nonfading channel, 10 dB on a fading channel.

Three data rates are provided: 9600 bits/s to accommodate a voice-excited vocoder permitting "phone-patch" operation; 4800 bits/s, to accommodate a pitch-excited vocoder requiring high quality input speech; and 200 bits/s for teletype only. Two teletype channels are multiplexed with each of the higher rates.

Terminal control functions, including antenna pointing, range and Doppler prediction, and data multiplexing and demultiplexing are performed by a Univac 1218 computer.

A. INTRODUCTION

The Lincoln Laboratory conducts a continuing program of research and development in the field of communications; presently, this effort is directed primarily at techniques for improved use of satellites for military command and control applications. A part of this program has been the construction of a transportable satellite communications terminal, called the Lincoln Experimental Terminal (LET), embodying several recent developments. This terminal allows experimentation in a realistic environment with techniques which heretofore have been primarily confined to the laboratory, and demonstration of their utility for operational communication circuits.

This chapter will be devoted to a detailed description of the terminal's signal processing system. Companion chapters present an overall description of the terminal,[1*] plus detailed

*Numbered references for Chapter 2 are given on page 31.

descriptions of the vocoder,[2] the sequential encoder-decoder,[3] the hardware implementation of the modulation-demodulation system,[4] and the use of the computer in the LET system.[5]

For completeness, we present in Sec. 2-B a brief summary of the general characteristics of the terminal. We then discuss the modulation and coding structure, the bandspreading technique, the synchronization technique, and finally, present a brief discussion of the overall system performance.

B. GENERAL DESCRIPTION OF TERMINAL

The design objectives for the LET were as follows:

(1) Easy transportability, including airlift capability

(2) Operation in the military satellite communications band

(3) Ability to achieve voice communication at minimum required signal power

(4) Maintenance of communications in presence of heavy interference and/or jamming

(5) Operation over a wide variety of satellite media, including active repeaters, passive reflectors, and the moon.

The LET is primarily a speech communications terminal. In order best to satisfy the last three design objectives stated above, digital transmission is employed, using a vocoder to reduce the required bit rate. The Lincoln Laboratory developed vocoder[2] operates in one of two modes, depending on the location of the talker. In order for the vocoder to operate at its lower rate of 4800 bits/s as a pitch-excited vocoder (PEV), "high fidelity" input speech is required; this mode is used when the talker uses the high quality microphone at the terminal. On the other hand, when using a "phone patch" with the talker remote from the terminal, the bandlimiting and distortion inherent in a telephone link requires the use of the voice-excited vocoder (VEV) mode at approximately 9600 bits/s.

In both of these modes, two 100-wpm teletype channels are multiplexed with the speech transmission. A third mode, in which only the two teletype channels are transmitted, is provided for those situations when the path loss and/or interference prevents speech transmission.

The modulation and coding system described below achieves satisfactory vocoded speech at a received energy-per-bit to noise-density ratio (E_b/N_o) of 6 dB, or a P_R/N_o of 43 dB using the 4800-bit/s mode. (By comparison, a single FM voice channel requires 52 to 54 dB P_R/N_o for comparable quality.) Translating to required satellite effective radiated power (ERP), a 4800-bit/s link from a satellite at synchronous altitude to LET (15-ft antenna, 100°K receiver temperature) needs only 63 mW ERP. Thus, using the inherent multiple access capability of the modulation-coding system, 20 full duplex voice links between LET-like terminals can operate simultaneously through an 8-W ERP satellite with a 5-dB margin. Greater numbers of links can be achieved with higher satellite ERP and/or larger antennas at one end of the links.

Using the moon as a passive reflector, a satisfactory 4800-bit/s channel can be maintained between LET and another terminal, using similar terminal equipment but a 60-ft-diameter antenna. The 9600-bit/s link is marginal under these conditions.

C. MODULATION AND CODING STRUCTURE

A number of factors were involved in the choice of the modulation and coding structure. The desire for high efficiency in terms of the signal energy required to convey the information, for flexibility in the type of data to be transmitted, and for at least the potential capability of data security indicated the use of an all-digital system. Some form of bandspreading was needed to provide a high degree of immunity to interfering signals and jamming; this ability to operate in the presence of interfering signals allows the additional benefit of multiple use of the satellite repeater without a rigid network control doctrine. The requirement for operation on passive dispersive channels as well as active repeaters rules out the use of signal structures which are sensitive to multipath or which require phase stability over a sequence of signal elements. In addition, the slowly varying selective fading characteristics of at least one channel of interest (the moon) suggests that the channel signals should provide a high degree of frequency diversity.

The signal structure chosen to meet these requirements can be considered in three parts: the basic signaling element or waveform which conveys the modulation, the coding technique which provides controlled redundancy over a sequence of signaling elements, and the bandspreading technique which spreads the relatively narrow information modulation over the available RF bandwidth.

The basic channel signal is a 200-μs pulse of sine wave carrier. The frequency of each pulse is chosen from a set of sixteen frequencies, according to the information to be transmitted by the pulse. At the receiver, each of the sixteen frequencies on which the pulse might have been transmitted is examined by a channel receiver comprising a filter approximately matched to the signal waveform followed by an envelope detector. The outputs of the sixteen channels are ordered according to amplitude, and the receiver output is an ordered list of the channels containing the seven largest outputs.

Figure 2-1 shows the essential elements of the coding and modulation system. Input information is accepted into the convolutional encoder at one of two rates, one bit per 200 μs or two bits per 200 μs. The convolutional encoder adds three or two check bits, respectively, putting out four bits each 200 μs as required to select which of the sixteen frequencies is to be transmitted on the next pulse. At each rate the added parity check bits are based on the sixty most recent information input bits, i.e., the sequential encoder has a constraint length of sixty. Each channel signal element thus carries one or two information bits, depending on the rate being used.

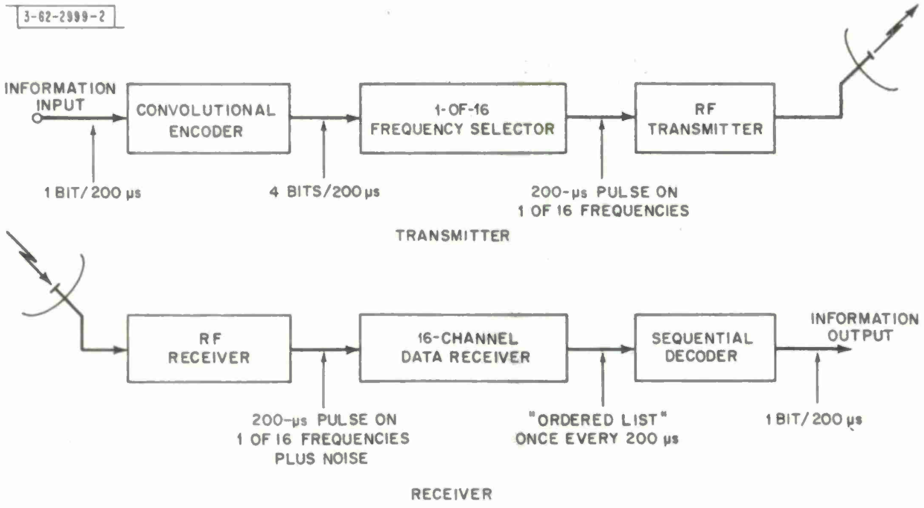


Fig. 2-1. Simplified terminal (without frequency hopping).

The third mode, in which teletype only is transmitted at an information rate of 200 bits/s, used the same coding structure as the low rate mode above; a 4-bit encoder output results from each information bit into the encoder. However, since a new information bit now appears only once each 5 ms, the encoder output remains constant for 5 ms so that (except for the frequency hopping described below) 24 successive 200-μs pulses carry the same information and thus have the same frequency (the 25th pulse in each 5-ms interval is a sync pulse as described below).

The sequential decoder consists of a 1000-word buffer memory, containing at any time the ordered list from the most recent one thousand outputs, and decoding logic operating on the contents of this memory. The rate at which the decoding logic can successfully decode the information in the buffer depends on the number of errors occurring on the channel. If the number of errors is too large, the decoding logic will not be able to keep up and information will pass through the buffer and be lost before it can be decoded; this situation is termed a buffer overflow. The percentage of information lost due to buffer overflows is the performance criterion comparable to probability of error in conventional systems (the probability of an undetected error in a sequential decoder of this type is essentially zero).

Each 200 μs, the decoder buffer receives a new ordered list from the receiver and puts out the decoded information bit or bits corresponding to the oldest list in the buffer, i.e., the list which it received one thousand pulse intervals earlier. Thus the decoder operates with a fixed delay of one thousand pulse intervals, or 200 ms, for either of the higher two rates.

The detailed operation of the sequential decoder and the trade-offs involved in the selection of its parameters are described in Chapter 5. One further point deserves mention here, however, because of its intimate interaction with the communication system timing and operating doctrine. Once a sequential decoder is unable to keep up with the incoming data, so that its input buffer overflows, both encoder and decoder must be restarted from a predetermined state; restarting may be initiated either by a feedback channel from receiver to transmitter or by periodically restarting the encoder and decoder whether or not buffer overflow has occurred.

In order to eliminate the dependence on a feedback channel and the attendant delay in restarting, we have chosen to restart the encoder and decoder periodically every one thousand channel symbols (the choice of restart period is only by coincidence the same as the buffer size). The input information sequence is divided into 975 symbol blocks, to each of which are appended a sequence of 25 zeros. The tail of 25 zeros permits adequate checking of the information bits at the end of the block while causing a rate loss of only 2-1/2 percent. When a buffer overflow occurs, the remaining bits in that block are lost and decoding recommences at the beginning of the next block. The resulting effect on the vocoded speech will be a short click.

D. BANDSPREADING AND FREQUENCY CONTROL

To provide the bandspreading required for protection against interference and jamming, the set of sixteen frequencies used for the transmission of each 200-μs pulse is varied in a pseudorandom manner. In addition, the transmitted frequency is corrected for the Doppler frequency shift on the transmitter-to-satellite link so that, within prediction error, the signals are reradiated by the satellite on their nominal frequency. This prediction is generated by a digital computer which is an integral part of the terminal.

Thus, as illustrated in Fig. 2-2, three sources combine to specify the frequency of each transmitted pulse:

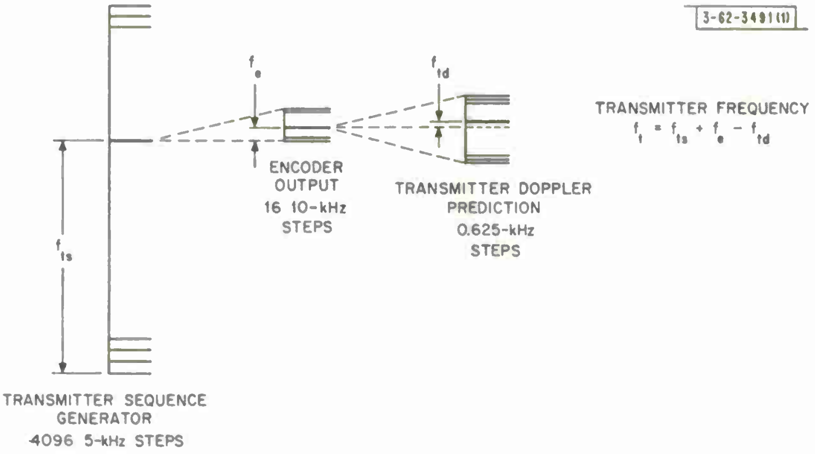

3-62-3491(1)
f_e
f_{td}
TRANSMITTER FREQUENCY
$f_t = f_{ts} + f_e - f_{td}$
ENCODER OUTPUT 16 10-kHz STEPS
TRANSMITTER DOPPLER PREDICTION 0.625-kHz STEPS
f_{ts}
TRANSMITTER SEQUENCE GENERATOR 4096 5-kHz STEPS

Fig. 2-2. Transmitter frequency generation.

(1) A new 12-bit output from the pseudorandom sequence generator specifies one of 4096 frequencies spaced 5 kHz. This frequency, designated f_{ts}, forms the base or carrier frequency for that 200-μs interval.

(2) The 4-bit output of the encoder shifts this base frequency by one of sixteen 10-kHz steps, ranging from zero, or no shift, to a maximum of 150 kHz.

(3) The 9-bit output of the Doppler predictor corrects the resulting frequency to the nearest 625 Hz, thus generating the final output frequency f_t.

Each of these digital outputs could control a separate frequency selector, with the final output frequency generated by a combination of these frequencies. A functionally equivalent but simpler procedure consists of adding the three digital control words, with appropriate offsets (Table 2-1), and using the sum to control a single frequency selector. The result of this addition is a 15-bit word, specifying one of 2^{15} = 32,768 frequencies, in which the zeroth-order bit represents a frequency increment of 625 Hz, the fourth-order bit (the lowest order in the encoder output) an increment of 10 kHz, etc. In order not to exceed the range of fifteen bits, the possible sequence generator outputs are constrained such that the sum can never exceed 2^{15}.

TABLE 2-1

TRANSMITTER FREQUENCY CONTROL WORD COMPOSITION

4-Bit Encoder Output								e_7	e_6	e_5	e_4				
12-Bit Sequence Generator	s_{14}	s_{13}	s_{12}	s_{11}	s_{10}	s_9	s_8	s_7	s_6	s_5	s_4	s_3			
9-Bit Doppler Prediction							d_8	d_7	d_6	d_5	d_4	d_3	d_2	d_1	d_0
Sum	x_{14}	x_{13}	x_{12}	x_{11}	x_{10}	x_9	x_8	x_7	x_6	x_5	x_4	x_3	x_2	x_1	x_0

The 16-channel data receiver must be retuned every 200 μs to the new set of sixteen frequencies used for that pulse. This is accomplished by stepping the receiver LO frequency in synchronism with the transmitter frequency hopping. The digital command signal for the receiver LO frequency selector is made up, as illustrated in Fig. 2-3, of the following three components:

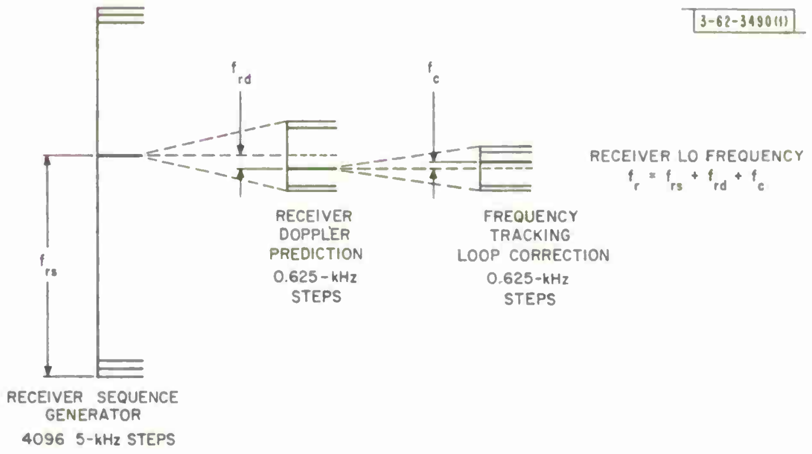

3-62-3490(1)
f_{rd}
f_c
RECEIVER LO FREQUENCY
$f_r = f_{rs} + f_{rd} + f_c$
RECEIVER DOPPLER PREDICTION
0.625-kHz STEPS
FREQUENCY TRACKING LOOP CORRECTION
0.625-kHz STEPS
f_{rs}
RECEIVER SEQUENCE GENERATOR
4096 5-kHz STEPS

Fig. 2-3. Receiver LO frequency generation.

(1) The output of the receiver's pseudorandom sequence generator.

(2) The Doppler prediction for the satellite-to-receiver link.

(3) The error signal from an AFC loop tracking the frequency of the received signal.

The receiver LO thus removes all frequency offsets except that resulting from the transmitted information; the mixer output is then the frequency of one of the sixteen channels of the data receiver.

The rapid and precise frequency stepping required both to generate the transmitted signal and to tune the receiver local oscillator is accomplished by digitally controlled frequency synthesizers[4] developed for this application. On the basis of a 15-bit digital command signal, the synthesizer generates one of 2^{15} frequencies spaced 625 Hz, giving a total bandwidth slightly exceeding 20 MHz. The synthesizer output stabilizes within approximately 5 μs following a change in the digital command.

E. TIME AND FREQUENCY SYNCHRONIZATION

From the previous discussion it is seen that the transmitted signal may be thought of as a continuous carrier whose frequency is changed in a pseudorandom manner every 200 μs; the information is superimposed, or modulated, on this carrier as a frequency shift having one of sixteen evenly spaced increments. The received signal is essentially a replica of the transmitted signal but delayed in time by the propagation delay and translated in frequency by the Doppler shift plus any frequency shift contributed by an unstable satellite oscillator. Both the path delay and the Doppler shift vary with time as the satellite moves with respect to the terminals.

In order to demodulate this signal, the receiver must not only step its LO in synchronism with the frequency steps of the received carrier, but also correct for the Doppler shift. To accomplish the former, the receiver requires a time base synchronized to the receiver signal within a small fraction of the 200-μs pulse period, i.e., within 5 to 10 μs. A sequence generator identical to the one at the transmitter operates from this time base to supply the stepping commands to the receiver LO synthesizer.

Accurate time and frequency standards combined with path delay prediction at each station provide the basis for such synchronization. Each terminal has a master clock set to Greenwich

Mean Time with an accuracy on the order of 1 ms; initial clock setting is based on broadcast time signals compensated for propagation delay. A frequency standard accurate to one part in 10^9 drives the clock counters and serves as a reference for the frequency synthesizers. An orbit computer at each terminal computes the propagation delay and Doppler shift for the path between that terminal and the satellite. The transmitter frequency and sequence generator time base are corrected by the computed Doppler and delay, respectively, so that the signal as reradiated by the satellite is nominally independent of the up-link Doppler and delay. In like manner, the receiver uses the orbit computer outputs to correct its LO frequency and time base for the Doppler and delay in the satellite-to-receiver path.

A number of sources of error combine to limit the precision of delay and Doppler correction which can be achieved from the orbit computations; chief among these are clock time uncertainty and imperfect knowledge of the orbit parameters. Therefore, closed-loop frequency and delay tracking is provided to correct the residual errors.

In order to provide a suitable signal for these tracking loops, the 4-bit information word out of the sequential encoder is suppressed (i.e., set to all zeros) on every 25th 200-μs pulse. The frequency of the resulting pulse then depends only on the pseudorandom sequence generator, and is thus known a priori to the receiver. (The information which would have been carried by this pulse is lost. However, the error correction capability of the sequential decoder fills in the gap with negligible loss in performance.)

The timing of the resulting transmitted signal is shown in Fig. 2-4. Based on computer prediction, the transmitter timing is advanced ahead of reference or satellite time by the predicted up-link delay. This results in the sync signal, and, of course, the communication signal, being radiated by the satellite at the nominal time, as if the transmitting terminal were actually at the satellite. The receiver then predicts the down-link or satellite-to-receiver delay and looks for a signal around this nominal time.

Frequency and time lock are performed on the sync pulse train in two stages: search and track. In search, a special sync receiver is tuned to the frequency of the next sync pulse 2.5 ms ahead of the nominal arrival time of that pulse. It remains on that frequency until 2.5 ms after that time. It is then retuned to the frequency of the next sync pulse. The sync receiver, shown in Fig.2-5, has three channels, one tuned to the nominal, or predicted, frequency of the sync pulse, one tuned 2.5 kHz high and one 2.5 kHz low. Taking into account the ability of a receiver channel to "see" a signal somewhat off its center frequency, this allows the sync receiver to span a frequency range of plus or minus 5 kHz around the nominal frequency. As shown in Fig. 2-6, the output of each sync receiver channel is sampled every 100 μs, resulting in fifty range or time gates on each of the three sync receiver frequencies during the 5-ms period the sync receiver is tuned for a particular pulse. The sampled output at each of these 150 range-frequency cells are continuously integrated in 150 digital registers, so that each register contains a running average of received energy at one of the 150 range-frequency points.

In Fig. 2-7, the complete sync-search grid or acquisition window is shown. Each point of this grid corresponds to one of the range-frequency registers. As soon as the number in one of these registers exceeds a pre-set threshold, the synchronization system makes a tentative decision that a signal is being received at that range-frequency offset and attempts to track on a signal around that point. Digital time and frequency tracking loops provide corrections to the nearest 5 μs and 625-Hz increment. In the event that the detection of the signal was, in fact, a false alarm, this is sensed by the synchronization receiver and the system reverts to the search mode.

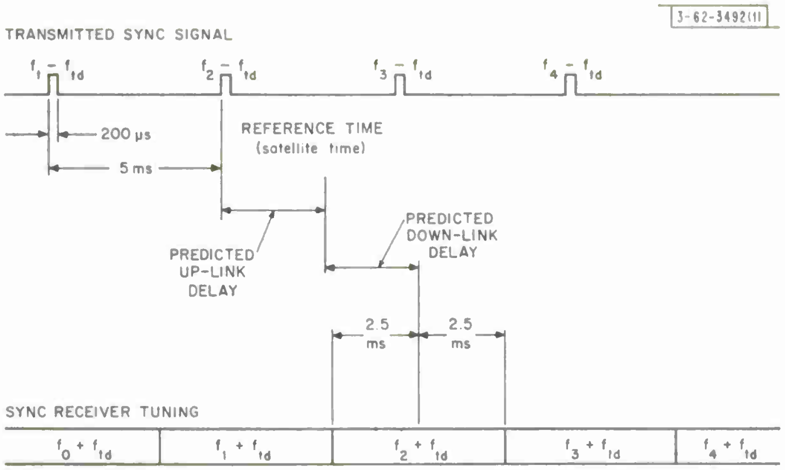


Fig. 2-4. Sync signal timing.

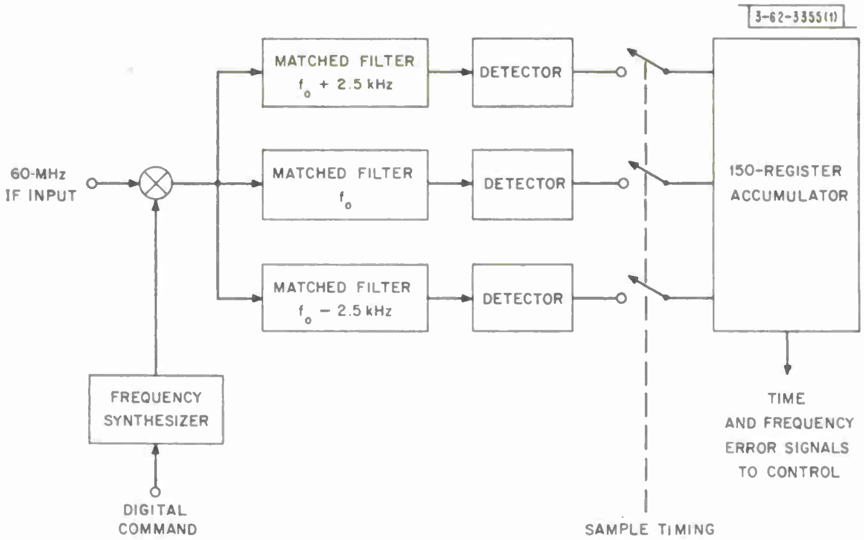


Fig. 2-5. Three-channel sync receiver.

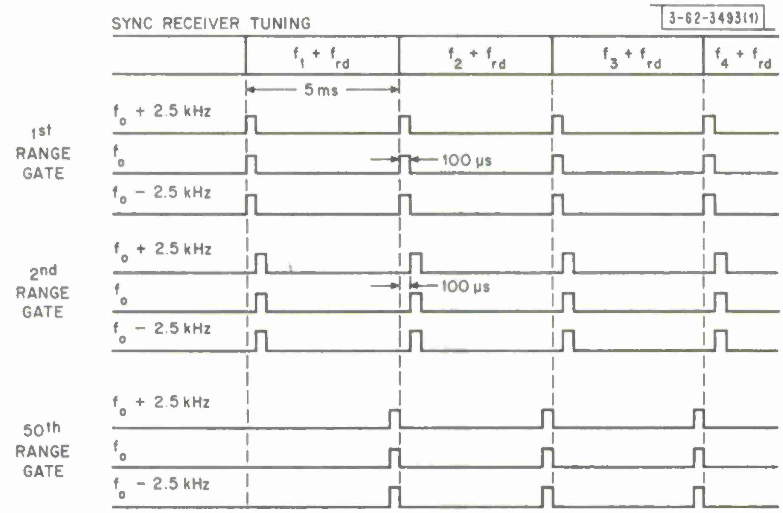


Fig. 2-6. Sync receiver timing.

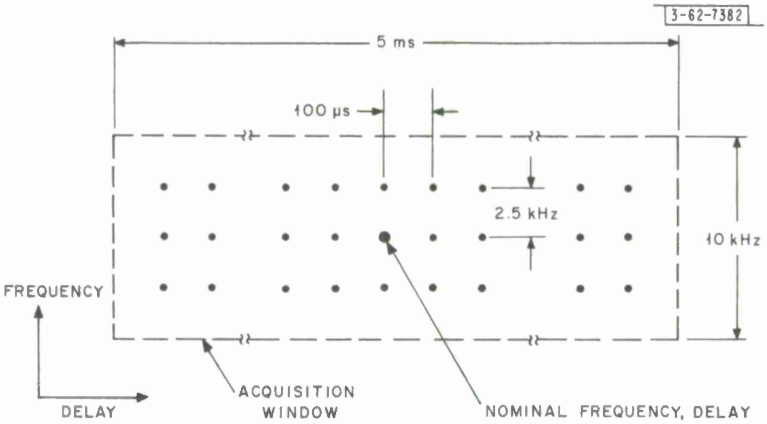


Fig. 2-7. Sync search grid.

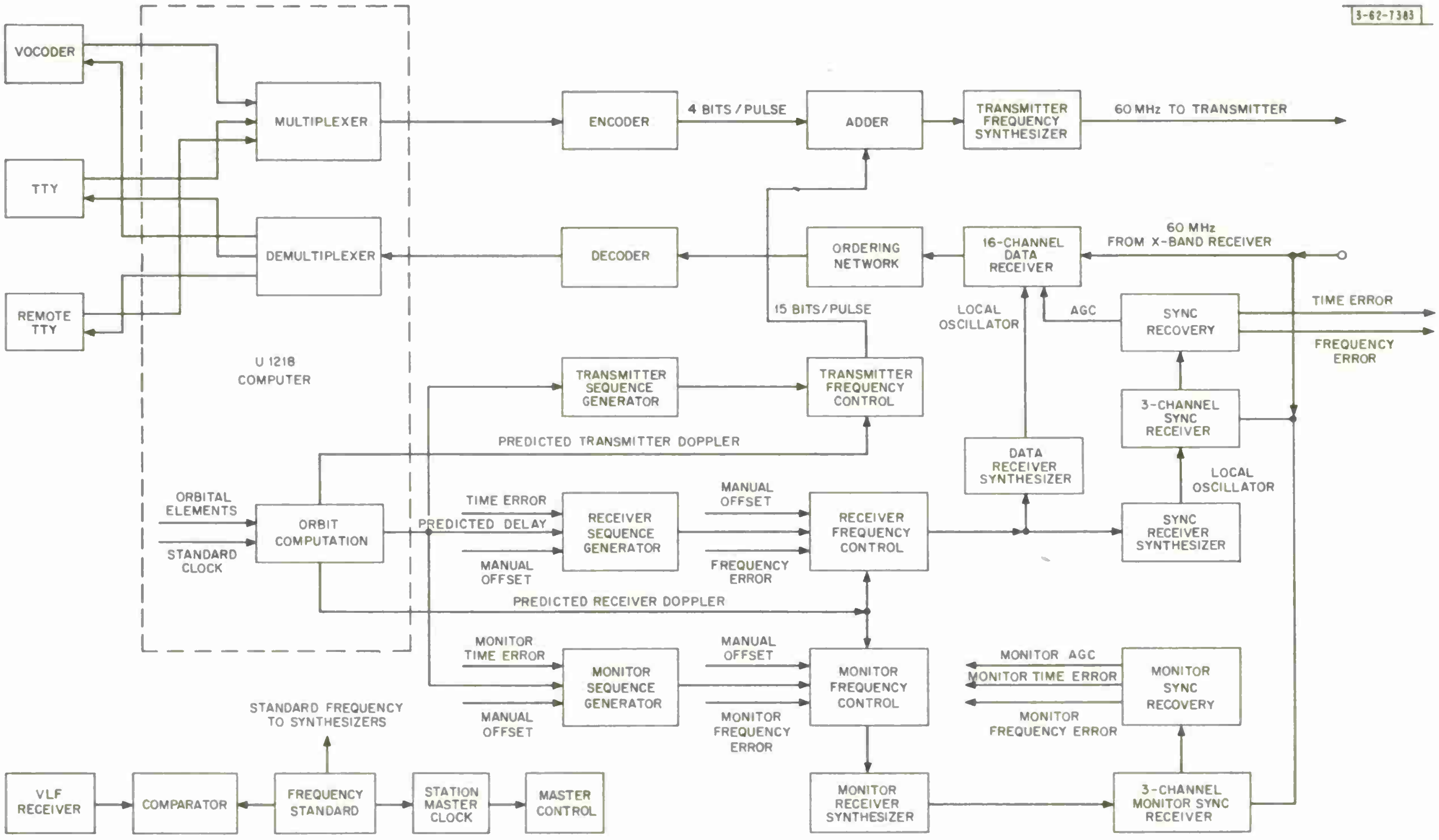


Fig. 2-8. Overall LET signal processing system.

F. SYSTEM BLOCK DIAGRAM

The overall block diagram of the LET signal processing system is shown in Fig. 2-8. It can most conveniently be described by dividing it into four major units: the data transmitter-receiver, the frequency selectors, the control unit, and the sync recovery unit.

1. Transmitter-Receiver

The transmitter-receiver system is essentially that presented in Fig. 2-1. The transmitter section contains the data sources which are multiplexed in the Univac 1218 computer and fed to the sequential encoder. The 4-bit encoder output is added to a 15-bit output from the transmitter frequency selector as described in Sec. 2-F-2; the sum specifies the appropriate output of the frequency synthesizer. The latter is generated at the 60-MHz intermediate frequency, which in turn drives the microwave transmitter.

The receiver consists of the 16-channel data receiver, ordering network, a sequential decoder, demultiplexer and data destinations. The demultiplexing is performed in the Univac 1218. The LO input to the data receiver is obtained from a frequency synthesizer, the output of which is specified by the receiver frequency selector.

2. Frequency Selector Unit

The frequency selector unit contains three essentially identical channels which develop the 15-bit numbers that control the various frequency synthesizers in the system. One such channel is required for each of the following: the transmitter, the receiver, and the so-called monitor receiver used to monitor the sync pulse transmissions of the station transmitter. The transmitter channel contains a sequence generator whose time base is adjusted according to the predicted path delay; the latter is computed by the Univac 1218 from the known satellite orbit parameters. The sequence generator output is combined with the predicted Doppler shift also generated in the computer to produce the required 15-bit number as described in Sec. 2-D.

The receiver channel is the same as the transmitter channel but with additional inputs. The sequence generator timing control has, as additional inputs, a measured time error and a manual offset. The frequency control also accepts a measured frequency error and a manual offset. Its output drives both the data receiver synthesizer and the sync receiver synthesizer.

The monitor channel is identical to the receiver channel. It drives the monitor sync receiver synthesizer.

3. Control Unit

This unit contains the time and frequency standards and the master control; these provide the accurate frequency inputs to the synthesizer and timing and control signals to the entire system.

4. Sync Recovery Unit

The sync recovery unit contains two identical channels for acquiring and tracking the sync pulse trains associated with the received and monitor signals. Each contains a 3-channel receiver (as shown in Fig. 2-5) tuned to the expected frequency and 2.5 kHz above and below. It locates the received sync pulses in time and frequency as described in Sec. 2-E and uses the measured values to adjust the time and frequency settings in the frequency selector units. AGC signals are also developed from the sync pulse train.

G. PERFORMANCE

From the outset the design of the LET signal processing system has been based on the goal of achieving and maintaining communications under adverse conditions, i.e., when the communication channel is degraded by natural and/or man-made effects. The approach to this has been twofold:

(1) Minimize the signal-energy-to-noise ratio required for acceptable quality communications

(2) Minimize the ability of interfering signals, whether by accident or by design, to disrupt communications.

The first of these assumes special importance for a satellite communications link between small ground terminals, since the already small amount of power radiated by the satellite must be divided among the various users, plus, if present, a jammer.

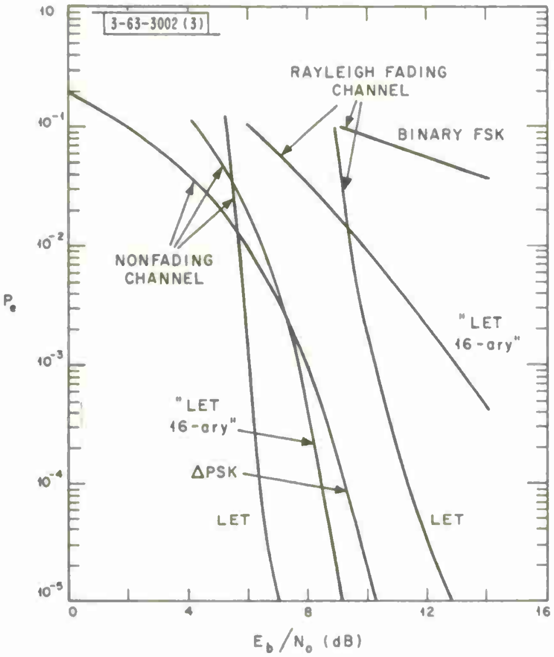


Fig. 2-9. Comparative performance of LET modulation technique.

Figure 2-9 illustrates a comparison between the signal-energy-to-noise ratio required for the LET signal processing system and that for two uncoded systems, a 16-ary alphabet system using the LET signal structure* and a conventional binary system; for the latter, differential phase-shift keying (ΔPSK) is assumed for the nondispersive channel, binary FSK for the dispersive channel.

Of course, these curves represent only the comparison of the efficiency for transmitting digits. For the transmission of speech, the combination of the vocoder and the LET digital signal processing system permits voice communications with 10 to 12 dB less signal power than is required for conventional frequency modulation techniques and, at the same time, provides the potential for security and interference rejection inherent in a digital transmission.

A more detailed discussion of the performance of the sequential decoder is given in Chapter 5.

*A structure in which a group of the same four information bits control the transmitter synthesizer output for four successive pulse intervals. These pulses are added at the receiver after detection, and a decision made on the sum (postdetection integration).

ACKNOWLEDGMENTS

The conception, design and construction of the system described in this chapter represents the efforts of a large number of people over an extended period of time. The number of people involved make it impossible to mention them individually, but their contributions are gratefully acknowledged.

REFERENCES

1. B. E. Nichols and P. Rosen, "Lincoln Experimental Terminal," Chap. 1.
2. J. Tierney and J. N. Harris, "Channel Vocoder," Chap. 6.
3. I. L. Lebow and P. G. McHugh, "A Sequential Decoding Technique and Its Realization in the LET," Chap. 5.
4. B. H. Hutchinson, S. B. Russell and J. W. Craig, "Modulation and Demodulation System," Chap. 3.
5. F. E. Heart and W. R. Crowther, "Computer System," Chap. 4.

CHAPTER 3
MODULATION AND DEMODULATION SYSTEM

B. H. Hutchinson, S. B. Russell and J. W. Craig

ABSTRACT

A bandspread modulation-demodulation system employing frequency hopping was designed for use with a sequential decoder to provide high efficiency (6 dB required E_b/N_o, nonfading) and substantial protection against multipath and interference on active or passive satellite communications channels. Three complete systems were built and demonstrated as part of the Lincoln Experimental Terminal (LET) program. Key components include: a digitally controlled frequency synthesizer capable of selecting one of 32,768 frequencies in a 20-MHz band with a worst-case switching delay of 10 μs; a bandpass crystal filter which closely approximates an ideal matched filter for the 200-μs pulsed-sinusoid channel signal used; and a 16-symbol "ordered list" signal processor of novel design. The feasibility and attractiveness of a broad class of communications systems using rapid wideband frequency hopping, multisymbol alphabets, and/or efficient demodulation matched to sequential decoding has been clearly demonstrated.

A. INTRODUCTION

The Lincoln Experimental Terminal[1,2*] uses a modulation-demodulation system designed to permit operation at low received signal-energy-to-noise-density ratio per bit (E_b/N_o) over a wide variety of satellite media including active repeaters, passive reflectors, and the moon. Frequency hopping of a 16-symbol alphabet and sequential coding and decoding[3] matched to this modulation system provide a high degree of multiple access and interference rejection capability with low error probabilities at a threshold E_b/N_o of approximately 6 dB on a nonfading channel and 10 dB on a fading channel. The realization of this system required the development of RF synthesizers capable of being rapidly frequency hopped over a wide bandwidth, filters matched to the transmitted signal to minimize the signal-to-noise ratio necessary for operation, and special analog and digital circuitry for processing of the received signals prior to decoding.

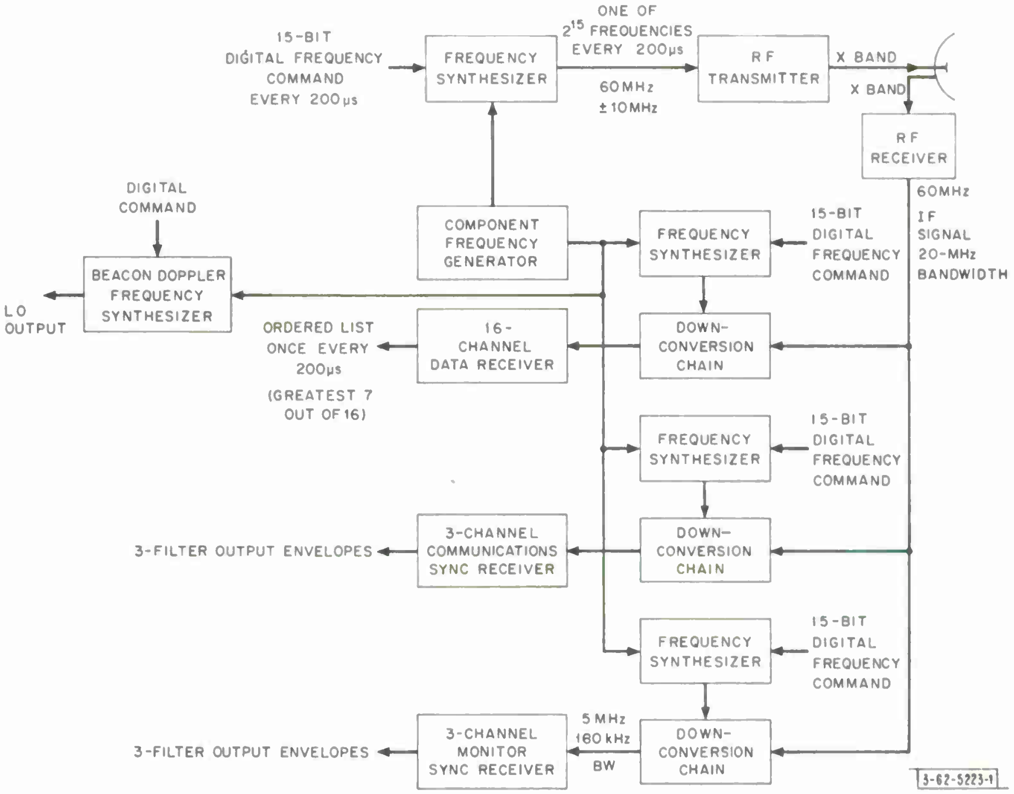
This report will describe the synthesizers, matched filters, and unusual aspects of the output processing section, with the emphasis on the development of the rapid-switching frequency synthesizers in Sec. 3-B. The channel receivers, the matched filters which form the basis of the channel receivers, and the output processing section of the data receiver are described in Secs. 3-C, 3-D and 3-E, respectively. Circuit design and variations of the techniques used in the LET modulation-demodulation system are explored in Secs. 3-F and 3-G, and the measured performance of the matched filters is covered in the Appendix.

The basic LET channel signal is a 200-μs pulse of sine-wave carrier; the frequency of each pulse is chosen from a set of sixteen frequencies spaced 10 kHz apart, according to the information to be transmitted by that pulse. In order to prevent intersymbol interference on multipath

[*] Numbered references for Chapter 3 are given on page 56.

channels and to provide protection against interference, the location of the set of sixteen frequencies within the total bandwidth of 20 MHz changes from one pulse interval to the next in a pseudorandom fashion.

Three data rates are possible with the LET system: ~10,000 bits/s, ~5000 bits/s and ~200 bits/s. The modulation-demodulation system does not change when the data rate changes from 10,000 bits/s to 5000 bits/s since only the format of the 4-bit message symbol changes. At 10,000 bits/s, 2 bits of data and 2 bits of parity comprise the symbol while at 5000 bits/s one bit of data and 3 bits of parity make up the symbol. In the 200-bit/s mode, the same 4-bit data symbol is transmitted 24 consecutive times. In all three modes, every 25th interval is a synchronization pulse, i.e., a frequency-hopped pulse which carries no information but is used as an aid in frequency and time acquisition and tracking.



15-BIT DIGITAL FREQUENCY COMMAND EVERY 200 μs
FREQUENCY SYNTHESIZER
ONE OF 2^{15} FREQUENCIES EVERY 200 μs
60 MHz ± 10 MHz
RF TRANSMITTER
X BAND
X BAND
RF RECEIVER
60 MHz IF SIGNAL 20-MHz BANDWIDTH
DIGITAL COMMAND
BEACON DOPPLER FREQUENCY SYNTHESIZER
LO OUTPUT
COMPONENT FREQUENCY GENERATOR
FREQUENCY SYNTHESIZER
15-BIT DIGITAL FREQUENCY COMMAND
ORDERED LIST ONCE EVERY 200 μs
(GREATEST 7 OUT OF 16)
16-CHANNEL DATA RECEIVER
DOWN-CONVERSION CHAIN
FREQUENCY SYNTHESIZER
15-BIT DIGITAL FREQUENCY COMMAND
3-FILTER OUTPUT ENVELOPES
3-CHANNEL COMMUNICATIONS SYNC RECEIVER
DOWN-CONVERSION CHAIN
FREQUENCY SYNTHESIZER
15-BIT DIGITAL FREQUENCY COMMAND
3-FILTER OUTPUT ENVELOPES
3-CHANNEL MONITOR SYNC RECEIVER
5 MHz 160 kHz BW
DOWN-CONVERSION CHAIN
3-62-5223-1

Fig. 3-1. LET modulation-demodulation system.

A block diagram of the LET modulation-demodulation system is shown in Fig. 3-1. Five RF synthesizers are used. The transmit synthesizer functions as a combined frequency hopper, tuned oscillator for precorrection of up-link Doppler, and 16-ary frequency-selecting modulator. Its input binary command, which changes every 200 μs, is formed as a straight digital sum of binary numbers representing pseudorandom hopping command, predicted Doppler, and live message information. Its output frequency, hopping over a band nominally centered at 60 MHz (LET-1 and LET-2) or 70 MHz (LET-3),* is added to a fixed high-accuracy X-band local oscillator in a conventional upper-sideband up-converter to form the actual channel signal which drives the transmitter power amplifier.

* LET-1 is a completely self-contained system including RF receiver and transmitter. LET-2 and LET-3 are signal processing subsystems used with existing antennas and RF equipments.

The remaining four synthesizers in LET are used as second local oscillators for four separate receiver-processors (which share a common stable fixed-frequency X-band LO and wideband first IF amplifier). Three of the receive synthesizers (communications, communications sync, and monitor sync) are internally identical, differing only in the source and format of their command signals, which are each formed as the sum of synchronized pseudorandom hopping command, predicted down-link Doppler, and AFC servo error signals. The output center frequencies are nominally 83 MHz (LET-1 and LET-2) or 274 MHz (LET-3). The fourth receive synthesizer (not used in LET-3) differs from all others in that its output bandwidth is only 1.28 MHz. Its command signal comes directly from the computer and represents only predicted Doppler for the signal from the satellite-borne beacon. This synthesizer is used merely as a computer-tuned LO to aid in initial beacon-signal acquisition; it is replaced by a VCO phase-locked to the incoming signal for normal beacon tracking following acquisition.

All the command words for the receive and transmit synthesizers are assembled in the transmit-receive control unit of the LET with the exception of the beacon Doppler synthesizer commands. The predictions of transmit and receive Doppler are generated in the computer and are sent to the transmit-receive control to be combined with the other required digital signals: the pseudorandom sequence, information to be transmitted, and time and frequency servo corrections as necessary for each synthesizer.

The communications and sync receivers extract information from the incoming signals after the wideband frequency-hopping has been removed by the receiver LO synthesizers. The two identical sync receivers each have three channels, each channel consisting of a bandpass matched filter and envelope detector. These channels, tuned to the expected received-signal frequency and 2.5 kHz above and below expected frequency, provide signals from which frequency and time synchronization are derived. The communications receiver contains 16 filter-detector channels matched to the 16 possible data frequencies (spaced 10 kHz apart). These channel outputs are processed to obtain a list of the number designations of the seven receivers with the largest output amplitudes, ordered from largest to seventh largest. The digital list information is passed to the sequential decoder. The use of an ordered list rather than a greatest-of-16 hard decision reduces the received signal-to-noise ratio requirement by several decibels.[3] The post-filter processor (lister) produces an ordered list after each 200-μs symbol has been received in the 5000- and 10,000-bit/s mode, or after the 24 repeated symbols have been added in the 200-bit/s mode (5 ms).

B. FREQUENCY SYNTHESIZERS

1. Essential Characteristics

The synthesizers used in LET each produce a single RF output signal (CW) whose frequency is uniquely determined by the state of a 15-bit parallel binary input command signal. The minimum output frequency increment, produced by a change of the least-significant bit, is 625 Hz (5/8 of 1 kHz). The output bandwidth is thus (625×10^{-6}) (32,768) = 20.48 MHz since all command states are used $(32,768 = 2^{15})$. Different units have different output-band center frequencies, depending on application. The accuracy and stability of a synthesizer's output frequency with the input command fixed is the same as that of the LET secondary frequency standard (~1 part in 10^9 worst case) to which all component frequencies are locked.

Programmable frequency synthesizers of various types are now widely available, sold as general-purpose instruments and as custom components for complex systems. Typically they are used as test equipment or as precise and quickly tuned carrier sources for tunable multichannel communications equipment. The LET synthesizer differs from these familiar types in two essential ways:

(a) It is basically designed to be controlled entirely by digital equipment as opposed to exclusive or optional manual control. As a natural result, its command format is pure binary rather than the more common BCD, and its internal organization is octal rather than decimal. Bit values are: $2^K(10\,\text{kHz})$, where $-4 < K < 10$, with resulting numbers like 10.24 MHz rather than a manually oriented 10.0 MHz.

(b) Its frequency-switching time is extremely short; after an abrupt change in the input digital command, the output frequency settles down to the new value within 10 μs for the worst case. The typical switching time is more like 5 μs. This performance is necessary in the LET application, where the switching time must be a very small fraction of the channel RF pulse length (200 μs). Theoretical calculations indicate that if the dead time on every frequency change was exactly 10 μs, the threshold of the LET demodulation system would degrade by roughly 0.5 dB.

2. Principles of Operation

The internal differences between synthesizers used in different applications are minimal. Nearly all the hardware and 9 of the 10 component frequencies supplied to each synthesizer are the same in all units. Even the beacon-Doppler synthesizer merely has some standard circuit cards removed and one special card added.

a. Basic Synthesis Scheme

The heart of a LET synthesizer is a chain of five cascaded identical modules (synthesizing stages). Each module subtracts its input frequency from one of a common set of eight fixed component frequencies (command selected), and divides the resulting frequency by eight. The frequency at the divider output is the input to the next identical module. Figure 3-2 is a block diagram of two such stages, showing the actual frequencies used in LET.

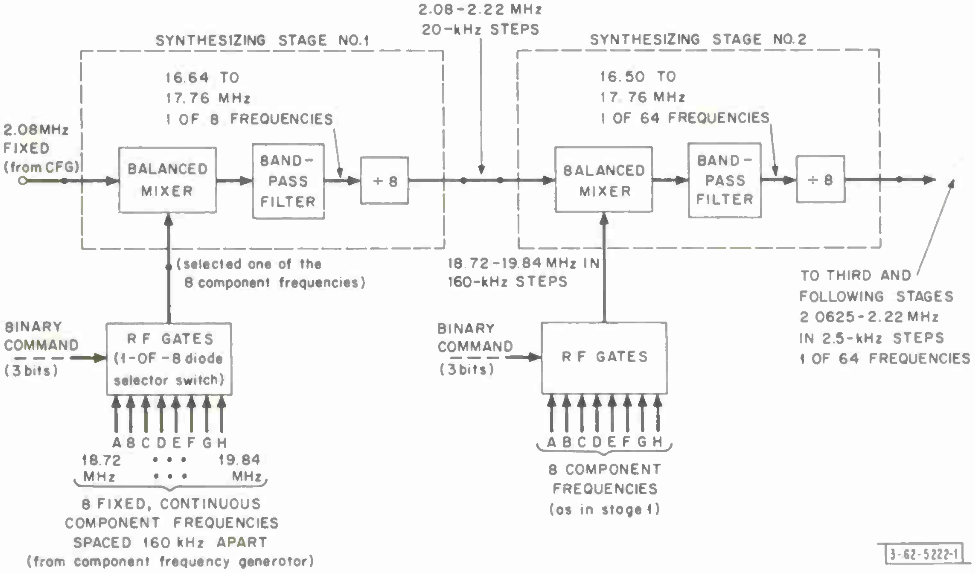


Fig. 3-2. First two stages in basic synthesizing chain of LET synthesizer.

The basic principle described was apparently discovered independently in several places, notably at the Naval Research Laboratory about 1949. This work was, however, not generally published until late 1963.[4] The idea was conceived independently by P. Drouilhet at Lincoln Laboratory in early 1962; a prototype rapid-switching synthesizer which successfully demonstrated the principle was completed in mid-1963 by B. Hutchinson. The LET synthesizer design is based on this prototype design, with slightly different frequencies and greatly improved hardware realization.

Refer again to Fig. 3-2 and consider first synthesizing stage No. 1. Three bits from the input digital command determine which one of eight diode-transistor RF switches is turned on, selecting one of the common set of eight continuous component frequencies, which are adjacent harmonics of 160 kHz. A fixed input frequency of 2.08 MHz (1/9 of the lowest component frequency) is subtracted from the selected component frequency in the balanced mixer. The bandpass filter, a conventional 5-pole Chebyshev design, passes only the lower sideband, rejecting all other mixer outputs. The frequency division is done by a 3-stage straight binary ripple-through counter which uses commercial high-speed integrated-circuit flip-flops. The output of the first stage is one of eight possible frequencies spaced 20 kHz apart, all close to the "starting" input frequency of 2.08 MHz. This permits the signal to be applied to the input of another identical stage.

The eight 20-kHz frequency steps from the first stage interpolate between the 160-kHz steps obtained by component-frequency selection in the second stage. The signal at the output of the second-stage bandpass filter can thus take on one of 64 frequencies, evenly spaced at 20-kHz intervals, about a center frequency of roughly 17.1 MHz. Division by eight produces the stage output, with 2.5-kHz frequency steps centered near 2.1 MHz. It should be evident that the output of a third stage would have $8^3 = 512$ possible frequencies, evenly spaced at intervals of 312.5 Hz.

The process of adding stages can continue indefinitely; the number of different frequencies after N stages is 8^N. In LET, $N = 5$, giving $8^5 = 32,768$ frequencies. The output of the chain is taken from the bandpass filter of the last synthesizing stage, which has no scale-of-8 divider. Since the bandwidth is essentially independent of the number of stages, spacing between adjacent frequencies decreases exponentially as more stages are added to obtain more frequencies.

b. Switching-Time Considerations

Note that the worst-case frequency switching time T_S of a chain of N stages is given by

$$T_S = T_G + NT_M$$

where

T_G = switching time of the component-frequency selection gates (one set of 8)

T_M = envelope delay of one module, measured from balanced-mixer input to frequency-divider output.

This relationship follows from the fact that, while all the component-selection switches (RF gates) switch at once, a frequency change must at worst propagate through the entire cascaded chain of synthesizing stages.

It was shown above that the number of output frequencies increases exponentially with number of stages N, while frequency-switching time increases only linearly with N, output bandwidth being invariant. This class of synthesizer is thus seen to be very useful for applications such as frequency-hopping communication systems which require very rapid switching among a large number of densely packed possible frequencies.

The switching time of the RF gates (T_G) and the delay through the frequency divider can be made very small. Achieving high speed in the RF switches is mostly a matter of using some sort of balanced circuit configuration with wisely chosen impedance levels which has inherently high isolation between the RF signal path and the switching command signal. Switching transients are thus kept out of the RF signal paths, and there is no need for heavy RC decoupling in the switching command circuits with accompanying loss of speed. Presently available diodes and transistors permit the realization of fractional-microsecond RF gates without inordinate effort. Frequency dividers consisting of flip-flop counters have a frequency envelope-delay time of less than one cycle of their output frequency (about 2.1 MHz in LET). Such counters capable of handling input frequencies of tens of megahertz are now realizable, using monolithic integrated circuits. This can be extended to hundreds of megahertz using discrete-component or hybrid multichip techniques.

The primary limitation on the switching time of this type of synthesizer is thus the envelope delay in the bandpass filter, which is, in general, inversely proportional to bandwidth. Study of the block diagram in Fig. 3-2 readily reveals that for a given synthesizer bandwidth, there is a minimum realizable filter center frequency below which mixer input frequencies and the unwanted output sideband cannot be rejected by the filter. An expression for this minimum frequency is easily derived, demonstrating that it is linearly proportional to bandwidth with other factors fixed. A limitation of the synthesis scheme is thus revealed: bandwidth expansion to reduce switching time requires all circuits to operate at a proportionately higher center frequency. From another angle, the maximum percent bandwidth obtainable directly from a synthesizing chain (i.e., without subsequent down-conversion) is somewhat limited, and is given roughly by 100/division ratio.

c. Choice of Parameters in LET Synthesizers

As stated earlier, the synthesizer output bandwidth required for LET is 20.48 MHz. However, the 10-μs switching time requirement can be met with synthesizing-stage filter bandwidths which are considerably narrower, on the order of 1 or 2 MHz. Synthesizing stages of this bandwidth can be built at a much lower center frequency than could stages with 20-MHz bandwidth, as explained above. Thus, the basic chain of stages in the LET synthesizer has an output bandwidth of 20.48/16 = 1.28 MHz, centered at about 17.14 MHz (7.5-percent bandwidth). The output bandwidth of 20.48 MHz is obtained by passing the output of the basic chain through a X16 frequency multiplier, whose output center frequency is roughly 274.2 MHz. The frequency increments are, of course, also multiplied by 16; the minimum increment at the output of the basic synthesizing chain is actually 625/16 = 39.0625 Hz. The overall synthesizer block diagram showing these features is shown in Fig. 3-3. The full-bandwidth output of the frequency multiplier is mixed with a stable fixed frequency f_s given by

$$f_s = (274.2\ \text{MHz}) - (\text{final output center frequency}).$$

Synthesizers with different applications requiring different center frequencies (60, 70, or 83 MHz) differ basically only in the value of f_s, with two minor exceptions. In LET-3 (70-MHz first IF), the receiver LO synthesizer outputs are taken directly from the X16 multiplier output with no final mixing step, since the second IF is roughly 204 MHz. In the beacon-Doppler synthesizer, the X16 multiplier and the first synthesizing stage are removed. An f_s of 23.0 MHz

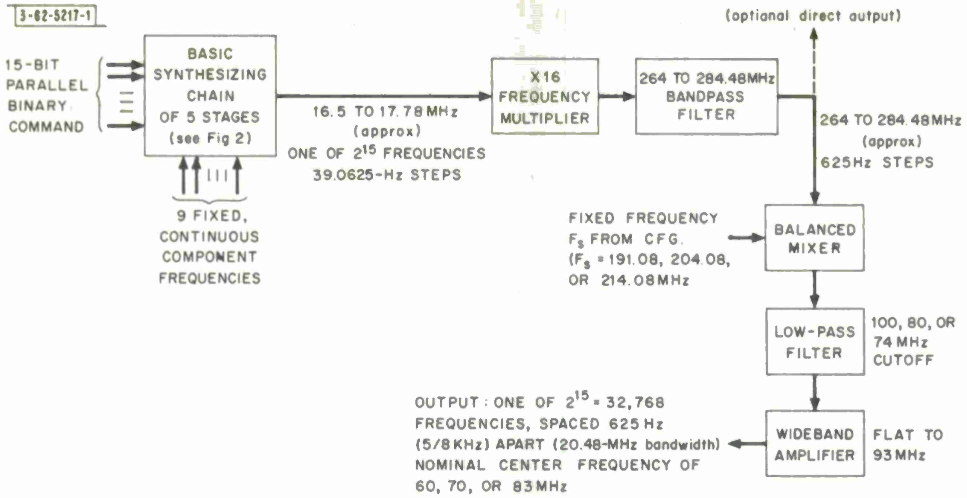

3-62-5217-1
(optional direct output)
15-BIT PARALLEL BINARY COMMAND
BASIC SYNTHESIZING CHAIN OF 5 STAGES (see Fig 2)
16.5 TO 17.78 MHz (approx)
ONE OF 2^{15} FREQUENCIES
39.0625-Hz STEPS
X16 FREQUENCY MULTIPLIER
264 TO 284.48MHz BANDPASS FILTER
264 TO 284.48MHz (approx)
625Hz STEPS
9 FIXED, CONTINUOUS COMPONENT FREQUENCIES
FIXED FREQUENCY F_s FROM CFG (F_s = 191.08, 204.08, OR 214.08MHz
BALANCED MIXER
LOW-PASS FILTER
100, 80, OR 74 MHz CUTOFF
OUTPUT: ONE OF 2^{15} = 32,768 FREQUENCIES, SPACED 625 Hz (5/8 KHz) APART (20.48-MHz bandwidth) NOMINAL CENTER FREQUENCY OF 60, 70, OR 83MHz
WIDEBAND AMPLIFIER
FLAT TO 93MHz

Fig. 3-3. Overall LET frequency synthesizer.

mixes with the 17.14 ± 0.64 MHz output from the basic synthesizing chain to produce an output at 40.14 ± 0.64 MHz.

Use of the X16 multiplier following the basic synthesizing chain allows the latter to be constructed with maximum frequencies under 20 MHz; direct synthesis of the full 20-MHz output band would require the synthesizing chain to be built in the 200- to 300-MHz region. The advantages in cost, simplicity, reliability, compactness, and power consumption of the lower-frequency circuits are, of course, quite significant. The basic LET system characteristic that makes the multiplier configuration possible is the large ratio of total frequency-hopped bandwidth (20.48 MHz) to the basic channel signal bandwidth (5 kHz); the switching time can thus be long compared with the reciprocal of the total bandwidth. This will be true in many frequency-hopped communications systems.

Design and construction of the frequency multiplier required some care and effort, but far less than alternative approaches not using the multiplier. Primary considerations were achieving a reliable drive and temperature-independent flat frequency response (within 1 dB) over the required 7.5-percent bandwidth while maintaining 50 dB or better suppression of unwanted harmonics. These objectives were achieved by using a cascade of four doublers. The first three doublers are identical in circuit configuration to a conventional full-wave rectifier with center-tapped transformer, except that the DC output is not used and the double-frequency "ripple" constitutes the desired output. This type of doubler is inefficient (7-dB loss with ideal diodes) but potentially very "clean" since the input frequency and its undesired third harmonic are suppressed to a degree limited only by the circuit balance achieved. The first two doublers are followed by 2-pole bandpass filters; these further suppress unwanted fundamental and third harmonic, but also suppress the sizable fourth-harmonic component present in a rectified sine wave. The latter produces odd-order components in the output of the next full-wave doubler in the chain if not suppressed. The third doubler drives the final (conventional class-C) doubler directly. The entire chain is followed by a steep-skirted 5-pole tubular coaxial filter of standard design centered at 274 MHz.

There is one outstanding inherent drawback to the use of a frequency multiplier following the synthesis chain: the degree of suppression of spurious sideband voltages is degraded by exactly the factor of multiplication, because of the multiplied index of the phase modulation produced by the unwanted sidebands. (The effect is worse if the frequency multiplier does not limit the amplitude modulation caused by spurious outputs.) Spurious outputs are 65 dB down or better at the output of the basic synthesizing chain; at the X16 multiplier output, this is reduced to 40- or 41-dB spurious suppression (voltage factor of 16 = 24 dB). This degree of suppression is more than adequate for LET and similar applications. Nearly all the spurious outputs at or near the worst-case suppression of 40 dB are spaced from the output frequency by multiples of 160 kHz, so that they always fall outside the band occupied by the bank of receiver filters. Also, synthesizer spurious outputs are nearly always well below receiver thermal noise when the system is used on a satellite link. Even when several frequency-hopped signals of differing amplitudes occupy the same satellite repeater bandwidth, interference from synthesizer spurious outputs is negligible compared to interference from intermodulation products generated in the satellite limiter.

There is a third alternative besides direct full-band synthesis or use of a multiplier: the final synthesizing stage could select from a special set of component frequencies whose spacing equals the full predivider bandwidth of the preceding stages. In the LET, for example, a one-of-16 selection of frequencies spaced 1.28 MHz apart could have been used in the last stage, or else a one-of-8 selection with the main synthesizing chain output expanded to 2.56-MHz bandwidth. The potential advantage of this approach is that the spurious suppression in the final output need be no worse than that of the basic synthesizing chain. The main disadvantage is a significant increase in complexity over that of the multiplier, especially if the full potential improvement in spurious rejection is to be achieved. At least 8 new component frequencies, "clean" down to −60 dB or so, would have to be generated and selected at VHF frequencies where such spectral purity can be rather difficult to achieve. Providing for a number of different output center frequencies while dodging spurious mixing products could be quite complicated. These considerations were decisive in rejecting this approach in favor of the multiplier for LET. It might be useful in some other application requiring wide bandwidth plus very high spectral purity.

d. Component Frequency Generator

A block diagram of the component-frequency generator (CFG) is shown in Fig. 3-4. Its only input is a 1-MHz signal from the station secondary frequency standard. Its useful output consists of twelve different fixed, continuous component frequencies required by the synthesizers plus one used by the receiver down-conversion chains (28 MHz). As mentioned earlier, nine component frequencies (those shown in Fig. 3-2) are used by all synthesizers; the final mixing frequency f_s is selected from the remaining three CFG outputs according to synthesizer function. Spurious outputs are 60 to 65 dB down from desired output frequencies (worst case).

All CFG output frequencies are locked to the reference-standard input signal. This provides a tremendous operational advantage in a terminal like LET; there are no oscillators in the CFG whose frequencies must be periodically checked and reset, and no equipment warmup period is necessary. The entire synthesizer-receiver complex is ready to operate as soon as power is applied. The accuracy of the terminal's frequency standard is of necessity very well maintained to achieve necessary real-time clock accuracy (one terminal uses a rubidium standard) and batteries adequate to operate the standard and clock for nearly a week without prime

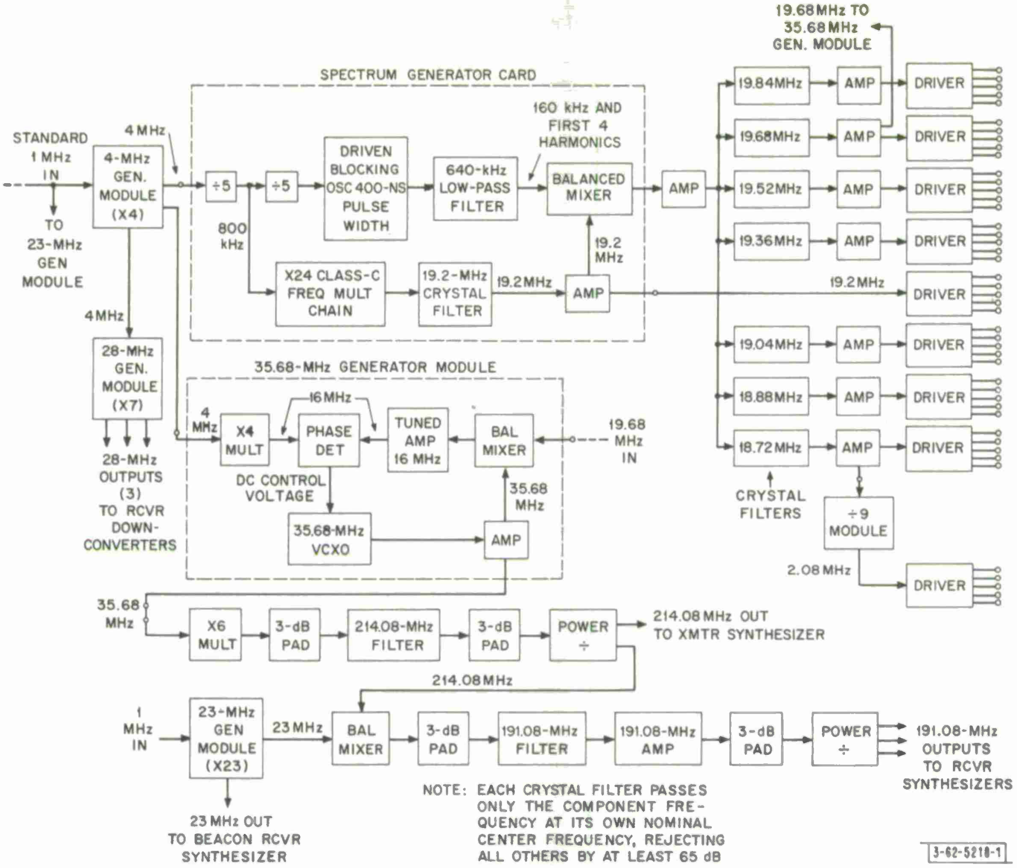


Fig. 3-4. Component frequency generator for LET frequency synthesizers and receivers (60-MHz IF).

power are included. The only temperature-control ovens in the entire LET system are those in the main and backup frequency standards. This contributes greatly to reliability and freedom from RFI as well as the instant-warmup mentioned above.

Principles of the CFG design and operation should be clear from the block diagram. The basic component-frequency spacing of 160 kHz is derived as (1 MHz) (4/25); all outputs are harmonics of either 1 MHz or 160 kHz.

A pulse generator running at 160 kHz drives a sharp-cutoff low-pass filter, producing as output the 160-kHz fundamental and 2d through 4th harmonics with nearly equal amplitudes. This low-pass line spectrum is mixed with 19.2 MHz to produce a double-sideband bandpass spectrum with 8 equal amplitude lines. Seven of the frequencies in this spectrum (plus the independently derived 19.2 MHz) are picked out by crystal filters to make up the basic set of 8 component frequencies. Other required CFG outputs are derived from this basic set and from 1 MHz in various straightforward ways clear from the diagram.

Three basic considerations influence the conceptual design of a component-frequency generator of this type:

(1) In designing the complete system, frequencies to be generated should be chosen at the outset (where choice is possible) so as to be derivable as simple multiples, submultiples, or integer ratios of the standard frequency or of another required frequency. Where no choice or only limited choice of an output frequency is possible, time spent in examining many possible combinations of existing frequencies can often pay off by revealing a much simpler generating scheme than the first "obvious"

approaches thought of. An example in LET is the generation of the required 214.08 MHz as

$$214.08 = 6\,[19.68 + 4(4)]$$

where 19.68 and 4 MHz were already available. It should also be remarked that internal or output frequencies which come out as continuing fractions are a nuisance which can usually be avoided or minimized by care in initial system-design choices.

(2) The possibility of spurious outputs arising from inherently imperfect frequency multipliers, dividers, and mixers plus unfortunate choice of frequencies must be constantly kept in mind from the very beginning. If this is not done, the result can be nasty surprises plus the need for a "panic" redesign and retrofit of expensive, state-of-the-art filters to remove by brute force spurious outputs which could have been avoided by proper design choices at the outset.

(3) Stray coupling can very easily ruin completely the performance of an excellent conceptual CFG design, no matter how generous the filter stop-band-rejection margins. This is especially true if really good spurious-output rejection (>60 dB) is sought. The trouble will, of course, never reveal itself until a complete system or subsystem is assembled, isolated circuits on a bench test naturally being immune. Strict attention must be paid to layout, shielding, decoupling, and isolation from the very beginning.

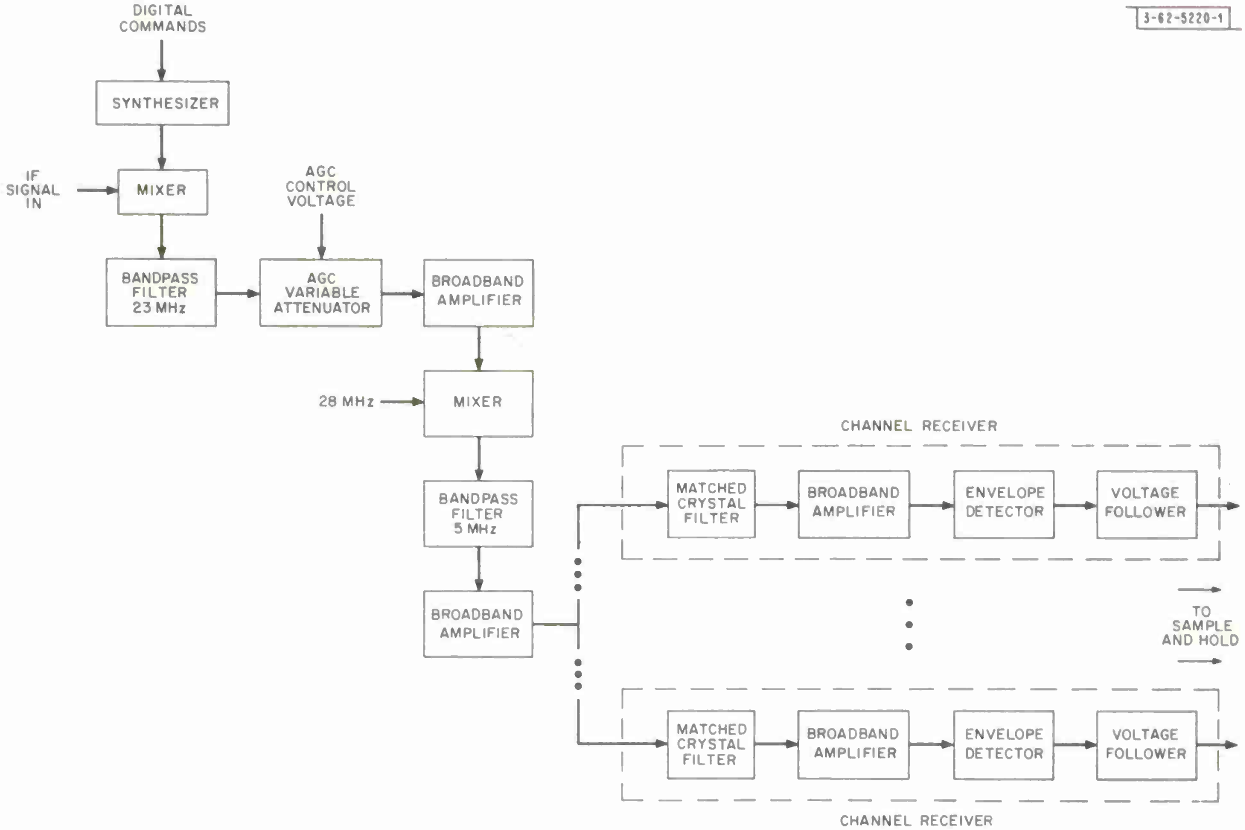


Fig. 3-5. Channel receivers.

C. LET CHANNEL RECEIVERS

The LET receivers have two features of significance which contribute to their simplicity and small physical size. These are the crystal matched filters which are matched to the 200-μs burst of transmitted energy, and the output processing section which uses a combination of analog and digital techniques to provide the ordered list of receiver designations on which the sequential decoder operates. This section deals with the channel receivers which utilize the matched filters to achieve their desired performance. Section 3-D discusses the matched filters in more detail and the output processing section of the LET data receiver is described in Sec. 3-E.

A signal processing frequency of 5 MHz was chosen in order to allow the matched filters to be built to the required specifications and to keep circuit design problems minimal. The received signal is heterodyned from a first IF frequency of 60 to 5 MHz in two steps as shown in Fig. 3-5. The pseudo-random frequency hopping is removed in the first mixer, whose local-oscillator frequency is hopped over a 20 MHz bandwidth always 23 MHz higher than the incoming signal. The signal (whose bandwidth is now reduced to ~160 kHz corresponding to the transmission of one of the 16-data frequencies) is then mixed to 5 MHz, with the desired bandwidth and selectivity obtained by the passive filters shown in Fig. 3-5. A voltage controlled attenuator provides AGC action based on signal and noise measurements made in the sync recovery section.

The 5-MHz signal drives the bank of channel receivers, each consisting of a crystal matched filter, a gain-stable wideband amplifier and envelope detector, and an operational amplifier connected as a voltage follower to isolate the detector from the subsequent processing circuits.

The data receiver has sixteen channel receivers, each one using a filter matched to one of the sixteen possible data frequencies which are separated by 10 kHz. The sync receivers each have only three channel receivers which are matched to the nominal sync frequency and to frequencies 2.5 kHz above and below the nominal. The outputs of the sync channel receivers are used directly by the sync recovery unit while the data channel receiver outputs are operated upon by the output processing section of the data receiver to generate an ordered list of channel designators. Particular emphasis was placed on gain stability as a function of time and gain tracking of the channel receivers as a function of temperature since the decision of which symbol was transmitted is made by comparisons of the sixteen channel receiver outputs. The channel receivers have a useful input signal range of more than 40 dB.

D. LET MATCHED FILTERS

A matched filter is used in the LET channel receiver because it is the optimum predetection filter in the sense that it maximizes the ratio of the peak instantaneous signal to rms noise. The LET matched filters are not exactly matched to the signal waveforms, but they are very good approximations. The derivation of the approximate transfer function is discussed in this section and the test setup and procedure for the measurement of the performance of the resultant crystal matched filter realizations is described in the Appendix. A summary of the test results shows the very good performance of these filters. First, however, a very brief summary of the elementary properties of matched filters for a signal in white noise is given. For a complete discussion see Turin.[5]

1. Matched Filters

For a known signal s(t) of finite duration in additive white noise, the matched filter has an impulse response h(t) that is the delayed time reversal of the signal, $h(t) = s(\tau - t)$, where τ is large enough so that $h(t) = 0$ for $t < 0$, and thus the matched filter is realizable. Then the response $f_o(t)$ of the matched filter to $s(t)$ is

$$f_o(t) = \int_{-\infty}^{\infty} h(x)\, s(t-x)\, dx = \int_{o}^{\infty} s(\tau - x)\, s(t-x)\, dx \quad , \tag{3-1}$$

which shows that the time variation of the output is the autocorrelation function of the signal, and it is maximum for $t = \tau$. The transfer function $H(j\omega)$ of the matched filter is then

$$H(j\omega) = S^*(j\omega)\, e^{-j\omega\tau} \quad , \tag{3-2}$$

where $S^*(j\omega)$ is the complex conjugate of the Fourier transform of $s(t)$. This states that the transfer function of a matched filter is equal to the complex conjugate of the Fourier transform of the signal multiplied by the delay factor $e^{-j\omega\tau}$. Another significant property of a matched filter is the maximum signal-to-noise ratio (defined in the above sense) at its output

$$\left(\frac{\text{peak signal}}{\text{rms noise}}\right)^2 = \frac{E}{N_o} \tag{3-3}$$

where $E = \int_o^T s^2(t)\, dt$ = signal energy and N_o is the noise power spectral density in watts per Hertz or joules for $-\infty < f < \infty$. For a proof of these properties and a complete discussion, see Ref. 5.

2. LET Matched Filters

In each channel of the LET receiver the noise has an essentially flat spectrum, so the impulse response of the appropriate matched filter is the time reverse of the signal, which is a 200-μs burst of sine wave of known frequency f_o but of unknown phase. Thus the signal can be written as an envelope $e(t)$ times $\sin(\omega_o t + \varphi)$ where $e(t)$ is the rectangular pulse $u(t) - u(t-T)$, $u(t)$ is the unit step and $T = 200\,\mu s$. For $f_o T >> 1$, the signal energy is concentrated in a narrow band of frequencies centered on f_o, and a straightforward analysis shows that the envelope of the response to the signal $e(t)\sin(\omega_o t + \varphi)$ of a filter matched to $e(t)\sin\omega_o t$ is essentially independent of φ. In fact, the fractional change in the peak value of the envelope is less than $(4\pi f_o T)^{-1}$ for any value of φ. So in the narrow-band situation, the filter can be matched to a pulse of any convenient phase without a degradation in performance. Furthermore, this allows us to approximate a matched filter for the signal $e(t)\sin\omega_o t$ by the impulse response $g(t)\sin\omega_o t$, where $g(t)$ is a satisfactory approximation to $e(\tau - t)$. That is, $g(t)$ is the impulse response of a filter approximately matched to the low-pass signal $e(t)$.

Then the response of the approximate bandpass matched filter to the signal is

$$f_o(t) = \int_o^t s(x)\, h(t-x)\, dx = \int_o^t e(x)\, g(t-x) \sin\omega_o x \sin\omega_o(t-x)\, dx$$

$$= -\frac{1}{2}\cos\omega_o t \int_o^t e(x)\, g(t-x)\, dx + \frac{1}{2}\int_o^t e(x)\, g(t-x) \cos\omega_o(t-2x)\, dx \quad . \tag{3-4}$$

For $f_o T >> 1$, the second term is very small and

$$f_o(t) = -\frac{1}{2}\cos\omega_o t \int_o^t e(x)\, g(t-x)\, dx \quad . \tag{3-5}$$

This shows that the envelope of $f_o(t)$ is proportional to the response of the approximate low-pass matched filter to the signal e(t), and thus the quality of the bandpass approximation is equal to that of the low-pass approximation, thereby simplifying the approximation problem. Note that the above argument is valid for any e(t) that is slowly varying compared with $\sin\omega_o t$.

Let the Laplace transform of g(t) be G(s), then the transfer function of the approximate bandpass matched filter is

$$H(s) = \frac{1}{2j} [G(s - j\omega_o) - G(s + j\omega_o)] \quad . \tag{3-6}$$

The poles of H(s) are those of G(s) shifted by $+j\omega_o$ and by $-j\omega_o$, and for $f_oT >> 1$, the zeros of H(s) are (to a very good approximation[†]) those of G(s) shifted by $+j\omega_o$ and by $-j\omega_o$. So the poles and zeros of H(s) are easily obtained from those of G(s).

The final step in the solution of the approximation problem is the determination of the rational function G(s). Previous work[‡] in this area had shown that the best[§] 3-pole, 2-zero approximation performs 0.236 dB below the ideal, so this G(s) was decided upon. However, it was desired to realize the filter with quartz crystals which (in the present state of the art) requires zeros on the $j\omega$ axis (the zeros in the above approximation are slightly off the axis). So by a computer aided search the best 3-pole, 2-(imaginary) zero transfer function,

$$G(s) = \frac{s^2 + \gamma^2}{(s + \sigma)\,[(s + \alpha)^2 + \beta^2]} \quad , \tag{3-7}$$

was found to have the following singularities,

$$\gamma T = 6.122 \text{ radians}$$
$$\sigma T = 2.097$$
$$\alpha T = 1.296$$
$$\beta T = 4.696$$

and its performance is 0.265 dB below the ideal.[¶] For T = 1 sec the unit impulse and matched pulse responses are shown in Figs. 3-6 and 3-7. The frequency response is shown in Fig. 3-8. Quartz crystal filters realizing the corresponding bandpass transfer function for $f_o \approx 5$ MHz and T = 200 μs were designed and constructed by Damon Engineering, Incorporated, under contract to Lincoln Laboratory. The setup for testing these filters and their measured performance are discussed in the Appendix.

[†] For the G(s) used in the LET filters, the error as a fraction of the bandwidth is approximately $1/4\, f_oT$.

[‡] For an experimental system called Wescom [Division 3 Quarterly Progress Report, Lincoln Laboratory, M.I.T. (15 August 1963), DDC 417499].

[§] In the sense of largest peak-signal-to-rms-noise ratio.

[¶] Identical results have been found independently by P.A. Meyer who has also found the best approximations for transfer functions with up to 11 poles and 10 zeros (with unrestricted zeros as well as zeros restricted to the $j\omega$ axis). See "Über Filter mit angenähert rechteckförmiger Impulsantwort," NTZ, 18, 249-255, 568-569 (1965).

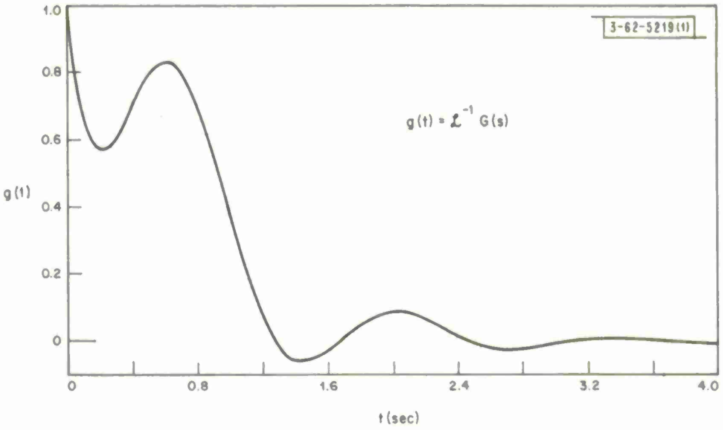


Fig. 3-6. Impulse response of 3-pole, 2-(imaginary) zero matched filter.

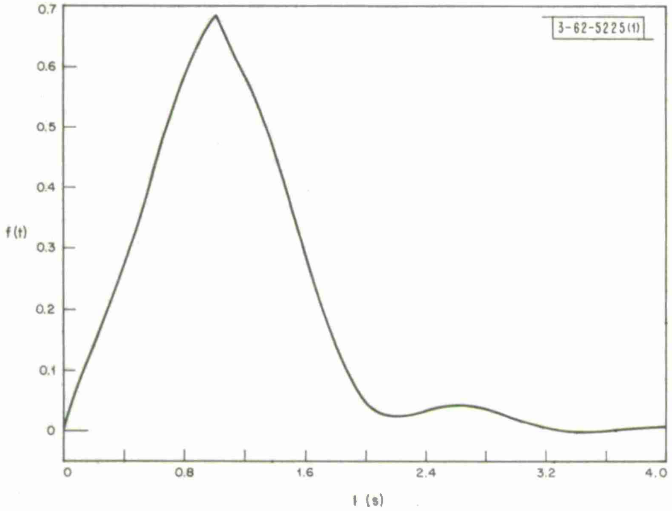


Fig. 3-7. Response of 3-pole, 2-(imaginary) zero matched filter to a matched (1 second) pulse.

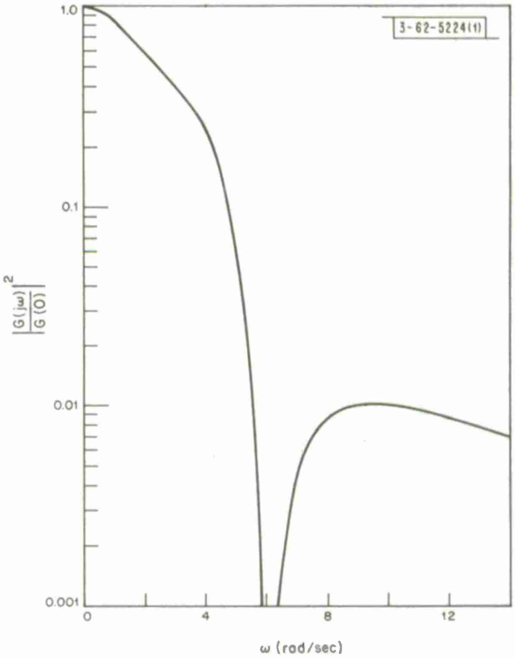


Fig. 3-8. Amplitude characteristic of approximate matched filter.

E. OUTPUT PROCESSING

The outputs of the data channel receivers are the sixteen envelopes of the matched filter responses. The output processing section of the LET data receiver shown in Fig. 3-9 lists the addresses of the seven highest amplitude matched filter responses in decreasing order based on a sample of the sixteen channels taken at the end of the 200-μs transmitted burst interval. The ordered list of the seven most probable transmitted symbols is determined every 200μs in either the 10,000- or 5000-bit/s mode, and once every 5 ms (twenty-four repeated data symbols) in the 200-bit/s mode.

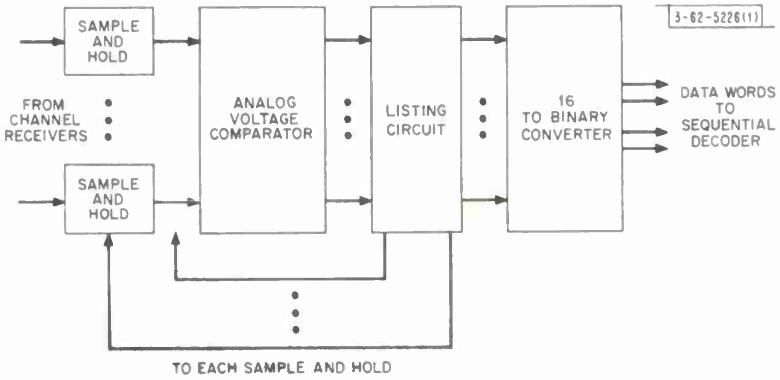


Fig. 3-9. Output processing section.

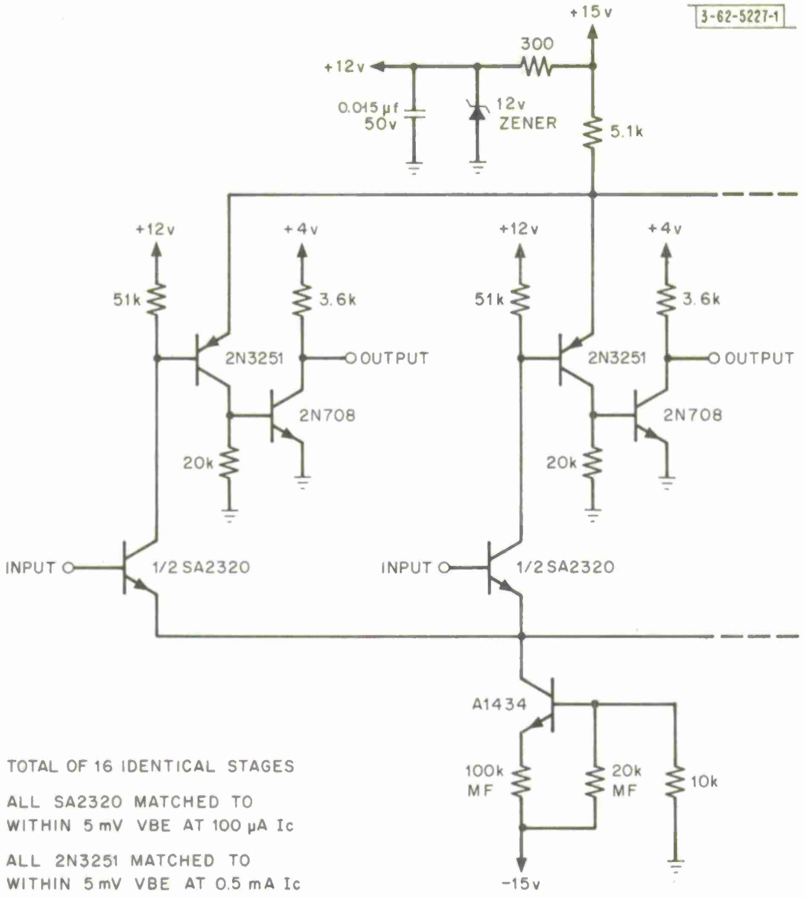
1. Analog Comparator

The function of the analog comparator is to determine which of its sixteen input voltages is the largest (most positive) and to register this decision by the presence of a logic level on the one of sixteen output lines that corresponds to the input line with the highest voltage. The comparator, shown in Fig. 3-10, consists of sixteen transistors, which are matched for base-to-emitter voltage to within 5 mv, sharing a single emitter current source. The voltage of this common emitter follows the most positive of the sixteen input voltages, which causes that transistor which corresponds to the most positive signal to conduct while the remaining transistors are cut off. A second similar stage of reverse polarity is used to minimize the region of indecision and this stage drives a set of logic-level switches. In order to achieve a hysteresis of approximately 5 mV over a 5-V input signal range at various temperatures, the first-stage transistors share a single heat sink. The comparator responds to an input in less than 5μs.

2. Listing Circuit

The listing circuit operates on the sixteen outputs of the comparator to decode the one-of-sixteen lines to a 4-bit address, and also controls the inputs to the comparator to cause an ordered list to be made. One channel of the sixteen channels of the listing circuit is shown in Fig. 3-11.

Initially the inhibit storage is empty. When the amplitude of the sampled signal on this particular channel is the most positive of the sixteen inputs to the comparator, the data flip-flop will be set. The first exclusive OR selects the first flip-flop that has been set as the valid data channel and ignores any of the following flip-flops in the chain that might have also been set because of comparator indecision. This single output signal is decoded to a 4-bit binary address



TOTAL OF 16 IDENTICAL STAGES

ALL SA2320 MATCHED TO WITHIN 5 mV VBE AT 100 µA Ic

ALL 2N3251 MATCHED TO WITHIN 5 mV VBE AT 0.5 mA Ic

Fig. 3-10. Analog comparator.

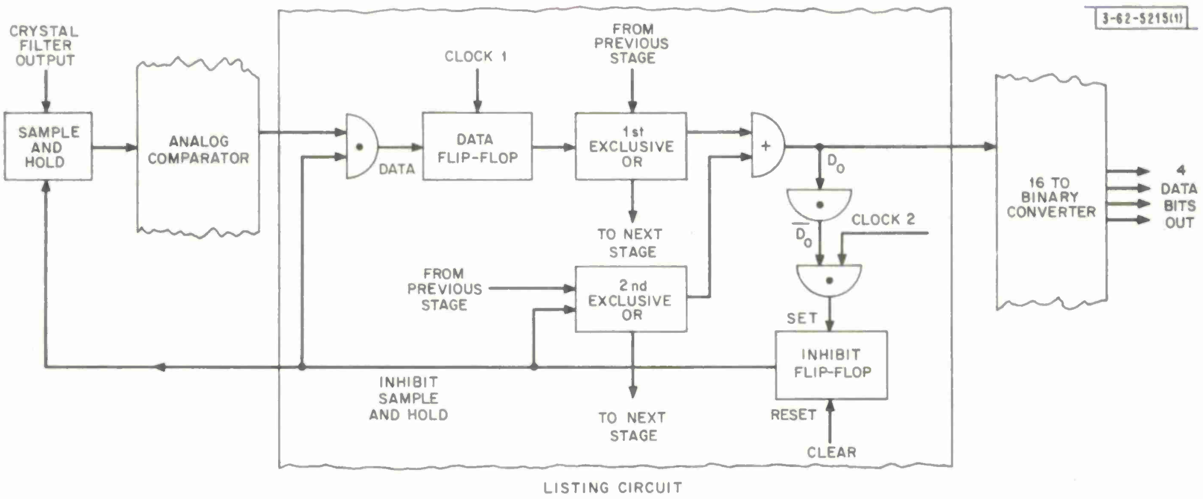


Fig. 3-11. One stage of listing circuit.

in the 16-ary-to-binary converter and sent to the sequential decoder. At the same time, the inhibit flip-flop corresponding to the selected output channel is set, removing that channel's output from the comparator input and disabling that data flip-flop input. The comparator's output then represents the second-highest analog voltage out of the 16 inputs and that channel data flip-flop is set while the previously set data flip-flop is cleared. The new channel address is decoded and its inhibit flip-flop is set without clearing the previously set inhibit flip-flop. This cycle repeats until the complete ordered list of seven 4-bit data words has been sent to the decoder, at which time the inhibit storage flip-flops are cleared. If at any time no channel is selected, the output of the 16th channel 1st exclusive OR is used to select one of the previously unselected channels using the 2d exclusive OR circuit chain. This prevents duplications in the list, which upset the sequential decoder ("no selection" would otherwise produce an output of 0000).

F. CIRCUIT DESIGN

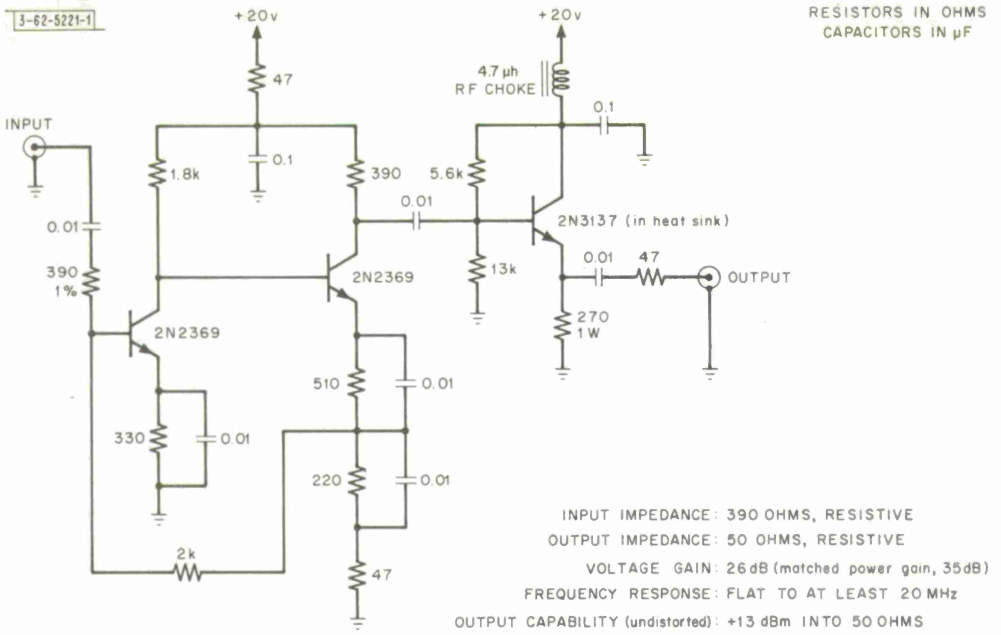
1. General Approach

In designing circuitry for the LET, emphasis was placed on reliability and stability of performance and on elimination of manual controls and adjustments wherever possible. The latter goal was to a large degree achieved. There are no front-panel operating controls for either synthesizer, receiver, or CFG, and none of the "back of the panel" controls requires periodic adjustment, being used only during initial setup or after infrequent parts replacement. Field experience has shown that the former goal was also reasonably well met; nearly all failures have been caused by either faulty interconnections or else seemingly random catastrophic parts failures.

In pursuit of the desired objectives, most circuits were designed with considerable signal-level margins, and flip-flop counter dividers were operated well below their maximum frequency. The DC voltage supplies (standard modular units) are regulated to ± 0.5 percent, though nearly all circuits will tolerate much wider variations. The design approaches followed sometimes resulted in extra complexity for the sake of predictability and performance margin; often, however, the same desire for reliability and stability of performance led to the choice of very simple circuitry. Some effort was made to achieve reasonable compactness within the limited development time available to match the compactness of the associated integrated-circuit digital equipment. Power consumption was very definitely a secondary consideration; however, an effort was made to eliminate easily avoided waste of power and to keep heat dissipation from becoming a problem.

To help achieve predictable temperature and voltage-independent performance, critical narrow-band filters and tuning elements are where possible lumped together as self-contained passive networks isolated from the active gain stages. Most amplifiers are wide-band RC types designed to have predictable resistive input and output impedances and liberal negative feedback to achieve stable, predictable gain. An amplifier circuit typical of those used is shown in Fig. 3-12. Similar circuits were used at frequencies up to 140 MHz in a noncritical application and up to 93 MHz in a fairly demanding slot (synthesizer final output).

In all circuits above 140 MHz (up to 284 MHz) and in a few circuits between 33 and 140 MHz, it was not practical to use the wide-band RC-amplifier lumped-filter technique. In such cases, more or less conventional tuned-amplifier techniques were employed, using low-Q noncritical tuned circuits with resistive swamping and padding to minimize effects of transistor variations.



INPUT IMPEDANCE: 390 OHMS, RESISTIVE
OUTPUT IMPEDANCE: 50 OHMS, RESISTIVE
VOLTAGE GAIN: 26 dB (matched power gain, 35 dB)
FREQUENCY RESPONSE: FLAT TO AT LEAST 20 MHz
OUTPUT CAPABILITY (undistorted): +13 dBm INTO 50 OHMS

Fig. 3-12. Typical wide-band RC feedback amplifier used in LET synthesizers.

For real frequency selectivity in this region primary reliance was placed on standard commercially available tubular coaxial filters.

Two techniques used in the CFG deserve passing comment. Counters using commercial (MECL) high-speed flip-flops were used very successfully for all frequency-division requirements (one scale-of-9 and two scale-of-5 dividers). In spots where a very "clean" frequency multiplier or multiplier-adder was required, a simple voltage-controlled crystal oscillator phase-locked to the desired harmonic of the input signal was employed with excellent results. The technique was especially useful in achieving relatively high prime frequency-multiplication factors such as 7 and 23. Use of a simple crystal-controlled rather than LC phase-locked oscillator precludes locking to the wrong harmonic and permits a very simple, narrow-band phase-locked-loop design because of its good short-term stability.

In all frequency ranges, frequency conversion and mixing operations were performed exclusively by broad-band double-balanced diode ring mixers. Suppression of the high-level mixer input frequency in these circuits was 20 to 40 dB relative to the output sum and difference frequencies, the poorer suppression being generally associated with VHF mixers. This performance was achieved without special diode matching; the diodes, however, were of very high quality, with very consistent forward characteristics and very short recovery time. At drive frequencies below 35 MHz, silicon diodes with less that one nanosecond recovery time were used exclusively; they also worked reasonably well up to 214 MHz, but hot-carrier diodes gave better performance in critical VHF applications. The transformers used in all mixers were wound on small ferrite toroids which had been plated with a double-layer copper eddy-current shield.[6] This technique provides the desired combination of excellent balance and very low interwinding capacitance while maintaining tight transformer coupling. The same diodes and transformer designs were used in the "full-wave rectifier" frequency doublers described earlier.

2. Construction

Most of the LET synthesizer, receiver, and CFG circuits were constructed by using commercially available miniature discrete components mounted on conventional printed-circuit boards with a ground plane on one side. All semiconductors were silicon. Standard integrated circuits were used for portions which were completely digital: frequency-divider counters, binary-to-octal command conversion logic, and receiver output processing logic.

Each separate LET synthesizer is housed in a drawer which occupies 3-1/2 inches of panel space in a standard 19-inch relay-rack cabinet (see Chapter 1, Fig. 1-11). Primary AC power requirements are about 50 watts. The CFG occupies a drawer requiring 5-1/4 inches of panel space, and can supply necessary CW component frequencies to five or fewer synthesizers. It requires about 80 watts of primary power. The LET receivers occupy three 5-1/4 × 19-inch sliding drawers, including power supplies. One drawer contains the communications down-conversion chain and 16 channel receivers. A second drawer contains the rest of the communications receiver (analog comparator and output processing). The third drawer contains both sync receivers. Figure 3-13 is a top view of the 16 matched filters mounted on their individual channel receiver cards. The digital portions of the system used integrated circuit elements mounted on small printed circuit boards which were attached to large mother boards on which were most of the interconnections. The mother boards were connected by under-chassis wiring. Each receiver drawer requires about 50 W of primary power.

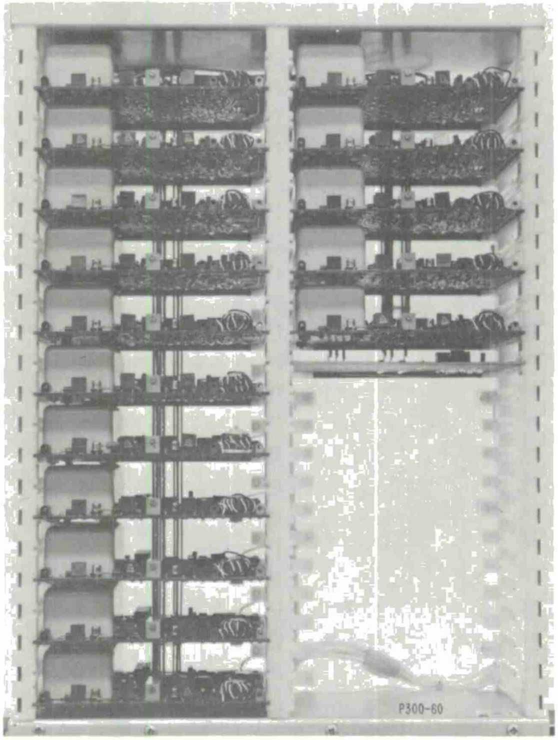


Fig. 3-13. Top view of data matched crystal filter bank.

G. VARIATIONS OF THE TECHNIQUE FOR OTHER APPLICATIONS

1. Synthesizers

The experience of building and operating the LET has demonstrated that the frequency synthesis technique used is a very flexible and useful one and is not especially difficult to realize in practical equipment with the performance required for LET. Clearly, synthesizers with less stringent requirements, particularly narrower bandwidth, would be even easier and less costly to build. The feasibility of frequency-hopped communications systems with bandwidths of 20 MHz or less and hopping rates of 5 kHz or less has thus been clearly demonstrated. It is also clear, however, that improvements in these two key parameters and/or secondary ones such as spurious-output suppression and power consumption are well within the present state of the art.

a. Performance Improvements

The maximum hopping rate, in particular, could be greatly increased with a reasonable increase in size and power consumption over the LET synthesizers merely by increasing the synthesizing-stage bandwidth. As discussed in Sec. 3-B-2-b, this implies an increase in the frequency of operation of the circuits. The components are certainly available or could easily be developed to build synthesizing stages in the 100- to 300-MHz region, implying bandwidths of 5 to 20 MHz and total synthesizer switching times of 1 or 2 μs. It is thus reasonable to say that hopping rates on the order of 50 kHz are within easy reach of a nominal development effort.

Output bandwidths beyond 20 MHz in a rapid-switching synthesizer are certainly practical in at least some applications, particularly if spurious-suppression requirements are nominal. Again, the limiting factors would tend to be size, weight, cost, and power consumption rather than inherent impossibility; these problems will, of course, decline rapidly in severity as circuit techniques advance.

b. Alternative Basic Configurations

The LET synthesizing stages select the lower sideband from the mixer output, resulting in frequency subtraction in each stage. Consequently, output frequency increases with increasing component frequency in odd-numbered stages and decreases with increasing component frequency in even-numbered stages. Command bits are inverted in even-numbered stages so that the output frequency always increases when any externally supplied command bit changes from 0 to 1.

The synthesizer was constructed this way because the in-band spurious mixer outputs are of higher order (and presumably lower amplitude) as compared with an arrangement in which the component frequencies are below the filter passband (upper sideband selection, frequency addition). Experience has shown, however, that the advantage of lower sideband selection is somewhat academic; with a well-designed mixer, the high-order spurious products will generally be negligible in either case. Lower-sideband selection caused no serious problems in the LET synthesizers. However, it is recommended that upper-sideband selection (frequency addition) be used in future synthesizing-stage designs; the command-bit inverters and associated jumper strapping are eliminated, and the internal frequencies come out simpler and easier to remember or to derive mentally. (The lowest-possible bandpass filter output frequency is the same in all stages if frequencies are added rather than subtracted.)

In a system which includes only one or two synthesizers rather than LET's four or five, total hardware size and/or cost might be reduced by using a one-of-4 rather than a one-of-8 component-frequency selection. The complexity of the component-frequency generator is reduced by slightly less than half, while the required number of synthesizing stages is increased by $\log_4(8) = 1.5$ for a given total number of frequencies. The synthesizing stages are slightly simpler, and they (and the component-frequency generator) can be built at a lower frequency for a given bandwidth. A one-of-8 selection makes more sense for a multisynthesizer installation where one CFG serves many synthesizers.

Use of a one-of-2 or one-of-16 selection seems to offer few if any advantages. Many prominent spurious mixer outputs, including a strong third-order product not canceled in a balanced mixer, fall in-band with a one-of-2 selection. Use of a one-of-16 selection requires a very complex CFG and a very high center frequency for a given bandwidth.

Many instrument-type synthesizers with cascaded identical stages employ variations of a synthesizing stage design using two mixers, two filters, and an extra fixed common component frequency. Such schemes basically permit very good spurious-output suppression to be achieved at the cost of increased switching delay and equipment complexity. For the LET application, it was desirable to push the tradeoff in the other direction, so the single-mixer stage design was used. The same considerations will apply in many frequency-hopped communications systems.

2. Receivers

The performance of the matched crystal filters described above has shown that good approximations to ideal matched filters are within the present state of the art. These filters may be used to build receivers which use large orthogonal signal sets yet are physically smaller than similar receivers using conventional filter techniques. In instances where suitable passive bandpass matched filters are not readily realizable because the pulse lengths are short compared with the center frequencies of the filters (bandwidth-to-center-frequency ratio $> 1/10$), a quadrature detector and low-pass matched filter scheme might be used to advantage.

The techniques used in the LET to provide an ordered list of the channel addresses may be applied to N channels with a list length L where both N and L can be large or small without causing much additional complexity. These considerations make the use of a frequency-hopped communications system using a multifrequency data alphabet quite attractive.

APPENDIX
MEASUREMENT OF MATCHED-FILTER PERFORMANCE

The performance of an approximate matched filter is determined by the peak-signal-to-rms-noise ratio at its output compared with that of the exact matched filter. The S/N ratio of the approximate filter is determined by measuring separately its response to the matched signal and to a wide band of Gaussian noise having constant spectral density in this band. Figure 3-14 shows a block diagram of the apparatus used for these measurements. Measurements of both signal and noise into and out of the filter (at points A and B) provide the necessary information.† The S/N ratio of the exact matched filter is easily calculated by Eq. (3-3).

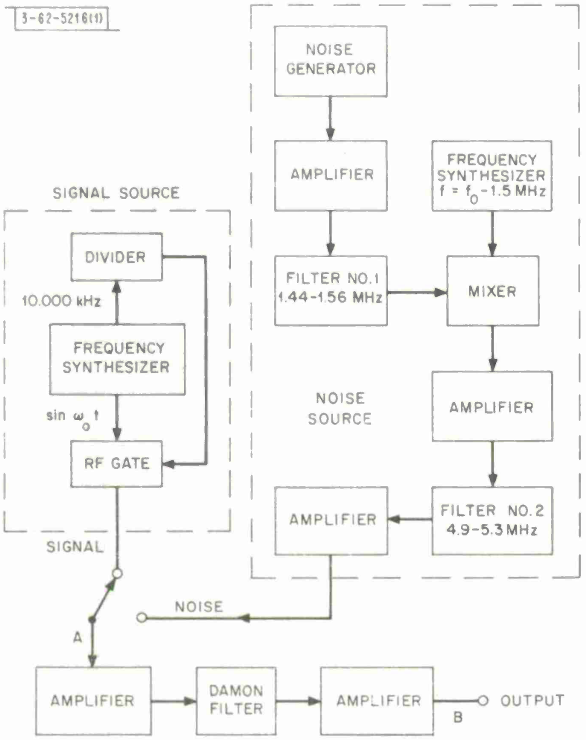


Fig. 3-14. Matched filter test setup.

The signal source in Fig. 3-14 provides a 200-μs burst of sine wave of frequency f_o (the center frequency of the filter) every 3.2 ms. The divider output opens and closes the gate, and the closed gate attenuates the signal approximately 60 dB. The repetition rate of the RF pulses is small enough so that the response of the filter decays completely between pulses. Thus, the output at B is a repetition of the response of the filter to its matched signal. The measurements required are the peak values of the RF pulse at A and the response at B. These were measured directly by observing the signals on the 6 cm screen of an oscilloscope. The input level was kept constant and the amplifier gains were adjusted so that the output peak level was approximately equal to that of the input, thereby eliminating errors due to the attenuator miscalibration in the oscilloscope. By adjusting the peak signal deflection to be nearly full-screen, errors in reading the peak value are minimized.

† The amplifiers before and after the filter have constant gain and are wide band so they do not affect the S/N ratio.

The noise performance of the filter is measured in an analogous way, but here we measure the rms value of the noise at points A and B using a Keithley Model 121 true rms, wideband voltmeter which has an accuracy of 1 percent of full scale. The noise at A must have a spectral density that is flat over a band of frequencies wide compared with the bandwidth of the matched filter (5 kHz), and it must be centered on f_o. The necessary bandwidth of the noise at A can be determined by observing that for frequencies far removed from f_o, $|H(j\omega)|$ is proportional to $1/|\omega - \omega_o|$. So, for a finite band of noise applied at A, the error in the measurement at B due to the neglect of these tails of $|H(j\omega)|$ can be found. Thus, for an error of 1 percent, a bandwidth of 102 kHz is required. The noise was kept flat to within 0.1 dB. A noise with these characteristics was generated as is shown in the block labeled noise source in Fig. 3-14.

The spectrum of the noise generator is flat to less than 0.1 dB from about 1.3 to 1.6 MHz. Filter 1 is a 5-pole Chebyshev bandpass filter with a bandwidth of 120 kHz, a nominal passband ripple of 0.03 dB, and a center frequency of 1.5 MHz. It determines the spectral shape of the noise applied to the matched filter and along with filter 2, insures that it is flat over 120 kHz. Note that since filter 1 has sharp cutoff and its passband lies well within the flat portion of the spectrum of the noise generator, the bandwidth of the noise from the generator is immaterial. If the generator produced white noise of the same spectral density, the output of filter 1 would be unchanged. The frequency of the synthesizer is set to $f_o - 1.5$ MHz, and the sum-frequency output of the mixer is at f_o MHz and passes through filter 2 which rejects other mixing products. Filter 2 is a 4-pole Chebyshev bandpass filter with a bandwidth of 400 kHz, a nominal passband ripple of 0.03 dB, and a center frequency of 5.1 MHz. Thus, the output of the noise source is a band of noise 120 kHz wide in its flat portion and centered on f_o. For the purpose of the noise measurement, the above considerations allow us to replace the actual noise source by an equivalent white noise source whose power spectral density is $N_e = V^2_{nA}/2W_n$ where V_{nA} is the rms noise voltage at A and W_n is noise bandwidth of filter 1 in hertz. W_n was computed by numerical integration from uniformly spaced, measured values of the amplitude characteristic.

Finally, the ratio of the S/N out of the approximate matched filter to the S/N out of the exact matched filter is

$$\eta = \left(\frac{V_{sB}}{V_{sA}}\right)^2 \left(\frac{V_{nA}}{V_{nB}}\right)^2 \frac{1}{TW_n} \quad ,$$

where

V_{sA} = peak voltage of the signal pulse at A,

V_{sB} = peak voltage at B in response to the signal,

V_{nA} = rms noise voltage at A,

V_{nB} = rms noise voltage at B.

The theoretical value of η for this approximation is 0.9408. Measurements of η were performed on 70 of the 88 filters supplied by Damon Engineering, Incorporated, with the following results:

Average value of η = 0.918

Standard deviation of η = 0.016.

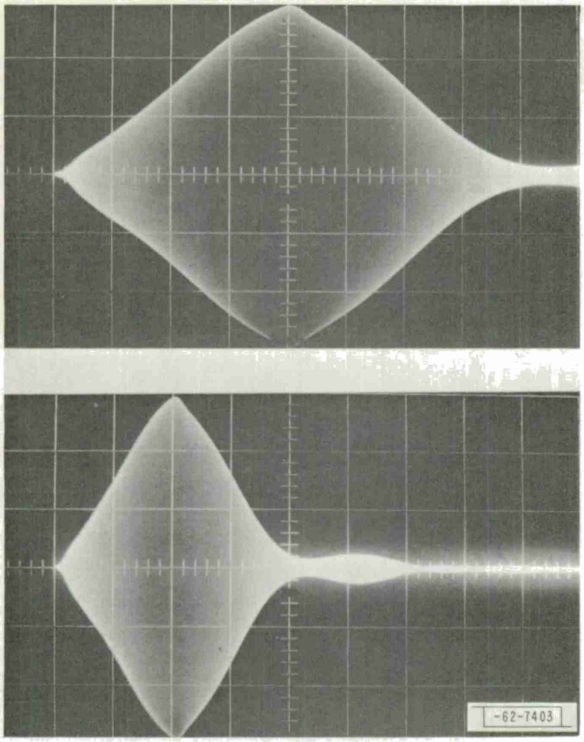


Fig. 3-15. Response of typical Damon crystal filter to its matched 200-μs pulse. Time scale on upper trace is 50 μs/cm, and on lower trace, 100 μs/cm. Note how well envelope matches Fig. 3-9.

Only three of the measurements were outside the limits of two standard deviations from the mean, and the measured average of η is 2.5 percent below the theoretical. This, together with the compactness of the measured values, attests to the high accuracy and uniformity of Damon's realization as well as of the measurements. A photograph of the response of a typical filter to its matched signal is shown in Fig. 3-15. Note the good agreement between the envelope of this response and the response of the low-pass filter in Fig. 3-7.

REFERENCES

1. B. E. Nichols and P. Rosen, "Lincoln Experimental Terminal," Chap. 1.
2. P. R. Drouilhet, Jr., "Signal Processing System," Chap. 2.
3. I. L. Lebow and P. G. McHugh, "A Sequential Decoding Technique and Its Realization in the LET," Chap. 5.
4. R. R. Stone and H. F. Hastings, "A Novel Approach to Frequency Synthesis," Frequency 1, No. 6 (Sept. – Oct. 1963).
5. G. L. Turin, "An Introduction to Matched Filters," IRE Trans. Inform. Theory IT-6, 311-329 (June 1960).
6. R. S. Berg and B. Howland, "Design of Wideband Shielded R. F. Transformers," Rev. Sci. Instr. 32, No. 7 (July 1961).

CHAPTER 4
COMPUTER SYSTEM

F. E. Heart and W. R. Crowther

ABSTRACT

A digital computer functions as a major subsystem of a satellite communications terminal called the Lincoln Experimental Terminal. The considerations which dictated its role in the communications terminal are examined, and the implementation of that role in terms of a computer program is discussed. Interesting features of the system include: antenna pointing and transmitter-receiver tuning; message traffic control with digital speech going through the computer memory in real time; unusually flexible computer-driven console controls; and asynchronous interrupt-based program design. The computer system has operated successfully as part of the terminal for over a year.

A. INTRODUCTION

Terminals for satellite communication often involve computational tasks (such as orbit computations) which require the availability of a digital computer. Many other signal processing tasks permit implementation either with a general-purpose computer or with special-purpose digital hardware. The increasing capability and decreasing cost of small, reliable, general-purpose computers offer an attractive opportunity to integrate many of these tasks into a single computer. Such an integration means time sharing a single piece of equipment and can provide an important simplification of the overall equipment system. Furthermore, a general-purpose computer offers potential flexibility and operator control which is difficult to match with special-purpose hardware. Lincoln Laboratory has been exploring this approach in the design of several kinds of compact communication terminals and has constructed the Lincoln Experimental Terminal (LET)[1-5*] using this approach.

The LET is a complete, self-contained, air-transportable communications ground terminal using satellite or moon channels. It handles a voice channel in digital vocoded form and two teletype channels. The digital bit streams are multiplexed, passed through a sequential (convolutional) encoder to add redundancy, and then pseudorandomly frequency hopped over a 20-MHz band to provide a high degree of interference rejection. A Univac 1218[†] militarized one-rack general-purpose digital computer with an 8192-word memory has been integrated into the LET. This computer directs the antenna at satellites or the moon from a starting point of orbit parameters and tabulated moon positions, respectively. Actual multiplexing of the vocoder and teletype bit streams, and the transmission format control are handled inside the computer, as are receiver tuning and many system control operations. Most operator control of the terminal is

[*] Numbered references for Chapter 4 may be found on page 75.

[†] See Appendix A, page 71.

handled through the computer via specially designed multipurpose console controls of unusual flexibility. The computer program, and the related interface equipment form one of the major subsystems of LET.

The design of a complex communication terminal raises groups of alternate implementation choices at every step. A fundamental one for LET was whether tasks should be assigned to a general-purpose digital computer or to special-purpose digital equipment. In the design of LET, this problem was attacked in the following way: A digital computer was sensible for antenna pointing, but what computers, available as shelf items, were adequate for the antenna pointing task and suitable for a limited-space transportable environment? The Univac 1218 best met the constraints at that time. All the other LET tasks were then examined to see which ones might sensibly be integrated into the particular computer selected. This chapter discusses the task allocation, and then describes:

Functional design
Man-machine interactions
Program system
Operating experience.

B. COMPUTER TASK ALLOCATION

The LET required the design of many major subsystems, including the physical antenna, cryogenic receivers, transmitters, power supplies, multichannel receivers, and filters. From this larger list, a substantial subset was concerned with various aspects of control and signal processing.[6] Inclusion of a digital computer for pointing computations raised the possibility of implementing other functions within the computer. These LET functions were considered:

Generation of range and Doppler estimates for time synchronization of pseudorandom sequence generators, and receiver frequency tuning

Sequential encoding and decoding on the communication bit stream

Generation of displays for operators, and provision for flexible operating controls

Receiver acquisition logic, for fine searching in range and Doppler

Multiplexing vocoder and teletype message traffic to provide format control, and the associated demultiplexing

Master clock and associated timing chains

The vocoder or portions thereof

Generation of pseudorandom sequences and resetting such sequences at arbitrary times

To the extent that functions could be programmed into the 1218, the overall system might be easier to implement and more reliable. However, it seemed clear that functions requiring service more frequently than every few hundred microseconds would have to be minimized or avoided. Further, use of the 1218 for a particular task would usually require an interface whose complexity had to figure in each choice. With these considerations in mind, each task was examined.

The computer was selected to implement pointing. A pointing computation providing angles to drive the antenna can be done easily in 50 ms. The mathematics of satellite orbits requires computation of a new point on the orbit only every two seconds to achieve our required system accuracy, using linear interpolations to provide intermediate points as the antenna servo requires them. Since this computation also provides range (path delay) and range rate (Doppler), the computer should provide data for time and frequency synchronization.

The sequential encoder-decoder was not suitable. The decoder requires one "step" every 6 μs. A "step" in the 1218 would take upward of 800 μs, putting it out of consideration by more than two orders of magnitude. The encoding might have been barely possible in the computer, but it is such a simple task that the interface is as hard to build as the encoder itself.

A whole set of operations centering on human control are suitable. These must run five or ten times per second to match the human response time. The only question is whether or not the job is bigger than the interface. Driving Nixie displays, which involves binary-decimal conversion, was worth the interface; so was the construction of a general-purpose knob, to control any parameter available to the machine.

The receiver acquisition tracker, which hunts for the signal inside a range and Doppler window, was doubtful. At the time LET was designed, it was not clear how fast the tracker would have to respond to acquire a target satisfactorily. The final specification for the tracker response put it outside the computer. This was part of a conservative design philosophy which hesitates to close servo loops in a general-purpose computer unless it is obvious that the timing requirements can be met with adequate margin.

Message traffic control was also doubtful. The function seemed to be well matched to the computer, since it involved buffer storage and bit manipulation, but the rates were rather high. A careful examination of the problem disclosed that computer implementation was practical, although it might take as much as half the machine capacity. It was allocated to the computer, where it actually took one-third the capacity.

The system clock, exclusive of the basic oscillator, could have been put in the computer. It was kept separate and provided with battery power supplies for reasons of reliability; the reset problem was quite serious.

The timing chain also appeared to be well matched to the computer, involving much bookkeeping and digital addition. Unfortunately, the data rate here proved to be too high, and the special equipment was constructed. The timing chain and the message traffic basically run at the same speed, but messages could be packed 16 bits to a word while the timing yielded to no such trickery.

The vocoder was not a real contender. It involved much analog as well as digital circuitry, and to put half a vocoder in the computer seemed impractical. The rates were also somewhat high.

The sequence generator might have been put into the program. However, it would not have been possible to program the feedback shift register algorithm directly and an easily programmable alternative was not immediately obvious. Furthermore, the shift registers could be very simple special-purpose hardware, and the decision went in that direction. The reset issue was assigned to the computer; a program in the LET system can begin at a known time and "giant step"* the pseudorandom sequence to determine reset codes.

It is interesting to speculate on how this division might appear in future terminals. With a factor of ten increase in speed, the whole digital portion of the signal processing equipment, except for the decoder, might be done in a general-purpose machine. This factor of ten could come from an improvement in computer memories or a relaxation of transmission rate. The decoder might also be eliminated by a coding scheme which permitted a more natural computer implementation.

*See Appendix B, page 72.

C. FUNCTIONAL DESIGN

In line with the foregoing reasoning, the LET computer system was given four major tasks (Fig. 4-1):

(1) Point the antenna

(2) Assist in time and frequency synchronization

(3) Multiplex message traffic

(4) Communicate with the operators.

The computer can point the antenna at four different classes of objects, with an accuracy of about one-tenth of a beamwidth (~0.05 degree). The first class of objects is earth satellites in orbits to which Kepler's equations apply. (In particular we are interested in the Lincoln Experimental Satellite[7] series of active satellites developed at Lincoln Laboratory.) Satellite pointing is accomplished by direct solution of orbital equations from a starting point of orbit parameters. The second class of objects has only one member, the moon. Pointing at the moon is accomplished by interpolation in a prestored table (on paper tape) of moon positions at half-day intervals. This approach seemed more promising than attempting a real-time solution of the multibody equations. The third class of objects are those which are stationary in space such as the sun and bright radio stars. Communication via these targets is impossible, but they can be detected with a radiometer and used to boresite the antenna. The fourth class includes objects which are stationary on earth. This option might be used to point at a target set up for boresiting purposes, or simply to put the system in a known unmoving position.

The pointing task includes the real-time variation of any of the parameters which define the particular object. When pointing at a satellite one can vary azimuth, elevation, or any of the satellite parameters, by using a handwheel on the operator's console. In actual operation, an errant satellite can usually be found by varying mean anomaly, which is the most difficult parameter to predetermine.

Receiver synchronization in time and frequency, the second computer task, involves predicting one-way path delay and Doppler for each of the four transmission paths involved in LET.

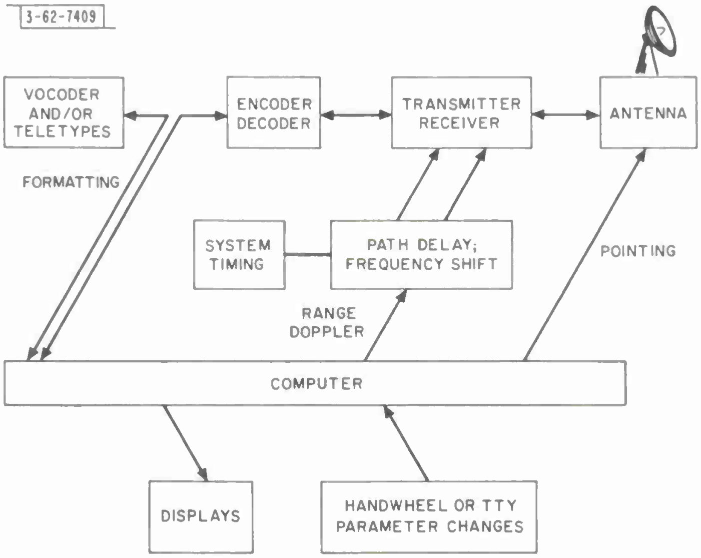


Fig. 4-1. LET computer tasks.

These paths carry the transmitted signal, the received signal from a remote site, the received signal from our own site, and the beacon signal (for which path delay is not needed). The nominal path delays are all alike, while the Dopplers are different because of the different frequencies involved. Values computed from the geometry of the situation are sent to the corresponding equipment where they are used to tune that equipment to the correct frequency and delay.

As in the pointing task, all important parameters are under control of the operator using the handwheel at the console. He can change the tuning of all the equipment at the same time by varying one of the parameters which controls the system geometry (like mean anomaly), or he can select a path delay or Doppler individually and enter a manual bias to its tuning. In particular, he can easily search in frequency or time for a missing signal. In this regard, site operators have found it impractical to search for a satellite by varying two parameters at once, be they pointing parameters, tuning parameters, or some combination. Instead, a careful search should be made in one variable while holding the rest fixed. An arrangement of one handwheel and display, with an array of buttons to select the variable, is ideal for this sort of search.

The third computer task is the coordination of various messages from site to site. Two teletype and one voice message must be buffered and multiplexed into the single bit stream which drives the LET encoder. Similarly, the incoming single bit stream must be demultiplexed and fed to the teletypes and the vocoder. Since LET must both transmit and receive, the single 1218 computer must handle these bit streams in both directions at once. These bit streams run at three different average rates, up to 10 bits/s (depending on the noise which must be overcome), and must stop periodically to allow the encoder to reset itself. The computer provides an ideal tool for changing formats to match rate changes and for buffering the bit stream.

The last computer task is communication with the operator. Aspects of this task have been mentioned above, and the next section is devoted to an expanded discussion of man-machine interactions.

D. MAN-MACHINE INTERACTIONS

The computer has a large number of parameters which are intended to be set by the operator, and an even larger number of quantities whose values he may wish to examine. A good deal of thought and effort was spent to make the interaction simple for the operator. Table 4-1 presents the regular system parameters. These parameters fall naturally into three classes:

(1) Quantities like orbit parameters that are set to predetermined values and normally left alone for long periods.

(2) Quantities like receiver Doppler bias that are continuously adjusted until satisfactory operation is achieved.

(3) Binary decisions, such as changing pointing from the moon program to the satellite program.

We handle the first class of parameters by using the same teletype that is used in the regular communications transmission. The operator signals that his message is for the computer by throwing a switch on the machine. This switch position is transmitted to the computer along with each teletype code, so that there is no confusion about the operator's intent. The operator types a two-letter code to identify the parameter (for example, PI means satellite parameter inclination) followed by a number, including sign or decimal point if desired. The number is interpreted as the value of that parameter in units which the operator considers natural for that quantity (degrees in the case of inclination, even though the program works in revolutions).

TABLE 4-1

SYSTEM PARAMETERS

Name	Units	Function
Month	xx months	Entering Current Date
Day	xx days	
Satellite Epoch Month	xx months	Entering Date of Validity of Orbit Parameters
Satellite Epoch Day	xx days	
Satellite Epoch Fraction	.xxxxx days	
Satellite Ω	±xxx.xx deg	Orbit Parameters
i	±xxx.xx deg	
ω	± xxx.xx deg	
e	.xxxxx	
n	xxx.xxx day^{-1}	
Mean Anomaly	.xxxxx rev	
Site Latitude	±xxx.xx deg	Site Parameters
Longitude	±xxx.xx deg	
Height	±xxxxx nmi	
Range Bias:		Range (Time Delay) Tuning
Transmit	±xxxxx nmi	
Monitor Receive	±xxxxx nmi	
Communications Receive	±xxxxx nmi	
Doppler Bias:		Doppler Tuning
Transmit	±xxxxx.x kHz	
Monitor Receive	±xxxxx.x kHz	
Communications Receive	±xxxxx.x kHz	
Beacon	±xxxxx.x kHz	
Right Ascension	±xxx.xx deg	Fixed Right Ascension and Declination Parameters
Declination	±xxx.xx deg	
Distance	xxxxxx nmi	
Azimuth	±xxx.xx deg	Manual Antenna Control
Elevation	±xxx.xx deg	
AZ-EL Range	xxxxxx nmi	Fixed AZ-EL Parameters (use with manual antenna control)
AZ-EL Doppler	xxxxx.x kHz	
Select: Moon	Yes – No	Program Control
Satellite	Yes – No	
AZ-EL	Yes – No	
RA-DEC	Yes – No	
Bogus Time	xxxxxx Min	Nonreal-Time Clock Control
Bogus Restart	Yes – No	
System Clock	HHMMSS	
Core Display	Address	Inspect and Change Core While Running
Change Core 1	New Value	
Change Core 2	Yes – No	
Octal-Decimal Mode	Change	
Test Mode	Yes – No	Equipment Test Operating Mode
Remove Refraction	Yes – No	
Transmitting Frequency	xxx.xxx kHz	X-Band Operating Frequencies
Receiving Frequency	xxx.xxx kHz	
Beacon Frequency	xxx.xxx kHz	

A design feature of considerable convenience specifies that it shall be possible to change any system parameter at any time while the system is running. The system has no initialization in the customary sense; it starts in the middle of some built-in track (which happens to be the tracking of a moderate altitude circular polar satellite from Lincoln Laboratory on 31 December 1964). As the orbit parameters, date and site coordinates are changed in any order, the system goes through a corresponding set of pointing tasks until the desired task is reached. Thus, mistakes are easily corrected by retyping the offending parameter. It is a simple procedure to vary any of the parameters to seek a missing satellite, although this is not normally done via the teletype.

The second class of parameters consists of those whose exact values are not known in advance (for example, Doppler bias), and which are "tuned" to optimize system performance. They are handled by a computer program control panel, which consists of a handwheel, 25 backlit buttons, three switches, and two 6-digit nixie displays (see center section of Fig. 4-2). The parameter is selected with one of the buttons, and the button lights. This action automatically makes the handwheel active for that parameter, and shows the parameter's current value in one of the displays (for example, Doppler would appear in kHz, with one digit to the right of the decimal place). The second display shows any manual bias already included in the parameter. Moving the handwheel increases or decreases the bias, so that the current position of the wheel is of no importance, but only its motion. One auxiliary switch allows the operator a choice of rates for either large scale slewing or fine adjustment. Although quantities like orbit parameters are intended to be handled by the teletype, they also occupy handwheel buttons. Thus it is possible to vary mean anomaly, and it is easy to inspect the current values of parameters. Typically, a careful operator will examine parameters as he enters them to insure against typing errors. Since the number of parameters is larger than a convenient number of buttons, some of the less used parameters are second options on the 25 regular buttons and the use of the second option is signaled by another auxiliary switch.

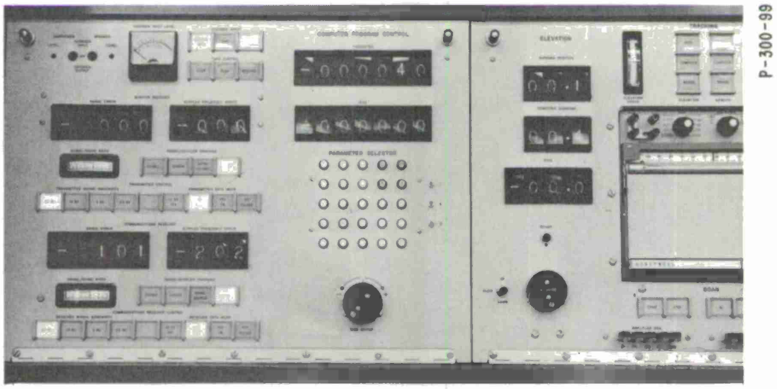


Fig. 4-2. Computer control panel.

The third class of parameters, the binary decisions, are also incorporated into the buttons. Pressing one of these buttons clears the display and signals the appropriate message to the computer. These buttons remain lit in addition to any selected and lit parameter button.

In addition to the "regular parameters," discussed above, many other registers of the computer are of occasional interest. For example, the troubleshooting of each piece of hardware connected to the computer is frequently facilitated by examining or modifying selected registers. This kind of specialized parameter examination is handled via a button-selected feature called the "register display." In this somewhat unusual selection, the handwheel controls the address of a register in memory. The two displays then show both this address and the contents of the register. With this option one can watch memory while the system is running and observe, for example, teletype characters arriving in the computer memory. The display is normally in octal, but can be changed to decimal by yet another button. The inspected register may also be set to any desired value. To insure that this does not accidentally destroy a memory location, it is necessary to press three separate buttons consecutively to use this option. The display changed ten times per second, fast enough to avoid any unpleasant delay between taking an action and observing the result.

Another feature of the man-machine interaction is the display on Nixie indicators of ten quantities (like command azimuth) of sufficient interest to justify continuous display. These displays are scattered on the main panels among the servo or transmitter-receiver controls wherever they are most convenient.

E. PROGRAM SYSTEM

The program system which implements the above design was constructed as a group of smaller programs. Figure 4-3 is a map of memory showing the size and location of each of these programs. These programs were specified in detail and coded by different programmers, although the final system was assembled as one large program.* The following paragraphs will briefly describe each of the smaller programs. In these paragraphs, reference made to "pointing program" means one of four programs which produce pointing information for a satellite, the moon, a fixed point in space (fixed right ascension, declination, radius), and a fixed point on earth (fixed azimuth, elevation, range). "Pointing program" refers to the program the operator has selected for the current run.

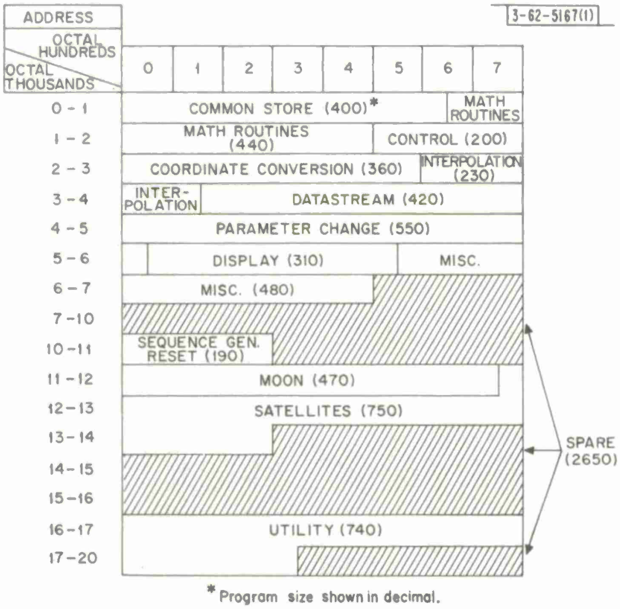


Fig. 4-3. Memory map.

*Many support programs were also produced. These are described in Appendix C, page 71.

The satellite program provides azimuth, elevation, range, and Doppler from a knowledge of orbit parameters and time. The computation is performed every 1/5 s. This is essentially the implementation of Kepler's equations with attention to difficulties in scaling and precision. The single equation (Euler's equation $E = M + e \sin E$) which cannot be written in closed form is replaced by one iteration of the recursive solution, using the solution of the previous 1/5 s as the new trial solution. This simple procedure finds the correct solution after a few fifths of a second. Prior to each pointing computation, the satellite program checks each of the orbit parameters. If one has changed, the program does an additional computation to update all the constants in the main computation which depend on this parameter. When initially started, the program tracks whatever random satellite happens to be in memory, and changes to the desired satellite as new parameters are entered. The program also makes an attempt to update those parameters most influenced by the earth bulge from their values at epoch to their current values.

The coordinate conversion program does that part of the pointing computation which is common to satellite, moon, and fixed right ascension-declination. Its inputs are X, Y, Z, X', Y', Z' in a nonrotating earth-centered coordinate system. Its outputs are azimuth, elevation, range, and Doppler at the site. Like the satellite program, this involves the implementation of a set of complex but well-understood equations. Again the problems encountered center on scaling and precision. The parameters of coordinate conversion – latitude, longitude, and height of the site – may be changed at any time and in any order. Coordinate conversion makes a very crude correction for refraction effects at low elevation, and provides three separate Dopplers based on three different transmitter frequencies (beacon, the local transmitter, and the other site's transmitter).

The moon program provides azimuth, elevation, range, and Doppler for the moon. Since the equations for the moon's motion are formidable, we chose to interpolate between values stored on a paper tape ephemeris. A convenient length of paper tape contains a month's data plus a few day's overlap. The tape has the moon's Cartesian coordinates, and their derivatives, plus second and fourth differences of these quantities for every 12-hour period. This information was obtained from a magnetic tape recording of this and similar data made by Jet Propulsion Laboratory.[8] Whenever a new date is entered, the moon program reads two points of the tape which bracket the current time, plus a third spare point 12 hours ahead. In the event that one desires to run more than 12 hours, the program will read additional points from the tape as old ones are discarded.

The interpolation program provides azimuth, elevation, range, and Doppler to the terminal by interpolating between values provided by the pointing computation program (moon, satellites, etc.). Interpolated values of azimuth and elevation are sent out to the servo system every 20 ms and new values of azimuth and elevation are computed by the pointing program every 200 ms. Range (path delay) and range rate (Doppler) are sent out at the same rate at which they are computed, every 200 ms. Manual biases, entered by the handwheel, are added by the interpolation program to the specified output quantities. In the case of range there are three range outputs which differ only in the manual biases added by the operator. These are transmitter range, communications receiver range, and monitor receiver range. In the case of Doppler, there are the corresponding three Dopplers and in addition a fourth Doppler called beacon Doppler. The interpolation program also acts as system timekeeper; its 20-ms outputs are tied to the system clock, and interpolation sets in motion several chains of events which result in timekeeping. The most important of these is the setting of a "pointing mark" every 200 ms which tells the control program to call the pointing program.

The Datastream program multiplexes and formats message bits from and to their various destinations. Input messages from a vocoder, a decoder, and two teletypes, are entered into the appropriate buffer region as they occur. A small amount of formatting and transferring from buffer to buffer occurs periodically and output messages for a vocoder, an encoder, and two teletypes are pulled from the appropriate buffer when the equipment so demands. The multiplexing consists of switching between buffers. Datastream would be a simple program, except that four of its requests may occur 600 times per second. At this rate, 50 instructions per request would use all the computer's capacity. With considerable care, Datastream has been held to one third of the machines total capacity. Datastream also permits selection of three different transmission rates (10,000 bits/s, 5000 bits/s and 200 bits/s) and produces a format appropriate to the rate.

Common storage is a region of memory in which all registers referred to by two or more programs are kept. This avoids programmer conflicts, and is convenient in a number of other minor ways.

The math subroutines are a collection of the math service routines used by the other programs. They include single precision square root, double precision square root, cube root, sine cosine, arc tangent, arc cosine, and some more exotic minor routines. These were all coded for the LET system and tailored to its accuracy-speed-storage needs. For example, cube root is small but slow because it is used only once when satellite mean motion changes, while square root is fast but large because it is used many times per second.

The control program does the necessary setup when the system is first turned on, and thereafter calls the pointing program every fifth of a second. Any left over time is spent in a loop in control waiting for the timekeeper part of the interpolation program to signal a new fifth of a second. While in this loop, control performs a multiplicity of small tasks, like restarting with an arbitrary time if requested, or clearing all biases if requested, or holding a fixed recognizable pattern in the computer console lights to signal "all's well."

The parameter change program processes all parameter changes. As far as this program is concerned, a system parameter is simply three registers to be set according to instructions from the handwheel or teletype. Every tenth of a second an input word tells the program which button is currently depressed and how far the handwheel has turned since the last reading. The program scales the new handwheel value, adds it to the parameter bias, and sets a mark. It also scales the new bias and total parameter value for the handwheel displays. Periodically the program looks for new teletype characters set up by the Datastream program. When a proper pattern of these is found, consisting of a recognizable two-character code followed by a number, the program scales the new value and puts it in a register for the user program. A mark is again left to tell the user that a new value has been received. Every parameter consists of three memory registers: one contains the mark (no change-new bias-new value), one the total bias or the new value, and one the nominal value (actual less bias). A large fraction of the parameter change program consists of tables of parameter locations, teletype codes, and scalings of various sorts. Once the three registers have been set, the further processing of a parameter change is done by the program concerned with that parameter.

The display program drives the various Nixie displays in the van (Table 4-2). Most of the displays are changed every 1/5 s, but some are changed at other rates. Display packs three Nixie digits and an identifying code in each output word. Display has a bookkeeping problem of

TABLE 4-2

CONSOLE DISPLAYS

	Units	Rate (/s)
Command Azimuth	deg	5
Command Elevation	deg	5
Actual Elevation	deg	5
Azimuth Bias	deg	5
Elevation Bias	deg	5
Doppler Error (monitor)	kHz	5
Doppler Error (communications)	kHz	5
Path Delay Error (monitor)	nmi	5
Path Delay Error (communications)	nmi	5
Computer Display 1 (value)	–	10
Computer Display 2 (bias)	–	10
Status	–	1

some magnitude when one considers that the scaling and number of digits are different for the different displays, that the display program must make up a block of five words at a time for each output, and that display takes an appreciable fraction (10 percent) of the machine's capacity.

The sequence generator program predicts the values of the sequence generators at 10-minute intervals. The sequence generator itself is a hardware random number generator of the shift register type. It is used as the random element in the LET frequency hopping scheme. In this scheme, a key point is that corresponding sequence generators at the two sites must generate identical values. If for some reason one generator is stopped, the sequence generator program provides a reset value so that the equipment can be manually restarted. The program is short, fast, and complicated. Since this technique has some intrinsic interest, additional detail is provided in Appendix B.

The miscellaneous programs consist of a number of simple routines. Included among these are fixed right ascension-declination and fixed azimuth-elevation, two of the main pointing programs. The first has been used to point at the sun and stars (radiometrically) for boresiting. The second has been used to provide known and unchanging messages for equipment troubleshooting.

F. PROGRAM CONTROL AND TIMING

This section will discuss the control logic. Particular emphasis will be placed on our use of interrupts, on how the programs keep track of real time, and on how the time is shared among the various programs. Figure 4-4 is a block diagram showing the control logic of the system.

The LET system control is built on the interrupt concept. About half the total computer time is spent in interrupt routines and three-fifths of the remainder is spare. Interrupt routines, therefore, dominate the system. The characteristic feature of an interrupt routine is that it operates only when a signal is received from the outside world. When this signal occurs, whatever

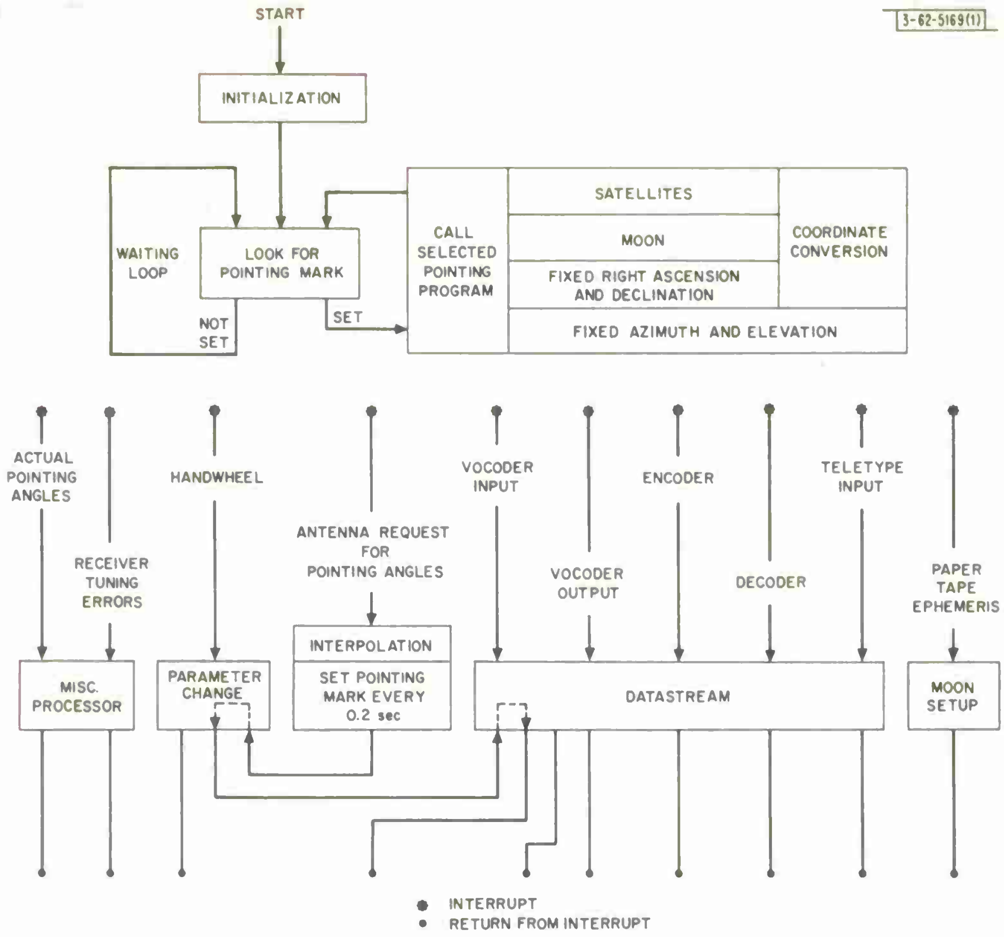


Fig. 4-4. Control logic.

is currently in progress is stopped and control is transferred directly to the interrupt routine. Naturally this is done in such a way that the interrupted program can be restarted at a later time without any loss of data. With ten major routines asynchronously interrupting the current operation, they often interrupt not only the main program, but also one another. In fact, four of the LET interrupt routines run 600 times each second, which means that on the average an interrupt will occur about every 420μs or every 50 instructions. With this sort of logic, it becomes impossible to trace in detail the exact sequence of control in the machine. Nevertheless, it is possible to write meaningful programs provided the routines are completely independent. In particular, no routine may use a value which another routine might change while interrupting the first routine.

The requirement of independence is not as severe as it might seem. Signals from the outside world are requests for more or less independent operations and the programs are designed to compute quantities well in advance of the time that they are needed so that no conflict can occur. Once independence is achieved several major advantages appear. Failure of any external equipment will have no effect except for the loss of data transfer to or from that equipment. Further, overall system logic becomes very simple because each program can be written as though it were the only program in the machine, and the problems of interaction reduce to the problems of not exceeding the capacity of the machine and of keeping real (clock) time.

Real time is kept in a rather strange way, which turns out to be quite easy to implement. The station clock, available on channel 5 of the computer, is not directly used to keep time. It is read only once, at the start of a run, and then only to the nearest four seconds. An interrupt from the clock occurs exactly on each 4-s mark. This interrupt is also used only once at the start of the run, to let us start exactly on the correct 4-s mark. For the rest of the run, real time is kept by the antenna control equipment. Since this equipment should ask for pointing information every 20 ms, the programs are designed to produce pointing data 20 ms apart. This means that the coordinate conversion program, which is called every 10 pointing outputs, rotates the earth by the fraction of a revolution corresponding to 200 ms each time it is called, and other programs do likewise. This system, in which each program keeps its own time in strange units like earth revolutions, has one major computational advantage. It avoids the double precision arithmetic which would be necessary to read and scale a 24-bit clock (in more conventional units) to the useful units.

There are two exceptions to the typical interrupt pattern. The interpolation program, in addition to its normal tasks, acts as system timekeeper. Its interrupt is used to keep real time as described above and occurs precisely every 20 ms. The interpolation program counts these interrupts and generates a signal (a mark) every 200 ms. The interpolation program also passes control to parts of the Datastream and Parameter Change programs which must be serviced periodically instead of by interrupt.

The second exception is the control program, which is started when the machine is turned on instead of by interrupt. The control program consists of three parts. There is a start-up section which operates once to get all the programs going properly. Then there is a waiting loop where the control program does miscellaneous nonurgent things while waiting for interpolation's 200-ms mark. Finally, there is the operating part which calls a pointing program when the mark is found. When the pointing program is finished, control is returned to the waiting loop with the mark cleared.

The program of timing interactions becomes complex as machine capacity is approached. The basic requirement is that no program interrupt itself; this means that the total time taken by a program P, plus all the other programs which might interrupt P, must be less than the interval between interrupts that call the program P. When the relation above does not hold, priority must be invoked to prevent some interrupts. Two simple priority arrangements were provided. In the first, the four very frequent interrupt routines of Datastream lock out all other interrupts while they operate, using a standard Univac interrupt-lockout instruction. In the second, the pointing computation was removed entirely from interrupts by the mark and the control waiting loop, giving it the lowest priority.

In the normal use of the system, pointing computations are made five times per second. Figure 4-5 shows how the fifth of a second is allocated among the various programs. Satellites will normally use only one-third of its allocated time, using the remaining two-thirds only when the parameter n (mean motion) changes. Similarly, parameter change normally uses only half its allocated time. The remaining time is not really available for any real-time process because it may disappear for arbitrarily long intervals. On the other hand, the time labeled "spare" in Fig. 4-5 is truly spare time, and is available for any task which might be found for it in the future.

It is interesting to note that the pointing computation could be cut from 20 percent of the total time to 2 percent by simply increasing the interpolation interval from 0.2 to 2 s. Beyond two

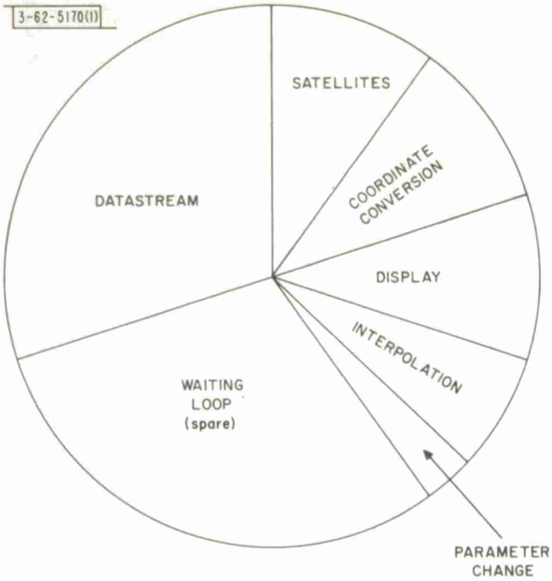


Fig. 4-5. Time allocation.

seconds the accuracy of linear interpolation falls below system specifications for low, fast satellites. In our case, there was no demand for the time which could be saved, and there was a definite advantage to running at the 0.2-s rate: The response time of the system to a manual change in orbit parameters is directly dependent on how often the pointing computation is done. In particular, it would be quite inconvenient to search for a satellite by manually varying mean anomaly if the system did not respond to the change for a second or so. When running at the 0.2-s rate, the response seems instantaneous to the human operator.

G. OPERATING EXPERIENCE

By mid-1966 three LET systems had operated for well over a year. The computer and the associated program functioned in a satisfactory fashion with very little "downtime." We were pleased by the simplicity and flexibility of the console controls and all operators found the arrangement convenient.

The insertion of a general-purpose machine into a complex system had two compensating effects, one good and one bad. On the negative side, certain equipment malfunctions were difficult to localize because "everything was so connected." When this became clear, special troubleshooting programs and techniques were designed. With these techniques the computer became an asset, since it could test all interconnecting equipment. (We would pay increased attention to this issue from the outset if the system were to be built again.)

A completely unexpected benefit of the design accrued to the computer's connection to an operating vocoder. During periods when the LET was not used for communications experiments, the computer-vocoder combination served as a useful research instrument. The flexible controls and displays allowed examination of new real-time speech bandwidth compression schemes in a fashion superior to other competing Laboratory computer facilities.

In general, the integration of a computer into such a communication system was successful and the technique is already planned for use in a new system.

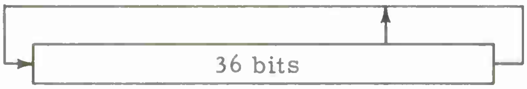
APPENDIX A
THE COMPUTER AND INTERFACE EQUIPMENT

The Univac 1218 is a one-rack militarized machine with an 18-bit word length and a 4-μs memory cycle time. Our machine has an 8192-word memory and paper tape program input. It has one core storage index register, hardware multiply and divide, no floating point arithmetic, and eight buffered input-output channels with interrupts. The machine is well suited to real-time process control because of its fine input-output system, so long as the 16 μs input-output transfer time per word is not objectionable. The cost of the machine was approximately $75,000.

There is a certain amount of equipment associated directly with the computer. This consists primarily of interface gear to buffer data transfers and to do conversion from integrated circuit levels to Univac's levels. The hardware amounts to three drawers of equipment with power supplies, containing 700 level converters and 1100 integrated circuit elements, 400 of which are flip-flops. Parts alone cost about $20,000, divided equally among the drawers, the integrated circuits, and the level converters. The interface equipment had to cope with 900 signals from the computer or the external gear, which implies an impressive number of wires.

APPENDIX B
SEQUENCE GENERATOR PREDICTION

The function of the LET sequence generators is to generate a pseudorandom sequence of 12-bit numbers, for use in the LET frequency-hopping system. This is done with a hardware linear shift register with one feedback tap.



Each time the register shifts one step, the new first bit is made of the mod-2 sum of the old last bit and the old 25th bit. Every twelve steps the last twelve bits are taken to be the new random number.

The function of the sequence generator prediction program is to provide reset values for the shift register in the event that timing is lost. The hardware includes provision for manual reset every ten minutes.

The straightforward solution of this problem would be to step a simulated shift register three million times (for each 10-minute period) and observe the answer. This is impossible in real time. The actual solution involves moving forward many steps at once (giant stepping). It depends on two properties of shift registers which will be stated here without proof:

$$F(I_1 \oplus I_2, t) = F(I_1, t) \oplus F(I_2, t) \qquad \text{(B-1)}$$

$$F(I, t_1 + t_2) = F(F(I, t_1), t_2) \qquad \text{(B-2)}$$

where I is a state of the shift register, t is a time, measured in shift register steps, F(I, t) is the new state which results from advancing state I forward t steps and $\oplus$ is the exclusive-or operation. Property 1 shows a way to giant step an arbitrary initial state provided one can decompose that state into the exclusive-or of a set of primitive states and provided one also knows how to giant step the primitive states. The simplest set of primitive states is the set of 36 different states each containing a single nonzero bit. The giant-stepped values of all these primitive states for a specific future time could be precomputed.

Actually, since the primitive states are 36 consecutive states of the shift register when properly ordered, the giant-stepped values at a particular future time are also 36 consecutive states. This means that only one giant-stepped state need be precomputed, and the remaining 35 states can be derived in a small number of steps.

Property 2 shows a way to do the precomputation to step the primitive state forward t steps. If one builds in F (primitive, 1) then repeated application of 2 allows one to compute

$$\begin{aligned} F(\text{primitive}, 2) &= F(F(\text{primitive}, 1), 1) \\ F(\text{primitive}, 4) &= F(F(\text{primitive}, 2), 2) \\ F(\text{primitive}, 8) &= F(F(\text{primitive}, 4), 4) \\ &\vdots \\ F(\text{primitive}, 2^{24}) &= F(F(\text{primitive}, 2^{23}), 2^{23}) \end{aligned}$$

Using property 1 to compute F(arbitrary, t) from F(primitive, t) when needed. Since the desired giant step of three million is not a power of two, one must write it as a sum of at most 24 powers of two and again apply property 2 to evaluate the sum.

Since each giant-step computation consists mainly of stepping a shift register 35 times, the whole process takes about the time necessary to step a simulated shift register (2) (24) (35) = 1680 times. In addition, the computation F (primitive, 10 min) can be done once and for all: the remaining 35 shift simulations can be done in a few tens of milliseconds. The storage necessary is three simulated shift registers or six machine words. The program itself takes a hundred or so words. These figures make the whole program negligible, both in running time and in machine space. For a program this short, it was remarkably complex and difficult to implement.

APPENDIX C
SUPPORT PROGRAMS

Various support programs were required to produce a program system of this size.

The Assembler-simulator is a program for the IBM 7094, which assembles 1218 programs from cards and runs them on a simulated 1218 in a simulated LET input-output environment. This tool was invaluable in a number of ways. It permitted:

- Programs to be written on cards instead of on less convenient paper tape
- Printing program listings in a convenient format in minutes by a line printer instead of in hours by a character printer
- Improving features of the Univac-furnished assembler
- Tracing of programs through interrupts in simulated time
- Debugging programs before the LET hardware was connected to the 1218, and even before the 1218 arrived
- Accurately reproducing troublesome conditions, a difficult task in a real-time system.

The 1401-card-image-to-paper-tape program, besides being generally useful, was used for two specific purposes:

(1) Convert the card decks to paper tape and assemble with Univac's Trim assembler.

(2) Punch 1218 programs assembled on the 7094 in binary on paper tape for loading into the 1218.

The first was not used often; the second became our standard operating procedure.

The console debug program is a set of debugging routines which can be operated from the 1218's teletype console. It includes Univac's utility routines UPACK which is a fine set of debugging routines for use when a typewriter console is not available, and expands those routines so that they can be operated from the standard Univac console or a teletype. It includes a simple trace option based on a small hardware modification to the computer: a device attached to a computer input channel that gives an interrupt 10 μs after it has been signaled, which allows exactly one main program instruction to be executed before the interrupt returns control to the trace routine.

The equipment test programs are a dull but useful set of routines which do tasks like sending the digit zero to all the Nixie displays, then sending the digit one, then two, etc. These were written as need appeared.

ACKNOWLEDGMENT

Portions of the program system were written by A.A. Mathiasen, P. Stylos, D.M. Hafford, D.C. Walden, L.R. Isenberg, and G. Blustein. The connection of the machine to its environment was implemented by P.D. Smith, D. Olsen (Univac), A.F. Dockrey, and E.J. Aho.

REFERENCES

1. B.E. Nichols and P. Rosen, "Lincoln Experimental Terminal," Chap. 1.
2. P.R. Drouilhet, Jr., "Signal Processing System," Chap. 2.
3. I.L. Lebow and P.G. McHugh, "A Sequential Decoding Technique and Its Realization in the LET," Chap. 5.
4. B.H. Hutchinson, S.B. Russell and J.W. Craig, "Modulation and Demodulation System," Chap. 3.
5. J. Tierney and J.N. Harris, "Channel Vocoder," Chap. 6.
6. P.R. Drouilhet, Chap. 2.
7. H. Sherman, D.C. MacLellan, R.M. Lerner and P. Waldron, "The Lincoln Experimental Satellite Program (LES-1,2,3,4) A Progress Report," AIAA paper No.66-271, AIAA Communications Satellite Systems Conference, Washington, D.C. (May 1966).
8. Peabody, Scott, Orozco, "JPL Ephemeris Tapes E9510, E9511, E9512," Technical Memorandum No.33-167, Jet Propulsion Laboratory (2 March 1964).
9. ____________________, "User's Description of JPL Ephemeris Tapes," Technical Memorandum No.32-580, Jet Propulsion Laboratory (2 March 1964).

CHAPTER 5
A SEQUENTIAL DECODING TECHNIQUE AND ITS REALIZATION IN THE LET

I. L. Lebow and P. G. McHugh

ABSTRACT

Probabilistic codes when suitably matched to modulation-demodulation systems allow communications which realize the theoretical performance predicted by the coding theorem. Sequential decoding is a form of probabilistic coding which allows realization in practical equipment to achieve this end. The Lincoln Experimental Terminal (LET) uses convolutional encoding and sequential decoding matched to a modulation system employing a 16-ary orthogonal alphabet and matched-filter envelope detectors, followed by an ordered list of the filters containing the 7 largest outputs. This coding system employing a constraint length of 60 bits and rates of 1 and 2 bits per orthogonal symbol achieves operation at an energy-to-noise ratio of 6 dB per information bit on an active satellite (Gaussian) channel.

The Fano decoding algorithm is employed. After a brief description of this algorithm, the realization of the LET encoder-decoder is presented. The machine using a commercial magnetic core memory, together with about 2000 integrated circuit elements, occupies about 20 inches of 19-inch rack space.

A. INTRODUCTION

There are many factors which influence the design of a communications system. They can be divided broadly into two classes: those which define the properties of the channel itself, and those which determine the character of the signals used in the channel. In the first category are parameters such as transmitter power, receiver sensitivity, antenna size and system bandwidth. For any communications medium some of these parameters are fixed and some are under the control of the user, who, in turn, attempts to maximize the performance of the channel subject to the inherent constraints under which he operates. The performance which is achievable given the channel specification is expressed in Shannon's theorem. If it is desired to communicate at a rate R bits/sec, then it is possible to find a set of channel signals of duration T, and a method of demodulating this set such that the resulting error probability P_e, is bounded above and below by exponentially decreasing functions of T, i.e.,

$$A_2\, e^{-TE_2(R)} < P_e < A_1\, e^{-TE_1(R)} \tag{5-1}$$

where the exponents are plotted in Fig. 5-1.[1]† The maximum rate for which this performance can be achieved is called the channel capacity C which depends upon such things as power, receiver sensitivity, bandwidth, etc. The numbers A_1 and A_2 are algebraic functions of T, dominated by the exponentials for large enough T.

† Numbered references for Chapter 5 may be found on page 101.

According to Eq. (5-1) and Fig. 5-1, one can achieve reliable communications at high rates (reasonable fractions of channel capacity) by using signals of sufficient duration. The lower bound to P_e defined by $E_2(R)$ is the best that can ever be achieved given complete freedom to choose 2^{RT} waveforms of length T, subject only to channel constraints. The upper bound given by $E_1(R)$ is computed by averaging over all possible sets of waveforms or by using "random coding."

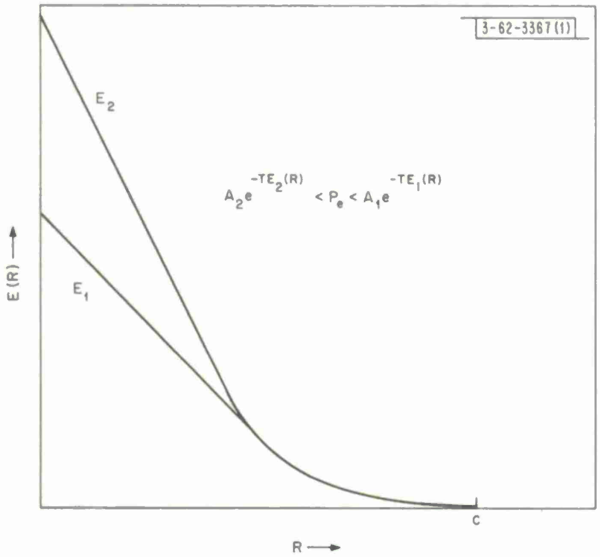


Fig. 5-1. Error exponents.

The bounds stated above imply the ability of the communicator to select 2^{RT} waveforms of length T out of the set of all possible waveforms of length T, subject only to channel constraints such as power and bandwidth, and then to detect them optimally. For these bounds to be of more than academic interest, one must find special methods of generating and detecting large numbers of signals. This is done by first quantizing the channel, i.e., by defining a relatively small set of waveforms of duration τ and then by constructing waveforms of length T as sequences of T/τ elementary waveforms. This has the effect of dividing the signal design problem into two parts: (1) modulation-demodulation, i.e., the specification of the generation and detection of the elementary waveforms and (2) encoding-decoding, i.e., the determination of sequences of the elementary signals and their decoding to optimize system performance.

This process converts a so-called continuous channel into a discrete channel. As before, one may compute upper and lower bounds to the error probability for such a discrete channel. But now the communicator has less freedom in his choice of signals and their detection, and so the resulting performance is poorer than for the continuous channel.

The same exponential behavior is obtained but the quantization process decreases the values of the exponents. In Fig. 5-2, we plot the upper bound exponents for a continuous and discrete channel. The more the channel is quantized, the lower the exponent for any value of R, and hence the greater the required value of T to achieve a desired error probability.

In designing a communications system one first defines a continuous channel to obtain the best practical E(R). Then this channel is quantized to achieve an exponent as close as practical to that of the continuous channel. Finally, the discrete channel is coded. The method of coding places limitations on the channel quantization which may be effectively employed, and therefore on the overall performance.

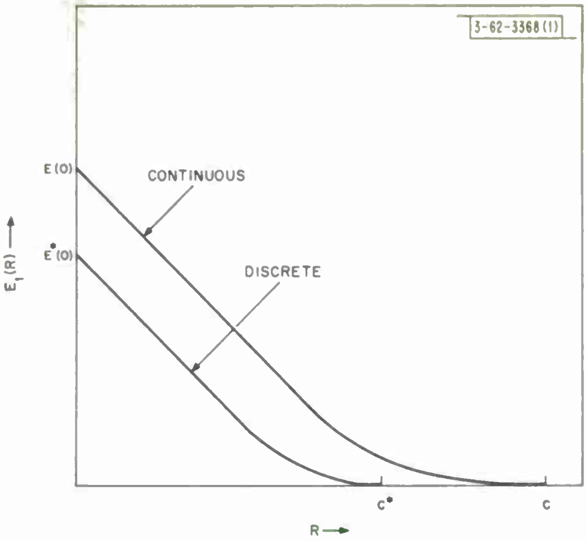


Fig. 5-2. Upper-bound exponents for discrete and continuous channels.

This paper is concerned with the sequential decoding[2] technique and its integration into the signaling system of the LET,[3,4] a terminal designed to communicate voice and teletype via an active or passive satellite repeater. Although we are applying the sequential decoding technique to specific discrete channels, there is no fundamental limitation on the type of discrete channel to which it can be employed. Before describing sequential decoding per se, we shall briefly describe the communications problem in terms of probabilistic block coding and use this to introduce the sequential technique.

B. BLOCK CODING FOR DISCRETE CHANNELS

We assume the channel model of Fig. 5-3. A transmitter and receiver define a continuous channel with capacity C. The other parameter that we use to designate the channel is E(0), the value of the upper bound exponent at R = 0 (see Fig. 5-2). The modulator and demodulator define the discrete channel. The input to the modulator is one of m symbols a_i which are mapped into one of m channel waveforms of length τ. The output of the demodulator is one of q numbers b_v which contain the information furnished by the demodulator about the received signal. If the channel is memoryless, then it is completely described by specifying $p(b_v|a_i)$, the probability of receiving symbol b_v given the transmission of a_i for all i and v. The parameters C* and E* (0), (Fig. 5-2), of the discrete channel are then functions of the $p(b_v|a_i)$.

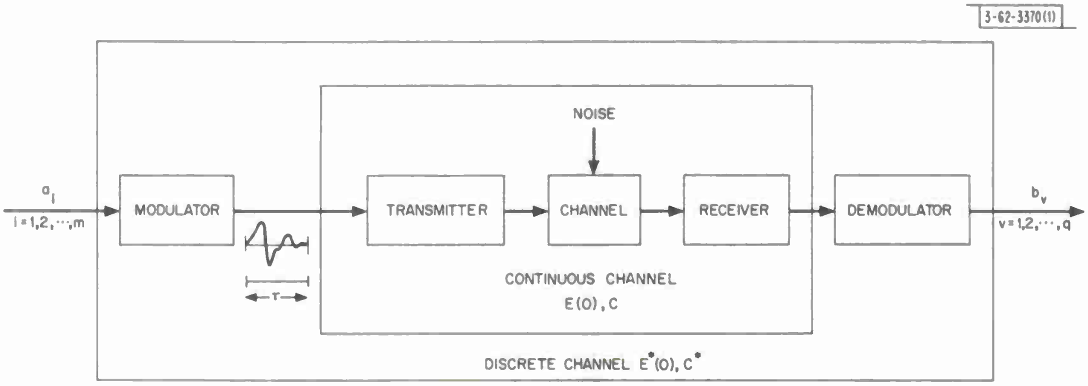


Fig. 5-3. Channel quantization model.

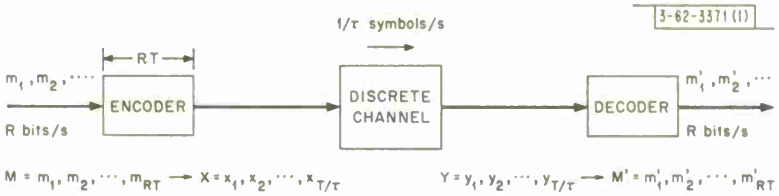

3-62-3371 (1)
1/τ symbols/s
RT
$m_1, m_2, \cdots$
R bits/s
ENCODER
DISCRETE CHANNEL
DECODER
$m'_1, m'_2, \cdots$
R bits/s
$M = m_1, m_2, \cdots, m_{RT} \rightarrow X = x_1, x_2, \cdots, x_{T/\tau}$
$Y = y_1, y_2, \cdots, y_{T/\tau} \rightarrow M' = m'_1, m'_2, \cdots, m'_{RT}$

Fig. 5-4. Block coding for discrete channel.

We code for this channel as in Fig. 5-4. An encoder takes a block $M = m_1, m_2, \ldots, m_{RT}$ of RT bits from an information source and maps this into a sequence of T/τ channel input symbols $x_1, x_2, \ldots, x_{T/\tau}$. The number $k = RT$ is called the constraint length of the code. There are 2^{RT} such sequences in the code designated by $X_1, X_2, \ldots, X_{2RT}$. The channel output is a sequence Y of T/τ output symbols $y_1, y_2, \ldots, y_{T/\tau}$. The decoder computes $P(X_i | Y)$ for each of the allowable sequences X_i in the code and sends out the sequence M' with the highest probability. Its decision is correct if $M' = M$.

The "matching" of a coder-decoder to a modulator-demodulator as in Fig. 5-4 calls for two things: (1) the ability to perform the indicated mapping of a sequence M of message bits into a sequence X of channel symbols and (2) the ability to perform the inverse mapping of Y into M.

The former implies the ability of the encoder to work with the channel input alphabet; the latter implies the ability of the decoder to operate with the channel output alphabet and the channel transition probabilities. Any limitation of the decoder in the use of different probability measures implies a limitation on the method of channel quantization which, in turn, may limit the resulting error exponent. The most flexible coding schemes are those which can utilize various alphabets and transition probabilities.

The decoder of Fig. 5-4 has the desirable properties described above but is generally impractical, since 2^{RT} probabilities must be computed to decode each block of RT bits. Sequential decoding provides the same capability of matching to channels but with far less computing. It does this basically by giving up the block structure in the encoder for a tree structure and by replacing the maximum likelihood decoder with a threshold decoder. We describe this in the next section.

C. CONVOLUTIONAL ENCODING AND SEQUENTIAL DECODING

The basic code structure for the block coding described in Fig. 5-4 is shown in Fig. 5-5(a) and (b). An infinite sequence of message bits $m_1, m_2, \ldots,$ is divided into disjoint blocks of $k = RT$ bits ($k = 4$ in the figure); parity check digits which are functions of the message bits in the block are then appended to the message bits to be applied to the modulator. Thus in the figure the check bits $c_4 - c_7$ are functions of $m_4 - m_7$ and these 8 bits are taken by the modulator to generate a block of channel symbols.

For sequential decoding a convolutional encoder such as that shown in Fig. 5-5(c) is used. Message bits are shifted through the k-bit shift register at a rate R bits/s. Every 1/R seconds, p (in this example two) bits are shifted out, the message bit and the output of the parity net. Thus, as is shown in Fig. 5-5(d), each parity check bit is a function of the k preceding message bits, a "sliding" rather than a "jumping" dependence.

The convolutional code structure can conveniently be represented in the form of a tree as shown in Fig. 5-5(e). The nodes of the tree correspond to the message bit selections, with the

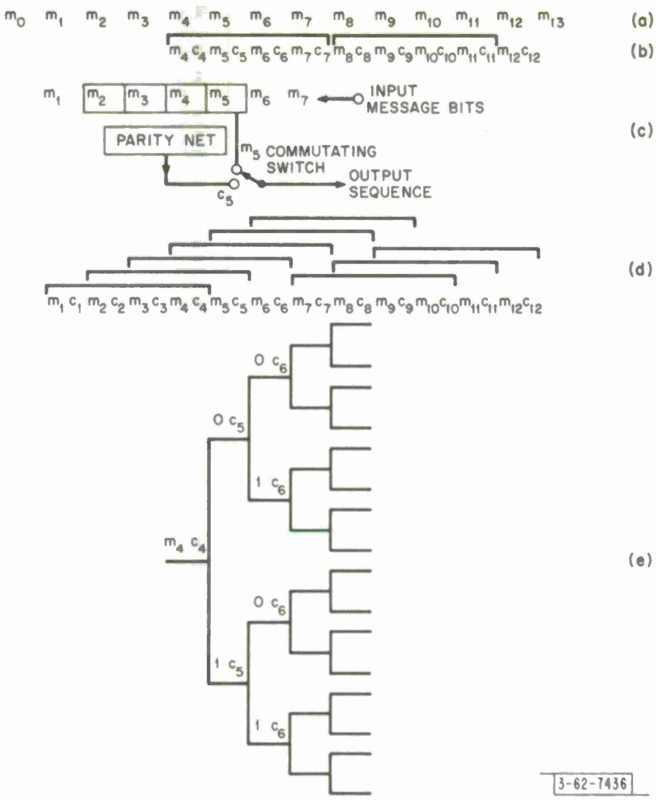

m0 m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 m12 m13
(a)
m4 c4 m5 c5 m6 c6 m7 c7 m8 c8 m9 c9 m10 c10 m11 c11 m12 c12
(b)
m1 m2 m3 m4 m5 m6 m7
INPUT MESSAGE BITS
PARITY NET
m5 COMMUTATING SWITCH
c5
OUTPUT SEQUENCE
(c)
m1 c1 m2 c2 m3 c3 m4 c4 m5 c5 m6 c6 m7 c7 m8 c8 m9 c9 m10 c10 m11 c11 m12 c12
(d)
m4 c4
0 c5
1 c5
0 c6
1 c6
0 c6
1 c6
(e)
3-62-7436

Fig. 5-5. Convolutional encoding.

upper branch at each node corresponding to a 0 and the lower branch to a 1. Once the message bit at a given node has been selected, the parity check bits which follow are completely determined, and hence are shown on the same branch. Thus a particular message sequence determines a particular path through the tree.

Also, in a simple extension, the encoder may take in q information bits at a time, in which case, each node of the tree would generate 2^q branches rather than simply the two branches shown in the figure.

As in the block case, a modulator maps the encoder output into channel symbols. Here, it is often convenient to use a channel alphabet such that a symbol corresponds to a branch of the tree. Thus, in Fig. 5-5(e), a suitable alphabet might be one with four symbols, each corresponding to one of the 2-bit sequences per branch. In the more general case in which q information bits and p check bits are generated per interval, an alphabet of size 2^{p+q} would result in a tree with a single symbol per branch with 2^q branches per node.

In the block case, an infinite sequence of message bits is divided into blocks of k-bits. Each of these is, in turn, mapped into a sequence $X = x_1, x_2, \dots, x_{T/\tau}$ of symbols. The decoder having received the sequence $Y = y_1, y_2, \dots, y_{T/\tau}$ of symbols for each transmitted block computes $P(X' \mid Y)$ for each of the 2^{RT} possible messages per block. In the convolutional case the transmitted sequence is an infinite one, $X = x_1, x_2, \dots,$ corresponding to an infinite path through the tree. The received infinite sequence, $Y = y_1, y_2, \dots,$ must similarly be compared with a replica of the transmitted message set, i.e., the tree, to find an infinite path X' which maximizes $P(X' \mid Y)$.

The sequential decoder attempts to do this one node at a time by finding a path X'_n, n branches long, such that $P(X'_n \mid Y_n)$ is sufficiently probable, where "sufficiently probable" is defined by a suitable threshold. If it is successful at length n, it tries to succeed at length n + 1 and thereby proceed indefinitely through the tree.

A "good" code, whether block or convolutional, is one in which the sequences X' in the message set are as different as possible for a given length. The longer the sequences, the greater the difference between any pair. This is the code tree property, and the sequential decoder uses it for its basic operation. Suppose the decoder starts out on the correct path and then proceeds forward along the correct path computing $P(X'_n|Y_n)$ from its knowledge of the noise behavior, where Y_n is the received sequence of n symbols and X'_n is the transmitted sequence. Similarly, if the decoder is on some incorrect path X", the probability $P(X''_n|Y_n)$ is computed. The code tree property makes $P(X'_n|Y_n)$ exceed $P(X''_n|Y_n)$ for any X''_n. At any time atypical noise events can occur which can make the correct path look bad ($P(X'_n|Y_n)$ small) or an incorrect path look good ($P(X''_n|Y_n)$ large). But the larger n the more the atypical noise events are averaged out and the more accurately the computation of $P(X|Y)$ distinguishes between a correct and incorrect path.

The sequential decoder is forced to start out on the correct path. As it proceeds forward along this path, it computes $P(X_n|Y_n)$ for the entire path up to the current node, and asks whether this probability is sufficiently large to warrant the continued assumption that the path is correct. If $P(X_n|Y_n)$ is sufficiently large, it proceeds forward to the next node, picks the more probable branch, computes $P(X_{n+1}|Y_{n+1})$ and continues. If $P(X_n|Y_n)$ is not sufficiently large, two hypotheses are possible: (1) the decoder is on the correct path which now looks improbable due to a currently occurring noise event or (2) some atypical noise events occurred in the past which caused the decoder to take an incorrect path. The decoder first adopts hypothesis (2), and searches backward in a systematic way in an attempt to find a better path. If it succeeds, it assumes that this new path is the correct one and attempts to advance. If it fails to find a more probable path under hypothesis (2), it returns to the original path which first looked bad, assumes that it is nevertheless the correct path and attempts to advance.

In this way the decoder advances to the right in the tree. The farther it advances (the higher n), the more confidence can be placed on decisions (nodes) early in the path. These decisions may thus be passed on to the user.

D. FANO ALGORITHM

In the preceding section the Fano sequential decoding algorithm[5,6] was described in gross. In this section we describe the algorithm in sufficient detail to lead to a description of its machine implementation.

At any point in the decoding process, the decoder is at the nth node of a tree on a path designated X_n. It computes $P(X_n|Y_n)$, where Y_n is the output sequence obtained from the demodulator. It thereupon compares $P(X_n|Y_n)$ with a threshold to determine whether or not the probability is sufficiently high to warrant accepting X_n as the correct path.

Using Bayes rule, the <u>a posteriori</u> probability may be expressed as

$$P(X_n|Y_n) = \frac{P(X_n, Y_n)}{P(Y_n)} = \frac{P(X_n)\ P(Y_n|X_n)}{P(Y_n)} \qquad (5\text{-}2)$$

where $P(X_n)$ and $P(Y_n)$ are the input and output sequence probabilities. $P(X_n, Y_n)$ is the probability of the joint event and $P(Y_n|X_n)$ is the channel sequence transition probability. If all tree paths are equally likely, then

$$P(X_n) = 2^{-R\tau n} = 2^{-\bar{R}n}$$

where as before R is the rate in bits/s, τ is the symbol duration and $\bar{R} = R\tau$ is thus the rate in bits per symbol. Assuming further that the channel is memoryless, Eq. (5-1) may be expressed as the product

$$P(X_n | Y_n) = 2^{-\bar{R}n} \prod_{i=1}^{n} \frac{p(y_i | x_i)}{p(y_i)} \quad . \tag{5-3}$$

It is generally more convenient to use an additive measure of probability as opposed to a multiplicative measure. Such a measure is obtained by taking logarithms. Equation (5-3) then becomes

$$L_n = L(X_n, Y_n) = \log P(X_n | Y_n) = \sum_{i=1}^{n} \left[\log \frac{p(y_i | x_i)}{p(y_i)} - \bar{R}\right] = \sum_{i=1}^{n} \lambda_i \tag{5-4}$$

all logarithms being taken to base 2. The expression $\log p(y_i | x_i)/p(y_i)$ is just the mutual information between the hypothesized channel input x_i and the channel output y_i. Thus λ_i, the difference between this mutual information and the information $\bar{R}$ carried by each symbol, should, on the average, be positive for the correct branch and negative for incorrect branches. In other words, L_n should be on the average an increasing function of n for the correct path, and a decreasing function for incorrect paths, although for any short sample the behavior of L_n can, of course, depart from its expected behavior.

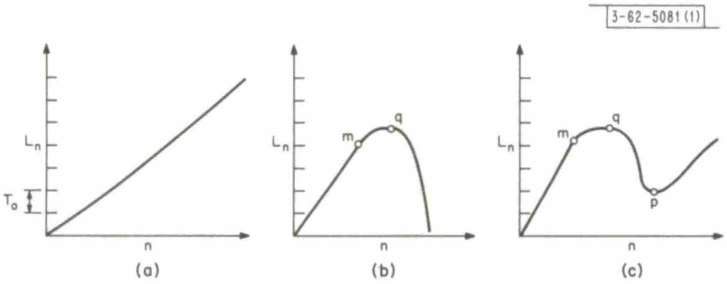


Fig. 5-6. Three cases of L_n vs n.

In Fig. 5-6 we plot L_n vs n for three cases. Figure 5-6(a) clearly represents a correct path, since L_n is a monotone increasing function of n. In Fig. 5-6(b), L_n begins to decrease at point q and continues decreasing indefinitely. This represents an incorrect path, although the portion of the path to the left of point q may be correct. In Fig. 5-6(c), L_n increases, starts decreasing at point q, but finally increases again at point p. This is a correct path which exhibits atypical behavior between points q and p. Since the decoder must find the correct path, it has an easy job as long as L_n behaves as in Fig. 5-6(a). It does this by defining a set of threshold values $T_1, T_2, \ldots,$ all separated by T_o, as shown in Fig. 5-6.

The decoder uses these thresholds to distinguish between two states. Suppose the decoder is at the nth node of a path X_n. The two states are defined as follows:

State (a) Path X_n is assumed correct; the decoder attempts to find a sufficiently probable X_{n+1} stemming from X_n.

State (b) Only the first (oldest or left-most) m branches, X_m, of path X_n are assumed correct; the decoder attempts to find a path stemming from X_m which looks sufficiently probable.

As long as the decoder is in state (a), the current threshold is defined to be that threshold line immediately below the current value of L_n. Thus as L_n rises (Fig. 5-6(a)), the current threshold value rises with it. If the decoder is in state (a) at node n, it enters state (b) if L_{n+1} drops below the current threshold. In other words, when a sufficiently large hilltop (Fig. 5-6(b) or 5-6(c)) is formed, the decoder enters state (b). The path is assumed correct only up to the oldest node above the threshold (node m above) at which state (b) was entered. The threshold is frozen at this value, and the hill above this threshold is searched in an attempt to find an alternate path stemming from node m which remains above the threshold. If such a path is found, the decoder reverts to state (a) and attempts to proceed forward. If no such path is found, the decoder reverts to the original path at the node which first fell below the threshold, resumes state (a) and drops the threshold to the next lower value. If L_{n+1} should drop below the new threshold, the top two layers of the hilltop are now searched in state (b) for a sufficiently probable path. This process continues indefinitely in the attempt to reject incorrect paths like that of Fig. 5-6(b) and to find the correct path of Fig. 5-6(c).

The depth of the strata by which the hilltops are searched is, of course, governed by the value T_o. The smaller T_o the more responsive the decoder is to an incorrect decision, but the more atypical noise events can spoof the decoder into calling the correct path incorrect. Conversely, a large value of T_o makes it less likely that a correct path will look incorrect due to such noise events, but the longer the decoder will remain on an incorrect path. In the former case, there will be many short searches; in the latter case, there will be fewer searches, but the average search will tend to be longer. It has been shown[7] that the average number of computations (number of nodes searched) is relatively insensitive to T_o over a wide range of T_o.

An illustrative example is shown in Fig. 5-7. Following L_n from the left we see a rising curve. Since an increasing L_n implies a correct path the decoder is in state (a). At the node labeled (1) L_n dips slightly; since it does not fall below a threshold line, the decoder remains by definition in state (a). L_n continues rising, finally dropping below the latest threshold level at point (2). The decoder is now defined to be in state (b). The path up to the left-most node

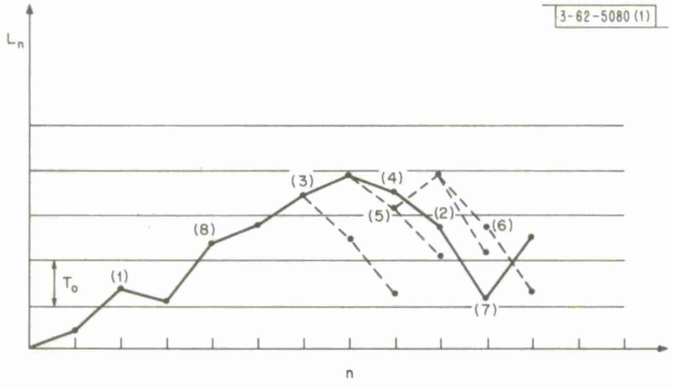


Fig. 5-7. Example of L_n vs n.

(3) above this threshold line is assumed correct, and an attempt is made to find some path stemming from this node which remains above the threshold. The path ending at point (5) meets this criterion and the decoder returns to state (a). However, this new path fails at point (6). Once again state (b) is assumed with the path up to point (3) assumed correct. Since no path stemming from (3) is found which remains above the threshold value, the decoder reverts to point (2) and state (a). In so doing, it assumes that local noise in this vicinity has caused this correct path to look improbable. It, thereupon, drops to the next lower threshold value, attempting to find a path which stays above this lower value. It fails to remain above this threshold at point (7), goes into state (b) in a search for a path stemming now from (8) which remains above this threshold. Finding none, it reverts to (7) in state (a), lowers the threshold, and is then able to continue forward.

A flow diagram for the implementation of the Fano algorithm is shown in Fig. 5-8. We assume a tree with b branches stemming from each node (in Fig. 5-5, b = 2). Those branches stemming from a node n are labeled by the index I(n), $(1 \leqslant I \leqslant b)$, in order of probability, i.e., I = 1 labels the most probable branch or the branch with the largest value of λ. Thus, at the point labeled start, the decoder computes $L_1 = \lambda_1(1)$ for the most probable branch and compares

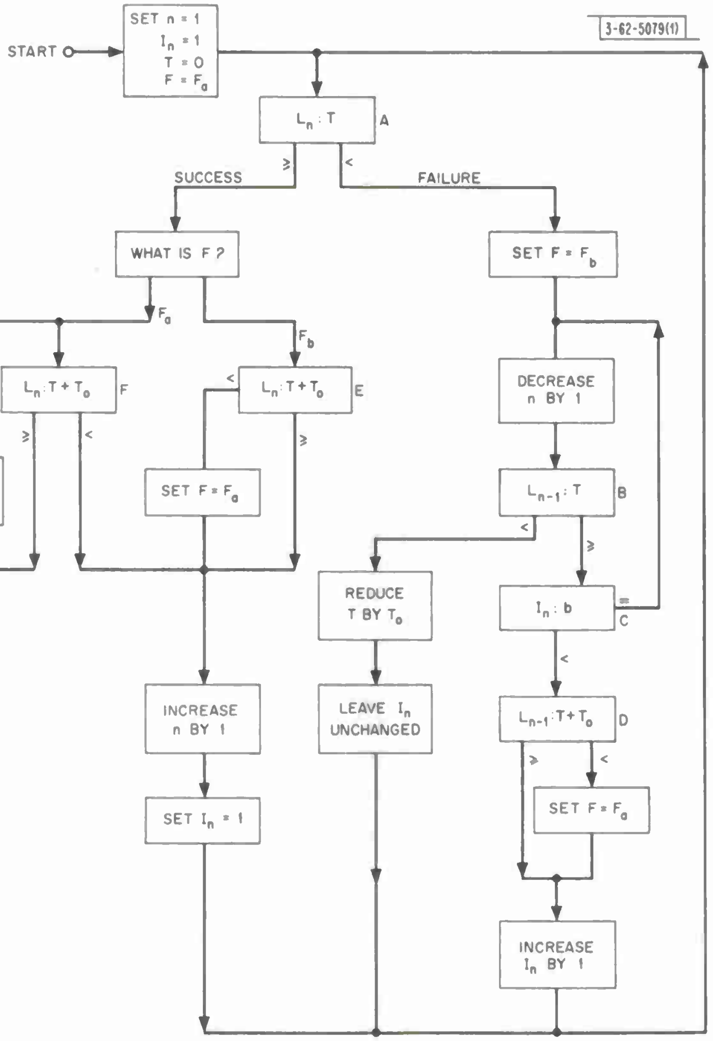


Fig. 5-8. Fano flow diagram.

it with the initial threshold value. The function F has two values, F_a and F_b designating the two state hypotheses (a) and (b). Since the decoder starts out on the correct path ($F = F_a$), if the first comparison is successful, it proceeds through the leftmost path of the flow diagram raising the threshold if necessary (Box F) and advancing to the right in the tree (increase n by one). This loop is followed as long as the decoder is in state (a).

With a failure in the comparison of Box A, the decoder enters state (b). The next comparison (Box B) tests for the left edge of a hill by comparing the value of L two nodes to the left of the one which just failed with the threshold. If this node is above the threshold, it asks whether all paths stemming from that node have been used (Box C). If not, it asks further whether the node is above the next higher threshold (Box D). In both cases, the next most probable branch is taken, and the decoder attempts to move forward. In the case of Fig. 5-9(a), the topmost layer of a hill is being searched, and a new branch represents a new path previously unexplored. In Fig. 5-9(b), the second level of a hill is being searched. As long as the path being examined lies in the top level, it would have to have been examined on the previous search with the next higher threshold; thus, it is not a new path, and a return to state (a) at this point would lead to a loop. If the attempt to move forward (Box A) in the case of Fig. 5-9(b) is successful, state (a) is entered only if L_n is in the lowest stratrum of the hill above the threshold (Box E).

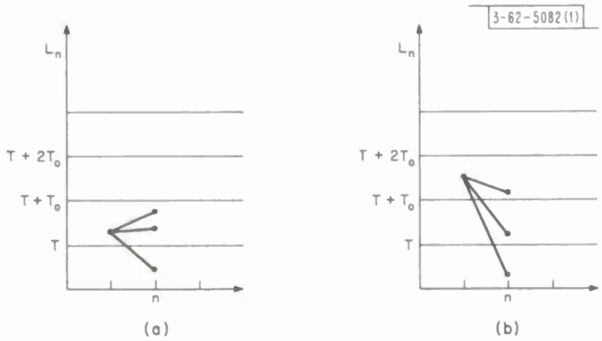


Fig. 5-9. Searching: (a) Node in layer above the threshold. (b) Node in second layer above the threshold.

Returning to Box C, if all paths stemming from the node have been exhausted, the decoder moves to the left and repeats the test for the left edge of the hill.

If in Box B, the left edge of the hill is detected, the decoder returns to the point at which the path first failed, and state (a) is resumed with a decreased threshold. The algorithm forces this to happen by retracing the original path one node at a time. This is done in the flow diagram by reducing the threshold, leaving $F = F_b$ and proceeding through Box A. The test with $T + T_o$ (the old T) in Box E detects the desired original path end, and state (a) is resumed.

E. COMPUTATION PROBLEM

In his original work on sequential decoding, Wozencraft[2] showed that the error bounds obtainable through the use of sequential decoding were essentially the same as those obtained for probabilistic block codes having the same constraint length.

There still remains, however, the question of decoder computation. In the block decoder, 2^k sequences had to be compared with each block of received data. In the sequential decoder, the number of computations is clearly a random variable. Whenever the noise perturbations are small, the decoder can advance rapidly. During large noise perturbations considerable searching may be necessary. Even though every incorrect path must eventually look "bad," the longer the decoder remains on an incorrect path the more branches will have to be searched to correct the error. Any realization of a sequential decoder must therefore provide buffer storage to hold incoming data from the demodulator, during periods in which long searches are taking place.

The method of operation of the decoder buffer storage may be explained with reference to Fig. 5-10. One may imagine the storage to be a window containing within it received data corresponding to N_1 nodes of the code tree, with the window sliding to the right at the rate at which data are transferred from the demodulator to the decoder. As the window slides to the right, the information contained in the leftmost node in the window is passed on to the user. Similarly, the rightmost end of the window passing over a node corresponds to new data entering the decoder. At any given time, the decoder is examining a branch located at some point P inside the window W nodes from the right edge. When P moves forward to the right end of the window, it is keeping up with incoming data, and the waiting line W is zero. When the decoder is forced by noise to search back, the point P moves to the left. The distance $N_1 - W$ between P and the left end of the window is just the length of the path between the node being delivered to the user and the node being decoded. From the discussion of the previous section, it is evident that the longer this path the more likely the node being delivered is on the correct path. When P reaches a fixed point P_1 which defines a minimum distance $N_1 - W_1$ from the left (W_1 from the right), the decoder is forced to stop to prevent the possibility of incorrect data being passed to the user. Once this occurs, decoding cannot be resumed until the decoder is resynchronized, i.e., restarted on the correct path. This can be performed on request via a feedback channel or else performed periodically by agreement, whether or not the decoder overflows. In the former case delays ensue; in the latter case, data are erased whenever an overflow occurs.

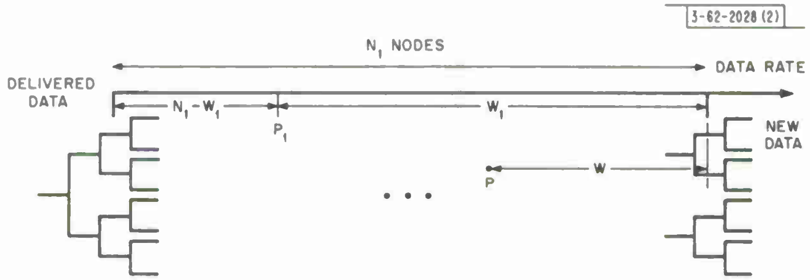


Fig. 5-10. Decoder storage utilization.

By making the distance $N_1 - W_1$ sufficiently large, the probability of actually delivering incorrect data can be made arbitrarily small. The question that remains is what is the probability of the decoder stopping? This is just the $\Pr(W > W_1)$ which in turn is related to the distribution of the number of computations, $\Pr(N > N_A)$, where N is the number of computations to decode a digit and N_A is any fixed number.

In his original work Wozencraft[2] estimated the behavior of the mean number of computations per decoded digit, finding an algebraic dependence upon the rate R for rates less than some maximum value called R_{comp} which is below capacity. For $R > R_{comp}$, the mean number of computations increases exponentially. The value R_{comp} depends only on the channel and is numerically equal to E(0) defined previously.[1] Other work both theoretical[8,9] and experimental[7,9] has led to the result that the number of computations per decoded digit follows a Pareto[10] distribution. This is to say,

$$Pr(N > N_A) \sim N_A^{-\beta} \tag{5-5}$$

where β is a function which depends principally on R/R_{comp}, decreasing to unity for $R = R_{comp}$. For $\beta \leqslant 1$, the mean number of computations diverges.

A plot of the exponent β vs R/R_{comp} is shown in Fig. 5-11. The points were obtained experimentally[7,9] by computer simulation for a variety of discrete channels. The dependence is approximately linear, and the solid line represents an average behavior. In all cases, the decoder parameters were cptimized at $R/R_{comp} = 0.91$. Presumably β would tend to 1.0 at $R/R_{comp} = 1.0$ rather than at 0.95, were the decoder parameters optimized at that point.

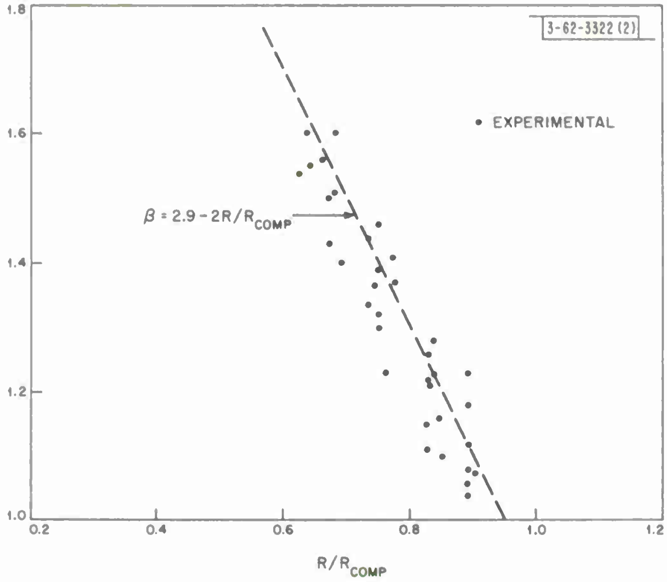


Fig. 5-11. Computation exponent β vs R/R_{comp}.

Thus a sequential decoder although capable of providing arbitrarily accurate decoding is limited to operation at rates below $R_{comp} = E(0)$ due to the nature of the waiting line statistics.

F. LET SIGNALING SYSTEM

In the remainder of this paper we shall describe the signaling system used in the LET.[3,4] The terminal has the function of passing a single vocoded voice channel with two multiplexed teletype channels via both active and passive (including dispersive) satellites and to achieve this in such a way as to yield a high multiple access or antijam capability. One of the implications of the latter is the desirability of operating at a minimum value of E_b/N_o, the ratio of the received signal energy per information bit to the single-sided noise power per cycle. The active satellite provides essentially an undisturbed signal; the only perturbation is additive Gaussian noise in the receiver. The capacity of such a channel[1] approaches

$$C = \frac{P_R}{N_o} \frac{1}{\ln 2} \tag{5-6}$$

and R_{comp} approaches C/2, in the wide bandwidth limit, where P_R is the received power. Thus, the minimum value of E_b/N_o required is ln 2 for a system operating at R = C. The minimum value achievable using sequential decoding is 2 ln 2. In the wide bandwidth limit, when a dispersive channel such as the moon is used as a passive reflector, the capacity is that of the Gaussian channel but R_{comp} is at most one-third that of the Gaussian channel.[12,13]

The LET signaling system uses a modulation-demodulation system[14] designed to yield a high value of R_{comp} or equivalently a low value of E_b/N_o. This, when matched to a sequential decoding system operating at a data rate $R < R_{comp}$, will yield the performance indicated by Fig. 5-11 and Eq. (5-5).

The required data rates to be accommodated are 4800 and 9600 bits/s corresponding to two modes of the LET vocoder.[15] When the additional incremental rates needed for the teletype and bookkeeping data are included, the actual signaling rates of 5000 and 10,000 bits/s are obtained. The elementary signal used to achieve this is a pulse of sinusoid 200 μs in duration. The two rates then correspond to one or two information bits per transmitted pulse.

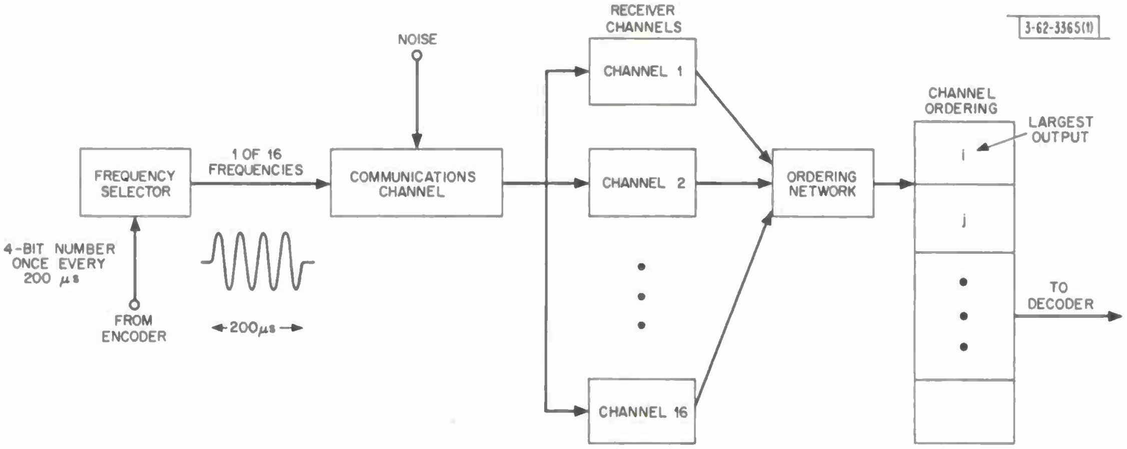


Fig. 5-12. LET modulator-demodulator.

The modulation system shown in Fig. 5-12 uses an input alphabet of 16 symbols obtained by assigning one of 16 frequencies to the transmitted pulse. The symbols of the alphabet are made orthogonal by separating the frequencies sufficiently. The received signal is detected in a bank of 16 receivers each containing a filter matched to one of the transmitted pulses followed by an envelope detector. The detector outputs are sampled and then listed in the order of decreasing amplitude. The identifying numbers of those channels with the seven largest outputs (in order) comprise the output information available for decoding. Envelope rather than synchronous detection provides compatibility with all the different types of satellites.

The convolutional encoder has a constraint length of 60 bits. It assigns one symbol per tree branch. Thus, each branch contains one of 16 symbols with either two (5000 bits/s) or four (10,000 bits/s) branches per node. This is shown in Fig. 5-13. Using the notation of the previous sections, each transmitted symbol (branch) x is a member of an orthogonal alphabet a_1, $a_2, \ldots, a_{16}$. The received symbol y is a vector $(b^1, b^2, \ldots, b^7)$, each of the b^j taking on one of the numbers a_i with no number appearing twice in the list. Suppose that $x = a_i$ is the transmitted signal, then the received symbol y either contains a_i in the jth position b^j of the list or does not contain a_i on the list.

Given the knowledge of the type of channel (Gaussian or Rayleigh fading, for example) and of the energy-to-noise ratio per symbol, we may compute the $\ell + 1$ conditional probabilities.

$$P_j = p(b^j = a_i | a_i) \quad , \quad j = 1, 2, \dots, \ell, \; i = 1, 2, \dots, m$$

$$P_{\ell+1} = p\left\{(b^j \neq a_i | a_i), \; j = 1, 2, \dots, \ell\right\} \quad , \quad i = 1, 2, \dots, m \tag{5-7}$$

of the transmitted symbol a_i appearing in the different list positions or off the list. Since the signals are orthogonal, all the symbols which were not transmitted appear in the different list positions with equal probabilities, and the channel transition probabilities $\{p(y|x)\}$ are given for all x and y by

$$p(y|x) = \frac{(m - \ell)!}{(m - 1)!} P_j \quad , \quad j = 1, 2, \dots, \ell$$

$$p(y|x) = \frac{(m - \ell - 1)!}{(m - 1)!} P_{\ell+1} \quad . \tag{5-8}$$

Once these channel transition probabilities are known, the value of R_{comp} may be computed[1] and the values of alphabet size m and list size ℓ selected from these computations.[9,11] Figure 5-14 shows a typical result for the active satellite channel. Under the assumption that each transmitted m-ary symbol carries 1 bit of information, the energy-to-noise ratio E_b/N_o is calculated to achieve a value of $R = 0.9 \, R_{comp}$ as a function of ℓ for fixed m. The values at the right of the figure are the asymptotic values which would be achieved if, instead of using a list, all the receiver information were employed. The mark labeled ∞ is for an infinite orthogonal alphabet; this value of 4.8 dB exceeds the theoretical maximum $2 \ln 2 \approx 1.4$ dB because envelope detection is used. The LET system value ($m = 16$, $\ell = 7$) of 5.7 dB is only slightly worse than the (∞) point.

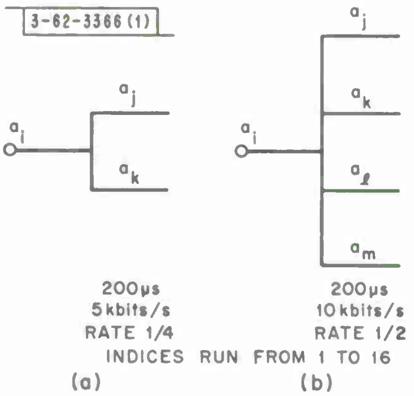


Fig. 5-13. LET code structure.

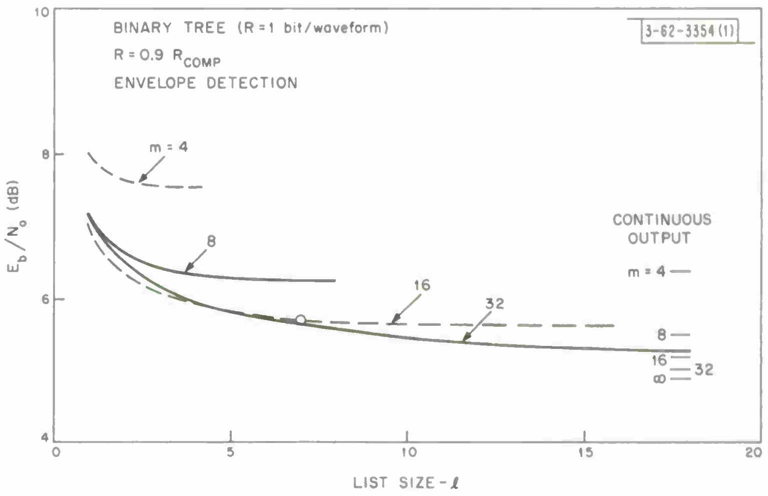


Fig. 5-14. E_b/N_o vs list size.

The LET decoder employs the Fano algorithm described in Sec. 5-D. At each step in the decoding process it computes the value λ for a hypothesized branch x^* where from Eq. (5-4)

$$\lambda = \log \frac{p(y|x^*)}{p(y)} - \overline{R} \quad . \tag{5-9}$$

If all input symbols are assumed equiprobable, all output symbols have the probability

$$p(y) = \frac{(m - \ell)!}{m!} \tag{5-10}$$

and the metric λ takes on the $\ell + 1$ values

$$\lambda^j = \log m \; P_j - \overline{R} \quad , \quad j = 1, \ldots, \ell$$

$$\lambda^{\ell+1} = \log \frac{m}{m - \ell} \; P_{\ell+1} - \overline{R} \tag{5-11}$$

obtained by substituting Eqs. (5-8) and (5-10) into Eq. (5-9). This becomes

$$\lambda^j = \log 16 \; P_j - \overline{R} \quad , \quad j = 1, \ldots, 7$$

$$\lambda^8 = \log \frac{16}{9} \; P_8 - \overline{R} \tag{5-12}$$

when the selected values of m and ℓ are inserted into Eq. (5-11).

For a given channel the values of the P_j in Eq. (5-11) are obtained from Eq. (5-7) using the threshold values of E_b/N_o. The value of $\overline{R}$ is, of course, 1 or 2, depending upon the mode of operation.

G. LET ENCODER AND DECODER - BASIC OPERATION

The LET convolutional encoder is a 60-stage shift register which has three parity networks connected to it. Each parity network generates a modulo-two sum of the bits in approximately 30 of the shift register stages. As described in Sec. 5-F, the output of the encoder is a 4-bit symbol whose format depends upon the rate. At 5000 bits/s ($\overline{R} = 1$), a symbol contains one message bit and three parity bits, while at 10,000 bits/s ($\overline{R} = 2$), it contains two message bits and two parity bits. The resulting branch structures were shown in Fig. 5-13 where a new branch is generated every 200 μs. The overall structure of the encoder is that shown in Fig. 5-5(c) with the exception that the commutating switch has 5 positions, 2 for message bits and 3 for parity bits with the selected switch positions controlled by the rate.

The function of the decoder is to determine the transmitted message sequence $M = m_1, m_2, \ldots$ given receipt of the sequence $Y = y_1, y_2, \ldots$ from the receiver bank and listing circuit. The decoder contains within it three major units: (1) a commercial magnetic core memory for the storage of the received data, the message digits m_i', and certain other information used in the decoding process, (2) a replica of the encoder for generating the tree, and (3) the necessary arithmetic and control circuitry for implementing the Fano algorithm of Fig. 5-8.

The machine timing is synchronized to that of the memory such that the time for processing a branch of the tree is approximately the memory cycle time. This is shown in Fig. 5-15. Successive symbols y_i are stored in successive memory registers. Suppose the decoder is performing the comparison of Box A in Fig. 5-8. The symbol y_n stored in a memory register is received from the memory after the memory read interval, approximately halfway through the memory cycle. During the write interval, the indicated comparison of Box A is made by computing λ_n^j from y_n and a generated tree branch x_n^*. The comparison with the threshold is completed shortly after the end of the write interval. A success (failure) implies forward (backward) motion in the tree. In the machine this corresponds to incrementing (decrementing) the

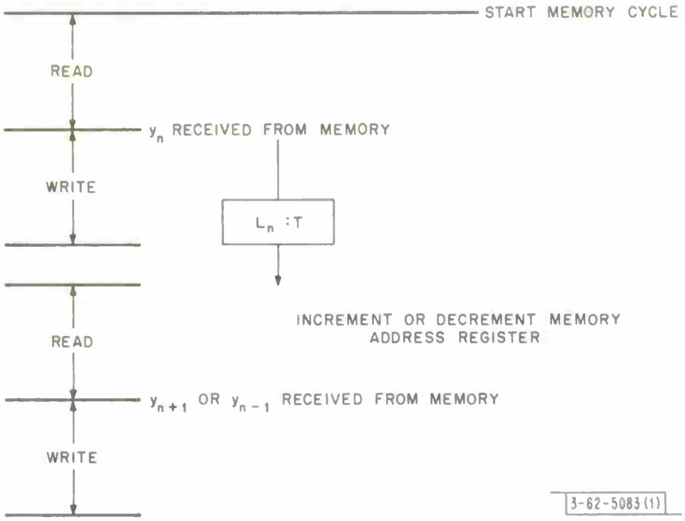

START MEMORY CYCLE
READ
y_n RECEIVED FROM MEMORY
WRITE
$L_n : T$
INCREMENT OR DECREMENT MEMORY ADDRESS REGISTER
READ
y_{n+1} OR y_{n-1} RECEIVED FROM MEMORY
WRITE
3-62-5083(1)

Fig. 5-15. Decoder timing structure.

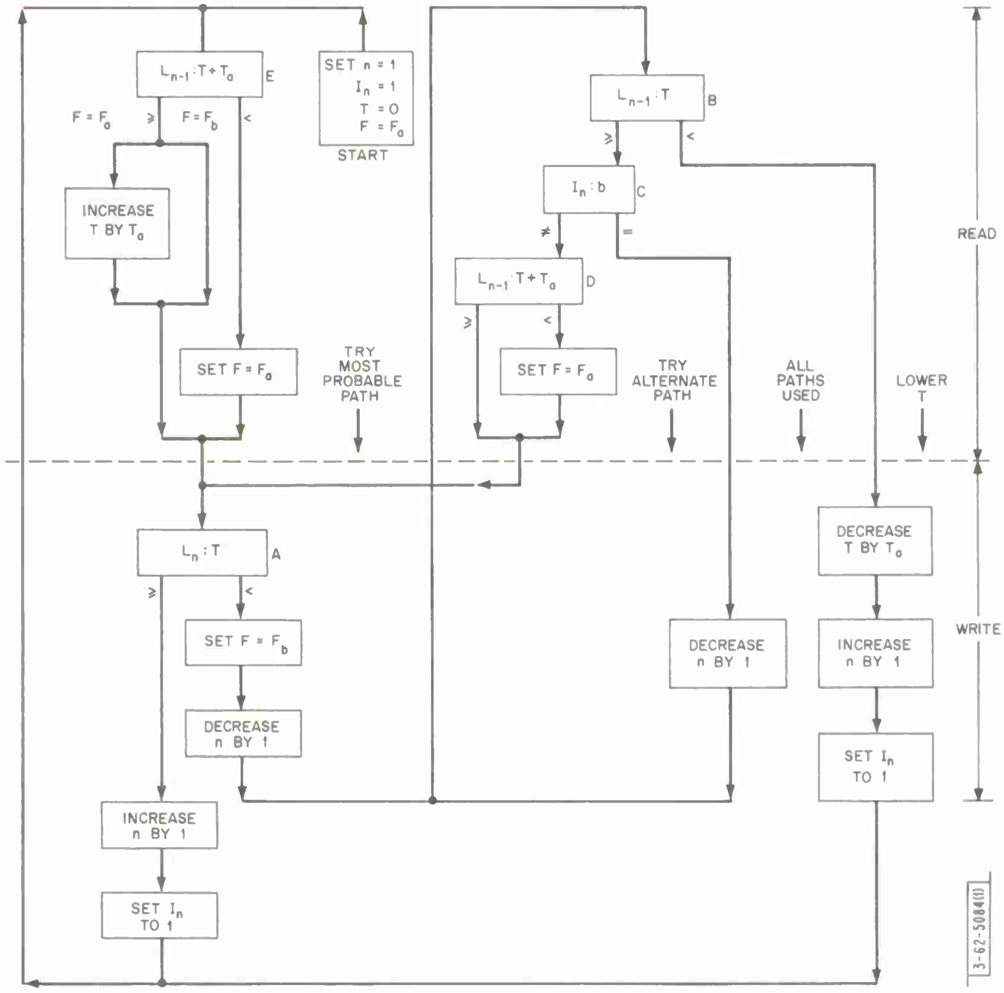


Fig. 5-16. Decoder flow diagram.

memory address register by unity, and then initiating another memory cycle in which $y_{n+1}(y_{n-1})$ is obtained from memory for the next comparison.

The decoding algorithm used in the machine is that of Fig. 5-8 with minor modifications to conform with the timing indicated in Fig. 5-15. This is shown in Fig. 5-16. A comparison of Figs. 5-8 and 5-16 shows that Box A and the left side of Fig. 5-8 appear in the left half of Fig. 5-16, while most of the right side of Fig. 5-8 is in the right half of Fig. 5-16. The dashed horizontal line in Fig. 5-16 partitions the operations in such a way that those above the line occur during the read interval, and those below the line occur during the write interval of the memory cycle. Thus, Box A referred to in Fig. 5-15 appears below the dashed line. All flow branches in the bottom half end with the specification of an increase or decrease in the index n, implying an increment or decrement in the memory address register with an initiation of a new cycle, that on the left after a success and that on the right after a failure.

Three simple differences between Figs. 5-8 and 5-16 should be noted. First, the upper left section of Fig. 5-16, beginning with Box E, is a slightly compressed version of the corresponding section in Fig. 5-8. Only a single threshold increase is allowed by restricting λ_{max} to be less than T_o. Second, this set of operations follows rather than precedes the incrementing of n after a success in Box A. Thus, Box E, in Fig. 5-16 refers to L_{n-1} whereas Box E in Fig. 5-8 refers to L_n. Finally, the exact procedure in retracing a path after lowering the threshold after Box B differs slightly in the two diagrams.

H. DECODER ORGANIZATION

In Fig. 5-17, we show the three sections of the decoder: the core memory, the encoder, and the arithmetic and control units.

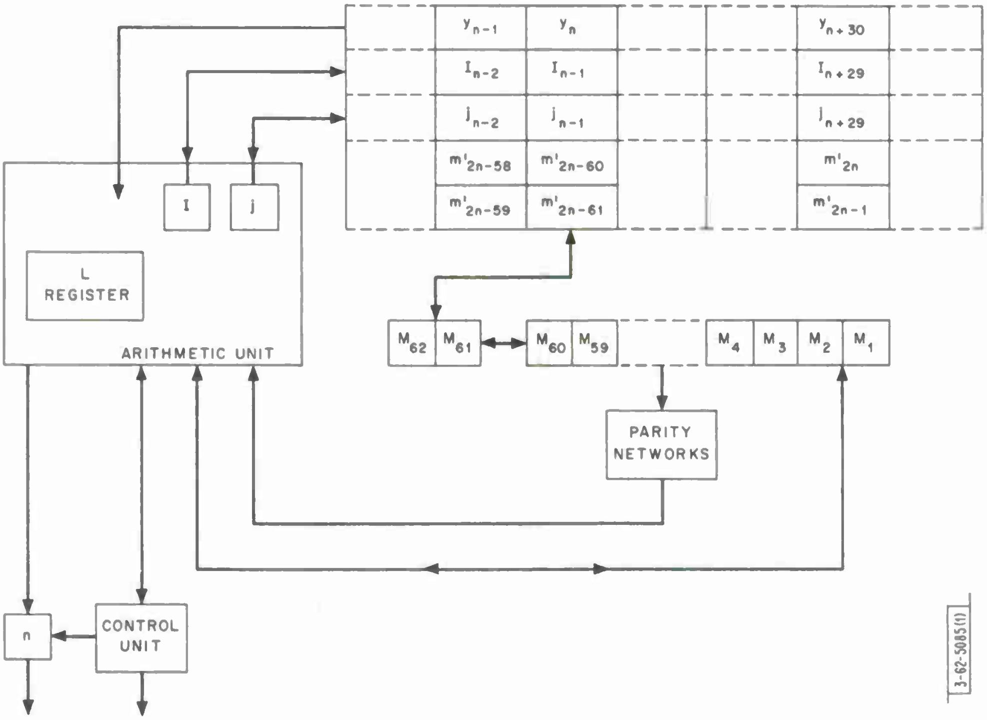


Fig. 5-17. Decoder block diagram.

Each memory word is divided into four parts. One part contains the 28-bit list, $y = b^1, b^2, \ldots, b^7$, which is generated by the receiver. A second part contains 2 bits which represent the variable I of Figs. 5-8 and 5-16 where I is the probability ranking of the branch chosen at a node. Two bits are adequate to represent I, because there are a maximum of four branches at a node. The third part contains the 3-bit number j which indicates the position on the list of the symbol chosen at a node. The fourth part contains 2 bits, the decoded message bits per symbol. Altogether there are 35 bits in each memory word, all of which are used at $\overline{R} = 2$, with 33 of them used at $\overline{R} = 1$. These various parts of a memory word are associated with different received symbols. If the word contains y_n, the list for symbol n, then it contains I_{n-1} and j_{n-1}, the quantities I and j referring to the preceding symbol (n – 1). The symbol associated with the message bits is dependent upon the rate. At $\overline{R} = 1$, the message bit is from symbol (n – 60), while at $\overline{R} = 2$, the message bits are from symbol (n – 30).

Information relating to successive symbols is stored in consecutive memory registers. In the decoder, the memory address register, designated n, plays the role of the index n in the flow diagrams. With this structure, at address n in the memory, we find the list for symbol y_n, I_{n-1}, j_{n-1}, and, at $\overline{R} = 1$, m'_{n-60}. At $\overline{R} = 2$, we will find m'_{2n-60} and m'_{2n-61}.

For control purposes, each memory word is divided into two sections designated Y and H. The Y-section contains the list, and the H-section contains the remaining bits. Each time a node is processed, the Y-section of the corresponding word is read with its value restored in the memory during the write cycle, while the operation in the H-section depends upon whether there was a success or failure at the preceding node. If the cycle on the right side of Fig. 5-16 is being executed, the decoder initiates a read and restore cycle in the H-section, while during execution of the left cycle, new information from outside the memory is written into the three parts of the H-section. As we show below, these bits describe a new path the decoder has taken.

The encoder part of the decoder is similar to the encoder at the transmitting end of the system. It contains a 60-stage shift register with the same parity network connections as the transmitting encoder. Suppose the decoder has examined the first (n – 1) branches of a path and is now testing the nth branch. At $\overline{R} = 2$, the tentatively decoded message bits corresponding to the first n – 30 branches (2n – 60 bits) are stored in memory, and the last 58 bits are in the encoder shift register stages M_3 through M_{60} as shown in Fig. 5-17. At $\overline{R} = 1$, the memory stores the first n – 60 branches or message bits, and the encoder shift register contains the last 59 decoded message bits in stages M_2 through M_{60}.

The basic arithmetic operation performed by the decoder is a comparison such as that of Box A, B, or D, in Fig. 5-16. Here a branch of the tree is compared with a received symbol. A selected branch at a node determines a value of the metric λ associated with the branch; this value of λ is thereupon added to an accumulated metric $L = \Sigma\lambda_i$ of Eq. (5-4), and the resultant value compared with a threshold. These operations are performed in the arithmetic unit of Fig. 5-17. Here are stored the values of y, j, and I received from memory, the accumulated metric and threshold, and the necessary circuitry for performing the comparisons.

To see this in more detail we describe the operation of Box A. Suppose that the list $y = [b^1, b^2, \ldots, b^7]$ is received from memory. At $\overline{R} = 2$, each of the b's may be expressed as a 4-bit number (m_1, m_2, g_1, g_2) where the m's and g's are the possible message and parity bits, respectively. Thus the entire list may be expressed as the matrix

$$\begin{bmatrix} b^1 \\ \\ b^2 \\ \cdot \\ \cdot \\ \cdot \\ b^7 \end{bmatrix} = \begin{bmatrix} m_1^1\ m_2^1\ g_1^1\ g_2^1 \\ \\ m_1^2\ m_2^2\ g_1^2\ g_2^2 \\ \cdot \\ \cdot \\ \cdot \\ m_1^7\ m_2^7\ g_1^7\ g_2^7 \end{bmatrix}$$

The encoder section of the decoder then generates the matrix of branches

$$\begin{bmatrix} a^1 \\ \\ a^2 \\ \cdot \\ \cdot \\ \cdot \\ a^7 \end{bmatrix} = \begin{bmatrix} m_1^1\ m_2^1\ c_1^1\ c_2^1 \\ \\ m_1^2\ m_2^2\ c_1^2\ c_2^2 \\ \cdot \\ \cdot \\ \cdot \\ m_1^7\ m_2^7\ c_1^7\ c_2^7 \end{bmatrix}$$

The parity check digits $c_1^j\ c_2^j$ are obtained from the encoder as functions of the 58 bits in $M_3 - M_{60}$ and the two bits m_1^j and m_2^j obtained from the list. The seven indicated pairs of parity computations are done in parallel, each using the partial parities obtained from $M_3 - M_{60}$.

If I = 1, then the most probable branch is to be selected. This is indicated by the smallest value of j for which $a^j = b^j$. This selected value of j then addresses a table in which the metric values of Eq. (5-12), λ^j, are stored. If none of the list members matches a branch ($a^j \neq b^j$, $j = 1, \ldots, 7$) then a branch is selected at random and λ^8 is obtained from the table.

At $\overline{R} = 1$, a similar computation occurs. Here each list member is of the form $b = (m_1, g_1, g_2, g_3)$. The encoder generates its branches using $M_2 - M_{60}$ together with the value of m obtained from the list.

At the time of these operations, register L in the arithmetic unit contains $L_{n-1} - T$. The selected value of λ is therefore added to the value of L and the sign of the sum is tested. A positive result indicates a success and a negative result a failure. If the result of the Box A comparison is a success, $(L + \lambda_n^j)$ is stored in the L register, the newly determined value of j is stored in the j register, and I is inserted in the I register. The message bits (or bit) for the newly chosen branch are put into stages M_1 and M_2 (stage M_1 at $\overline{R} = 1$), M is shifted left by two (or one), and n is increased by one. The decoder proceeds to the upper left of Fig. 5-16, and starts a new memory cycle at the new address n. The clear and write operation in the H-section of the word stores the contents of the j and I registers which now contain data for node (n − 1) in the memory. It also stores the contents of flip-flops M_{61} and M_{62} in the memory. These flip-flops contain the bits shifted from stages M_{59} and M_{60} of the encoder register after the

Box A comparison. The comparison in Box E is equivalent to comparing L to T_o since, when the threshold must be increased, T_o is subtracted from L.

If the comparison in Box A registers a failure, the decoder goes to Box B after shifting the encoder register right two places at $\bar{R} = 2$ (one place at $\bar{R} = 1$) and decreasing n by one. Note that the j, I and L registers are not changed. With respect to the new value of n, the L register contains $(L_n - T)$ instead of $(L_{n-1} - T)$ needed for the Box B comparison. At the beginning of the new cycle, the decoder arithmetic unit, therefore, has available j_n from which it can determine λ_n^j from its table. λ_n^j is subtracted from L and the result compared to zero. If it is positive, the contents of the I register are compared to b (2 or 4 depending on the rate) to see whether all branches at the node have been tried. If they have not, the decoder tries an alternate path using Box A. The quantities y_n, I_n, and j_n are obtained from memory as before. The message bits (or bit) are transferred from memory into M_{59} and M_{60} (or M_{60}). Tree branches are formed by effectively putting message bit combinations into the right end of the register in the same manner previously described. The decoder chooses the branch with the next highest probability. Using the $\bar{R} = 2$ case, this branch is chosen by comparing the four branches at a node with only those list members which have a higher j than that in the j register. The decoder picks the branch which matches the list member highest (lowest j) on this abbreviated list. If no branches are on the list, a branch is selected at random from among the previously unselected branches.

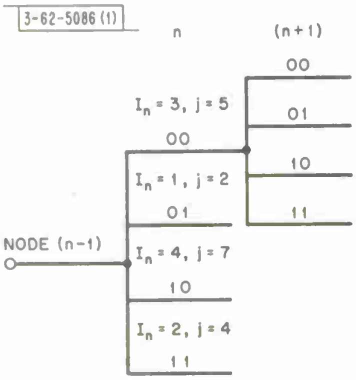


Fig. 5-18. Possible ordering of branches at node n for R = 2.

As an example, suppose the decoder is operating at node n of Fig. 5-18. Here each branch is labeled with the appropriate I and j values. The first time it processes this node, it will choose the (01) branch. Assume that it succeeds and tries unsuccessfully to decode node (n + 1). The decoder therefore backs up to node n, subtracts λ^2 specified by j = 2 from L. If the result is positive, the alternate branch (11) will be tried, because of the remaining branches it has the next highest probability (the next lowest j).

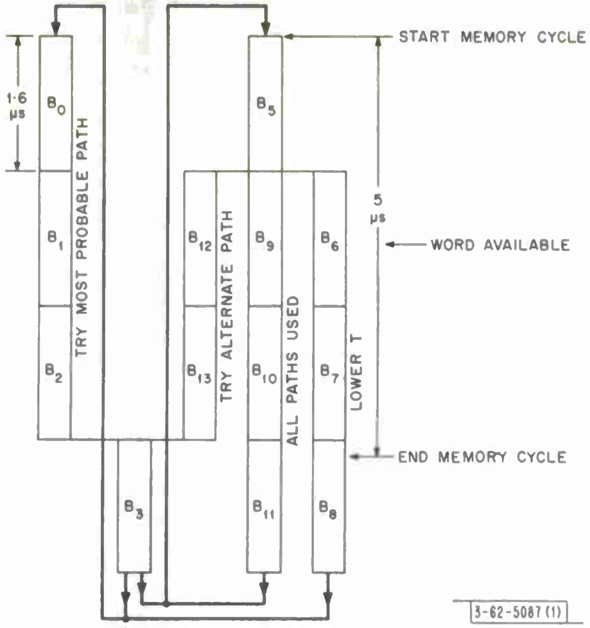
In discussing Box B, we assumed that $(L - \lambda_n^j)$ was positive. If it is negative, the decoder must lower the threshold and retrace its path. Since the L register already contains $(L_{n-1} + \lambda)$ and appropriate message bits are in the encoder register, all that need be done as shown in Fig. 5-16 is lower the threshold (by adding T_o to L) and increase n by one.

I. DECODER CONTROL

The control of the decoder is a sequential circuit with states labeled B_i which initiate the various arithmetic and memory operations described above. The operation of the control is defined by the combined state and timing diagram of Fig. 5-19. A computation cycle, defined in the diagram as a sequence of four states of equal duration, is slightly longer than a memory cycle. The two cycles start concurrently; the read part of the memory cycle ends during the second state, while the write interval ends near the beginning of the fourth state. The states shown in Fig. 5-19 are in correspondence with the operations of the flow diagram of Fig. 5-16.

The start of B_0 initiates a memory cycle at address n. All the operations shown in the top left half of the flow diagram are done either during or at the end of B_0. In general, comparisons and arithmetic operations are performed during a state, while the contents of registers

Fig. 5-19. State and timing diagram.



and flip-flops are changed at the end of a state. A branch is chosen and a metric, λ_n^j, generated during states B_1 and B_2. The branch is tested, and the operations dependent upon it, which are shown in the bottom left half of the flow diagram, are performed during and at the end of state B_3.

The beginning of state B_5 also starts a memory cycle at address n. During B_5, the comparison in Box B is performed. If L_{n-1} is less than T, the threshold must be reduced, and the decoder goes to state B_6. If L_{n-1} is greater than T, I is compared to b. If it is equal to b, the decoder has tried all paths at this node, and it goes back to node (n − 1) by entering state B_9. If I does not equal b, the decoder tries an alternate path by going to state B_{12}.

The decoder marks time in states B_6 and B_7. At the end of state B_8, T_o is added to the L register, I is set to one, and n is increased by one.

When all paths have been used, the decoder marks time in states B_9 and B_{10}. State B_{11} causes a failure with a subsequent reduction in n and a return to state B_5.

When the decoder is going to try an alternate path, it goes to state B_{12} where it decides whether to set $F = F_a$ or leave it alone by performing the comparison in Box D of the flow diagram. At the same time, it begins finding an alternate branch and metric, λ_n^j. It completes this task in state B_{13} and tests the new hypothesis in state B_3.

J. INPUT-OUTPUT AND PERFORMANCE

The sequence of operations defined by the flow diagram and state diagram describe the entire machine with the exception of data input-output. Whenever a new symbol y is available from the receiver, it is stored in a buffer register. At the end of the next computation cycle of the decoder, the sequence is interrupted while the new symbol is stored in the appropriate memory address. During this same memory cycle, the message bit or bits stored at this address are delivered to the user. The main decoding algorithm then continues from the point of interruption.

In Sec. 5-D we pointed out that a sequential decoder must be resynchronized on request or periodically because it will stop whenever it backs up so far (to point P_1, in Fig. 5-10) that the data transferred to a user might be questionable. In LET both the encoder and decoder are reset after every 1000 symbols (a block). The decoder memory stores exactly one block of data in such a way that the nth received symbol y_n of a block is always found in memory address n. When y_n is stored in register n, it replaces the nth symbol of the previous block. During this same cycle, the decoded bits for branch n − 30 (or n − 60) of the previous block are passed on to the user. If follows that the delay in the decoder is 1030 or 1060 symbols (~0.2 s) depending upon the rate.

The message bits of the final 25 symbols of each block are constrained to be zeros, and the decoder uses its knowledge of this to facilitate the completion of the decoding of the block by forcing the decoder to accept only the correct branches during this time. When the message digits are delivered to the user these zeros are, of course, ignored.

The memory utilization described thus allows buffer operation according to Fig. 5-10. The window size N_1 = 1000 branches extends from symbol j of one block to symbol j − 1 of the next. Point P represents the current position n of the decoder. The minimum sequence length $N - W_1$ to guarantee a negligible error probability (as opposed to overflow probability) is set at 120 symbols, or equivalently the maximum waiting line is 880 symbols. If this waiting line is exceeded, the decoder stops delivering data, jumps its address register, and starts decoding at the beginning of the next block.

We can estimate the probability of buffer overflow by making some assumptions about buffer behavior based upon experimental observation and using Eq. (5-5).[7,9] Suppose the decoder is operating at the rightmost end of the buffer of Fig. 5-10. Then the time required for the buffer to overflow is just the time for 880 symbols to enter the decoder. Since the decoder performs

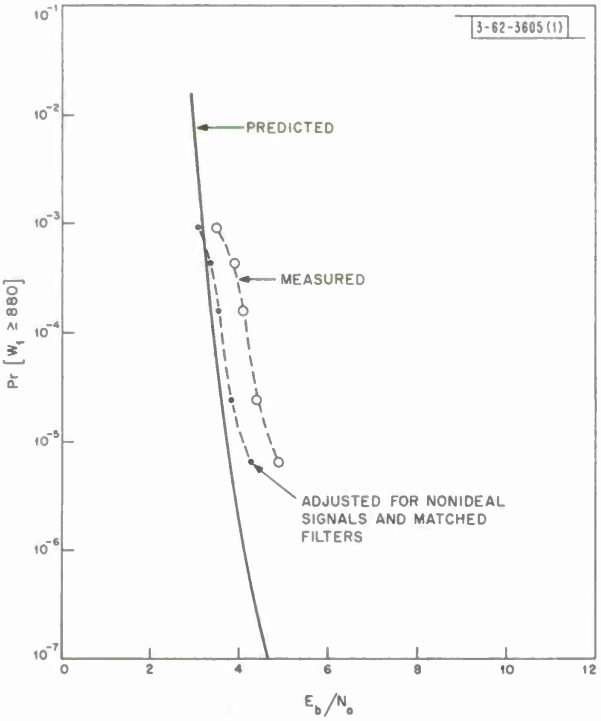


Fig. 5-20. Pr [W_1 ≥880] vs E_b/N_o.

one branch computation per computation cycle of duration μ, it performs τ/μ computations per input symbol time τ, and we may write for the overflow probability

$$P_{of} = P(W > W_1) = P(N > W_1 \tau/\mu)$$

which may be evaluated using Eq. (5-5) and Fig. 5-11.

If the searches are assumed independent from branch to branch, then an overflow may occur with equal probability at any symbol in the block, and the average number of symbols lost due to overflow is 500. The average fraction of symbols lost or the probability of erasure, P_e, is 500 P_{of}.

In Fig. 5-20 we plot P_{of} as a function of E_b/N_o. The solid curve was obtained from a computer simulation of the decoder using the independence assumption of the preceding paragraph. The measured points shown were obtained from the LET system. The essential difference between the simulation and the actual system is the fact that the receivers preceding the decoder are assumed ideal in the simulation; the decoders are identical. When the known departures of the system from the ideal are taken into account, the two curves are in good agreement, verifying the independence assumption.

K. CONSTRUCTION

The complete decoder consists of a commercial core memory and a logic drawer (see Chapter 1, Fig. 1-12) which contains 1980 integrated circuit units that perform all the functions necessary to implement the algorithm. In addition, the drawer contains approximately 75 discrete component level converters for controlling and interfacing with the core memory (see Chapter 1, Fig. 1-13). The integrated circuit units are mounted on small printed circuit boards designed to perform specific logical functions. These are mounted and connected on large printed circuit boards which are joined by under chassis wiring.

ACKNOWLEDGMENTS

A number of people besides the authors were involved in the development of the LET decoder. Here we acknowledge their contributions. The signal processing system, of which the decoder is a part, is due principally to P.R. Drouilhet. The system parameters were chosen by a combination of theoretical and experimental work. The theoretical work was performed by K.L. Jordan, R.F. Kennedy, and J.M. Wozencraft, while the experimental work was done by G. Blustein and K.L. Jordan. G. Blustein also contributed many useful ideas in discussions with the authors. He also wrote a program which simulated the actual decoder; data obtained from the program was used to debug the decoder and is presently used to trouble shoot the decoder. The design of a special list generator built into the decoder for self-testing was based upon helpful discussions with L.M. Goodman. J.J. Drobot was responsible for laying out the bulk of the printed circuit cards in the decoder, as well as the mechanical layout of the decoder drawer.

REFERENCES

1. R. M. Fano, Transmission of Information; a Statistical Theory of Communications (M.I.T. Press, Cambridge, and Wiley, New York, 1961).
2. J. M. Wozencraft and B. Reiffen, Sequential Decoding (M.I.T. Press, Cambridge, and Wiley, New York, 1961).
3. B. E. Nichols and P. Rosen, "Lincoln Experimental Terminal," Chap. 1.
4. P. R. Drouilhet, Jr., "Signal Processing System," Chap. 2.
5. R. M. Fano, "A Heuristic Discussion of Sequential Decoding," IEEE Trans. Inform. Theory IT-9, 64 (April 1963).
6. J. M. Wozencraft and I. M. Jacobs, Principles of Communication Engineering, (Wiley, New York, 1965).
7. G. Blustein and K. L. Jordan, Jr., "An Investigation of the Fano Sequential Decoding Algorithm by Computer Simulation," Group Report 62G-5, Lincoln Laboratory, M.I.T. (July 1963), DDC 412632.
8. J. E. Savage, "The Computation Problem with Sequential Decoding," Technical Report 371, Lincoln Laboratory, M.I.T. (16 February 1965), DDC 621713.
9. K. L. Jordan, Jr., "The Performance of Sequential Decoding in Conjunction with Efficient Orthogonal Modulation," IEEE Trans. Commun. Tech. Com-14, 283 (June 1966).
10. J. Berger and B. Mandelbrot, "A New Model for Error Clustering in Telephone Circuits," IBM J. Research Develop. 7, 224 (July 1963).
11. J. M. Wozencraft and R. S. Kennedy, "Modulation and Demodulation for Probabilistic Coding," IEEE Trans. Inform. Theory IT-12, 291 (July 1966).
12. R. S. Kennedy, Performance Limitations of Fading Dispersive Channels (Wiley, New York, to be published).
13. I. Jacobs, "The Asymptotic Behavior of Incoherent M-ary Communications Systems," Proc. IEEE 51, 251 (January 1963).
14. B. H. Hutchinson, S. B. Russell and J. W. Craig, "Modulation and Demodulation System," Chap. 3.
15. J. Tierney and J. N. Harris, "Channel Vocoder," Chap. 6.

CHAPTER 6
CHANNEL VOCODER

Joseph Tierney and J. N. Harris

ABSTRACT

A channel vocoder has been designed as the speech processor for a satellite communication terminal (the Lincoln Experimental Terminal). The vocoder functions at two separate data rates; a pitch-excited mode (PEV) at 4800 bits/s and a high-rate 9600-bit/s voice-excited (VEV) mode. The low rate takes advantage of a high-quality, dynamic microphone input at or near the vocoder while the VEV permits inputs from remote speakers using the ordinary switched telephone plant.

Each mode has several features worthy of note. The PEV uses a digital pitch extractor along with a combination analog and digital buzz-hiss detector. This mode transmits about 1200 bits/s for excitation and 3600 bits/s for 16 channels. In the VEV mode, bandpass sampling of the 300- to 900-Hz baseband signal results in an excitation rate of about 7200 bits/s. The spectral channel rate is 2400 bits/s for 10 channels. Both modes employ 4-bit coding relative to the instantaneous maximum spectral signal to reduce the size of the quantization steps.

A. INTRODUCTION

To provide a flexible speech signal for the LET system,[1,2*] a dual mode vocoder has been designed. At a rate of 9600 bits/s the vocoder provides a voice-excited vocoder (VEV) operation, while at a 4800-bit/s rate the vocoder operates as a pitch-excited channel vocoder (PEV). Both modes provide a high intelligibility speech channel, close but inferior to the quality of an average telephone circuit. The need for a dual-mode vocoder is clearly seen if we examine the kinds of input speech the LET processes.

B. TWO VOCODER MODES

In general, there are two classes of input speech sources to the LET vocoder. One class consists of local telephone plant input, implying a carbon button transmitter and no control over the line between transmitter and vocoder. In this case the critical frequency band containing useful voice fundamental frequency information may be very badly distorted in amplitude and phase, making the extraction of talker pitch difficult. The need to have a speech compressing processor which is relatively insensitive to the distortions introduced by the local telephone system leads to the voice-excited vocoder.[3] In this mode there is no pitch extraction process, but rather a portion of the input speech baseband from 300 to 900 Hz is encoded and transmitted without compression. The band is processed at the receiver to yield useful pitch information for the voice synthesis. Because of the baseband transmission, a digitized VEV requires about 9600 bits/s for speech transmission.

* Numbered references for Chapter 6 may be found on page 112.

The second class of input to the LET vocoder consists of talkers at the terminal who can use a high quality microphone, or talkers nearby the terminal who can use special input equipment, or at least, a high quality line to the vocoder so that the pitch extractor has to deal with the distortions of a carbon button transmitter alone and not the full repertory of distortions introduced by the carbon button and uncontrolled telephone line. The higher quality inputs associated with this class allow the more "classical" channel vocoder[4] to be used (including the digital pitch extractor previously described). This allows the encoded speech to be transmitted with 4800 bits/s.

Flexibility with regard to input quality in the VEV mode costs an extra 4800 bits/s transmitted. Where we have control over inputs we may do more efficient speech coding and use the lower 4800-bit/s rate. The degradation in going to the PEV mode and its lower rate compared to the VEV mode varies directly with the quality of the input. For very good input quality, the two rates have comparable output quality. As the input quality decreases, the pitch-excited mode begins to produce a poor excitation signal and the quality in this mode decreases faster.

C. SIGNIFICANT PROPERTIES OF LET-VOC

1. PEV

In the 4800-bit/s pitch-excited vocoder mode, standard channel vocoder speech parameters are derived with the configuration shown in Fig. 6-1. Talker pitch is derived and short time spectral envelope measured with 16 channels covering the range from 180 to 3300 Hz. These derived signals are sampled, digitized, and transmitted to the synthesizer which decodes and recreates synthetic output speech. Besides certain circuit configurations, which are novel and interesting, the general vocoder structure has certain interesting features.

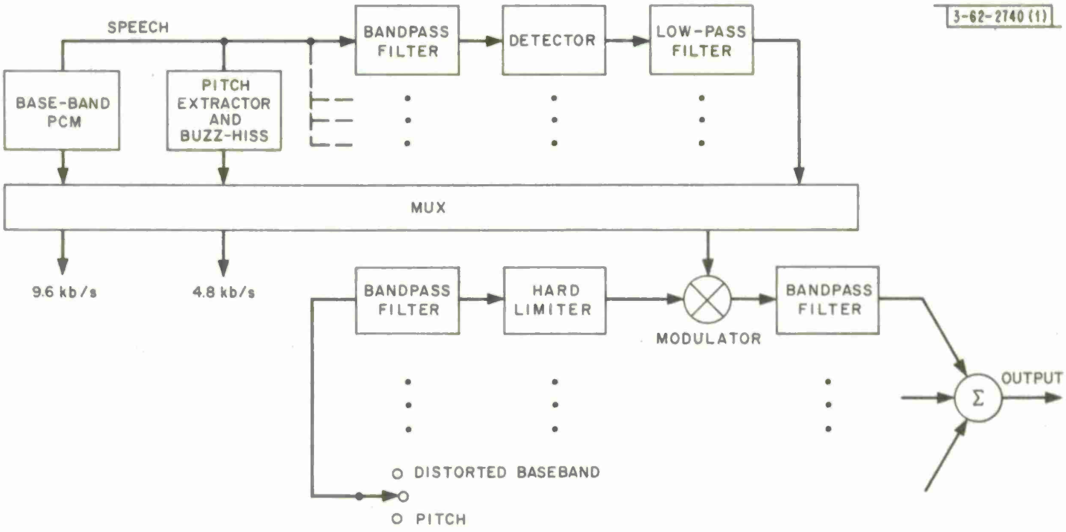
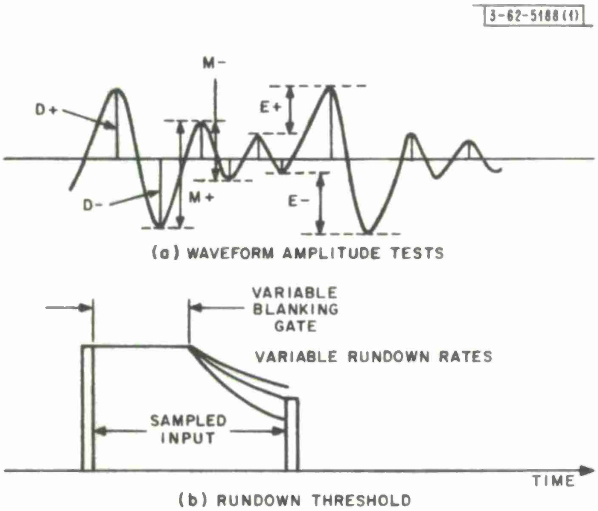


Fig. 6-1. Channel vocoder.

The pitch extractor used to extract spectral fine structure information from the input speech has been described in the literature.[5,6] The extractor consists of three basic operations in cascade: analog processing of the input speech time function to yield amplitude-modulated pulse trains, analog editing of these pulse trains to yield standard pulses marking events in time, and a digital computation which chooses the most likely time interval representing input talker fundamental pitch. Figure 6-2(a) indicates the operations on the input speech wave to produce amplitude-modulated pulses. There are three analog amplitude tests as shown. These tests operate on the input wave and also its negative to produce six separate pulse outputs. Each of the amplitude-modulated pulse trains is edited by a crude pitch detector consisting of a blanking interval and exponentially decaying threshold, as shown in Fig. 6-2(b). A pulse amplitude exceeding the threshold resets the run-down circuit and causes an output pulse to occur. The output pulses from the six crude pitch detectors are then fed into a digital processor which stores the periods between pulses from each detector and computes the most likely pitch period on the basis of certain coincidences between the six input lines. Since there are six separate, although correlated, input lines to work on, the extractor rarely makes severe pitch errors. In addition, the processing delay need not exceed more then one pitch period plus the computation cycle. This total time rarely exceeds 15 ms. Since the extractor works on the time waveform directly, there are no severe spectral window effects to smooth the results.

Fig. 6-2. Pitch detector analog processing.



(a) WAVEFORM AMPLITUDE TESTS

(b) RUNDOWN THRESHOLD

The extractor produces an 8-bit period estimate linearly encoded with the least significant bit representing an 80-μs increment. This 8-bit word is updated every 3.0 ms. At the receiver synthesizer, the 8-bit word is stored in a buffer register which is then counted down by one every 80 μs. When the register reaches zero, a pulse is emitted and the register is reset to the latest pitch word. This produces a pulse train of 100-μs pitch markers which are spaced in time by the 8-bit period word.

Along with the pitch extraction the PEV must make a voiced-unvoiced (or buzz-hiss) decision. In the LET vocoder this is done with a hybrid buzz-hiss detector that uses both analog thresholds and information from the pitch extractor about the periodicity of the input speech waveform. The periodicity buzz-hiss strategy has been described in the literature.[7] Basically, the digital computation for pitch period yields a measure of how close the six separate pitch detector inputs are to one another. If there is a large number of period coincidences, there is a high probability that the input time function is periodic. Conversely, if there are few coincidences the input is probably aperiodic. It is possible to use a strategy which takes into account the distribution of coincidences in the pitch detector to yield a digital buzz-hiss decision. This decision is combined logically with two analog thresholds. The overall B-H decision logic is shown in Fig. 6-3. A bandpass filter from 180 to 900 Hz prefilters the input to two analog threshold circuits. Threshold circuit 1 is set at some level below which the signal is considered a silence or definitely unvoiced because of the lack of energy in the band from 180 to 900 Hz. Threshold circuit 2 is set about 20 to 30 dB above the first and indicates signals that are almost always voiced because of the amount of energy in the band. In the region between the two analog detectors the digital measure is used as the deciding element to separate plosive and aspirate sounds which are unvoiced but contain large amounts of energy and would exceed simple analog thresholds. The complete detector then combines the digital decision and the low threshold decision in a logical AND circuit and the output of this is combined in a logical OR circuit.

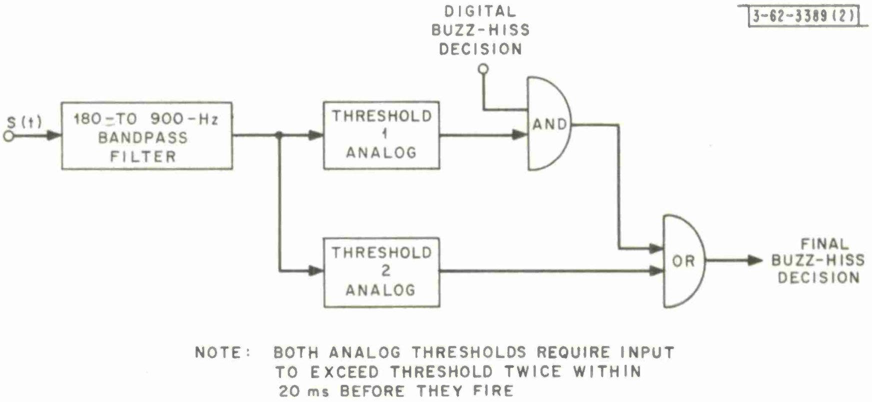

3-62-3389 (2)
DIGITAL BUZZ-HISS DECISION
S(t)
180-TO 900-Hz BANDPASS FILTER
THRESHOLD 1 ANALOG
AND
THRESHOLD 2 ANALOG
OR
FINAL BUZZ-HISS DECISION
NOTE: BOTH ANALOG THRESHOLDS REQUIRE INPUT TO EXCEED THRESHOLD TWICE WITHIN 20 ms BEFORE THEY FIRE

Fig. 6-3. Buzz-hiss decision logic.

At the vocoder synthesizer the excitation pitch pulses are not simply applied to modulators for spectrum weighting. The pulses are fed first into a linear filter which is simply four resonant circuits in cascade as shown in Fig. 6-4. The resonances are placed roughly at 600, 1500, 2200 and 3000 Hz. Each resonance is about 120 Hz wide. The purpose of the filter is to impart a more dispersive "vocal tract like" phase characteristic to the excitation pitch pulses which have a very coherent phase.[8] The output of the crude vocal tract analog filter is then fed into the excitation processing vocoder synthesizer used in the VEV. The filter bank/limiter bank processing flattens out the input pulse spectrum but tends to retain the formant-like phase structure we have introduced. The need for a VEV type of synthesizer for pitch excitation has been discussed elsewhere.[4,9] Its presence in the LET vocoder allows us to use this simple filter technique to reduce the harsh "buzzy" quality associated with pitch excitation.

Before the output of the formant filter is applied to the excitation filter bank, a quantity of white noise is added to this output. The amount of noise added must insure enough energy in

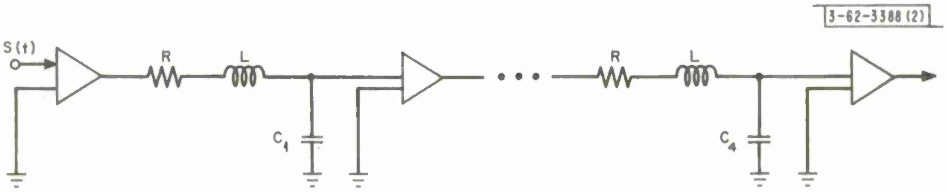


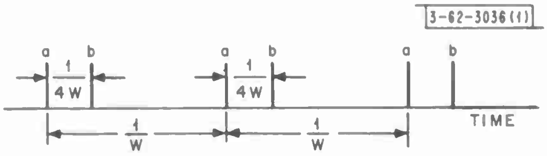
Fig. 6-4. Vocal tract analog filter.

each excitation filter output to suppress power supply noise and cause the limiter to switch fully during unvoiced sounds. At the same time the additive noise must cause no discernible degradation of the pitch signal for voiced sounds. For the LET vocoder, a signal-to-noise ratio out of each excitation filter for peak harmonic output/rms noise of the order of 50 dB satisfies these two conflicting conditions. An alternate solution would provide for noise gating on the buzz-hiss signal. This has not been necessary.

2. VEV

In the VEV mode, a band of frequencies from about 300 to 900 Hz is sampled and digitized with no compression and the band from 900 to 3000 Hz is vocoded and compressed in the conventional method with 10 channels. Frequencies below 300 Hz may not be reliably present in general telephone plant output. Thus the choice of 300 to 900 Hz for the transmitted baseband guarantees the presence of a talker fundamental if it is above 300 Hz or at least two harmonics if it is below 300 Hz. At the receiver, the baseband is distorted by half-wave detection and differentiation to generate higher harmonics for synthesizer excitation. In order to sample and reproduce the baseband and 10 channels within a 9600-bit/s format, it is necessary to encode the baseband in about 7200 bits/s. From computer simulation experiments, it appears that 6-bits/sample, logarithmic encoding is needed for undegraded reproduction of the 50- to 60-dB dynamic range signal. This leads to a sampling rate of the order of 1200 Hz. For a straightforward low-pass sampled function of bandwidth W Hz, we need a sampling rate of 2W. For our baseband this is 1800 Hz. Fortunately, we can use a more efficient sampling which is relatively simple to implement.[10] For a baseband of 600-Hz width, and center frequency of 600 Hz, a sampling consisting of two 600-Hz pulse trains separated by one quarter cycle will adequately reproduce the VEV baseband. The operation is shown in Fig. 6-5. Furthermore, at the synthesizer, both pulse sample trains can be played into the same desampling filter, i.e., a bandpass filter from 300 to 900 Hz to recreate the baseband. The filter must be carefully designed to have rejection of at least 35 dB at 300 and 900 Hz and beyond. The transmitted baseband then requires 2 × 6 × 600 = 7200 bits/s. The VEV samples are encoded in 5 bits (32 levels, 1.75 dB/level) plus a sign bit. Since the band from 300 to 900 Hz is available at the synthesizer, only 10 channels from 900 to 3300 need be transmitted in the remaining 2400 bits/s.

Fig. 6-5. VEV baseband sampling.



3. Common Spectral Measurements

The spectrum from 180 to 3300 Hz is covered with 16 bandpass filters. Each bandpass filter is followed by a half-wave linear detector feeding a 3-pole low-pass filter. The outputs from the low-pass filters are sampled at 50 Hz and all 16 channels are encoded into one of 32 logarithmically spaced levels (1.75 dB/level) essentially simultaneously and stored in an 80-bit channel buffer. The simultaneous sampling eliminates the skew or delay dispersion associated with sampling spread over the entire 20-ms framing interval. As each channel is fed out in the multiplex frame, it is subtracted from the maximum channel at that sampling time and the difference is transmitted in only 4 bits. This means that a window of about 28 dB is encoded and

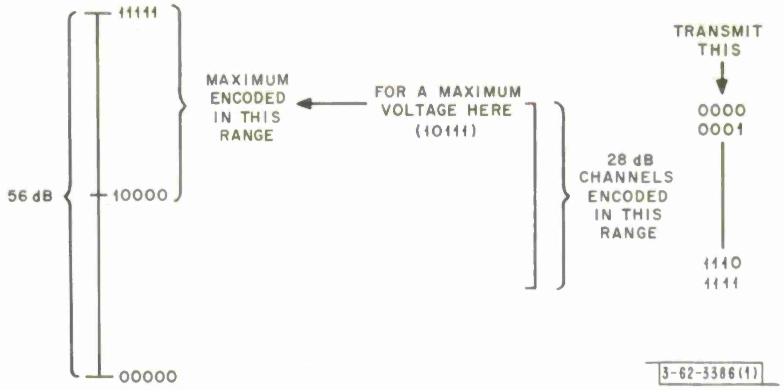

11111
10000
00000
56 dB
MAXIMUM ENCODED IN THIS RANGE
FOR A MAXIMUM VOLTAGE HERE (10111)
28 dB CHANNELS ENCODED IN THIS RANGE
TRANSMIT THIS
0000
0001
1110
1111
3-62-3386(1)

Fig. 6-6. Dynamic maximum channel encoding.

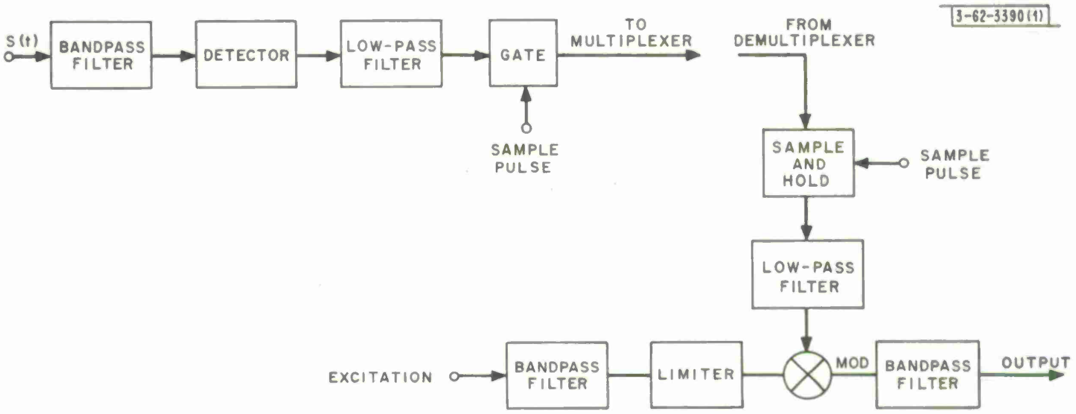


Fig. 6-7. Single vocoder channel.

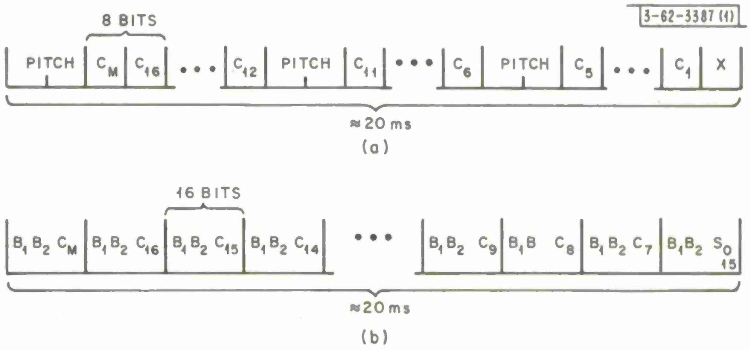


Fig. 6-8. Multiplex formats.

transmitted in any frame, but a 56-dB range is possible over all. The maximum channel is also transmitted with 4 bits. This can be seen more clearly in Fig. 6-6. This kind of dynamic encoding to effectively increase the encoding range for a fixed code has been used in vocoder systems designed by C. P. Smith, L. Cassel, and others.* Sampling all spectrum channels at the same instant instead of spreading the sampling across the frame as needed has been indicated to reduce reverberant effects.

In both modes of the synthesizer, the 4-bit spectrum channel signals are subtracted from the 4-bit maximum signal to give a 5-bit code. This is converted to an analog level which is pulsed into a channel sample-and-hold circuit. A typical vocoder spectrum channel is shown in Fig. 6-7.

D. MULTIPLEX FORMATS AND TECHNIQUES

All the digital multiplexing and timing circuits have been designed using microelectronic modules. Both multiplexing modes are designed to feed data in 16-bit groups to the LET Univac 1218 computer[11] for encoding and formatting.

1. PEV

In the PEV 4800-bit/s mode, 16 spectrum channels of 4 bits each, 1 maximum channel of 4 bits, and three 8-bit pitch words are transmitted in a 20-ms frame. The 8-bit pitch word from the pitch computer is sampled every 6.6 ms. The pitch words are uniformly spaced in the frame and there is a blank space at the end for use by the computer as shown in Fig. 6-8(a). When the pitch detector indicates the input speech to be unvoiced (silence and hiss) the multiplexer places the word 11110000 in the first pitch slot and 10100000 in the second and third pitch slots. These are impossible pitch periods. Besides indicating an unvoiced situation they are also the basis of frame synchronization in the PEV mode. In order to identify 4-bit words in the frame, the vocoder receiver searches for the 11110000 word, then searches for 10100000 at the right distance away twice. The second encounter with 11110000 resets the main receiver counter which remains stable until the search circuits find the 4-word hiss sequence in a different position, at which time the main counter will again be reset.

2. VEV

In the VEV 9600-bit/s mode, 10 spectrum channels from 900 to 3300 Hz, 4 bits each, 1 maximum channel of 4 bits, and 12 double samples of 12 bits each are transmitted in a 20-ms frame. The format is as shown in Fig. 6-8(b). The last 4-bit slot is used for a sync word which is simply 1111 or 0000 alternating. This is an unlikely event for a spectrum channel since it implies the channel will alternate full OFF to full ON each frame. In this mode the computer identifies the bits within one of the twelve subframes but does not provide frame sync. The sync circuits in this case search for 4 alternating 0000's and 1111's in successive subframes 12 apart. When this occurs, a sync flip-flop locks up the search until an error is detected. When an alternation is missed the search mode starts once more. Both VEV and PEV sync strategies have been designed for the error-free output of the LET sequential decoder.[12] Their effective operation in a noisy environment is not clear.

* Private communication.

3. Features Common to Both Modes

The changeover from PEV to VEV or back is effected with one toggle switch on the vocoder transmitter and another one on the vocoder receiver. These toggles change the control logic; they gate out the lower six spectrum channels and pitch detector in the VEV mode, or gate out the baseband sampler in the PEV mode.

Both receiving modes can lock up their sync search modes on an alarm from the station decoder so as to coast through burst errors[12] without resynchronizing.

E. DESIGN OF CIRCUITS

In the design of circuits for speech processing use, there are two phenomena that must be borne in mind at all times. First, there are many channels of essentially identical elements in the vocoder, all of whose outputs are summed together to form the synthetic speech. Consequently, one must be careful in the design of a single channel regarding drift, leakage output, noise, and distortion, so that the final output is acceptable. Second, although for many purposes the vocoder can be looked upon as a linear device, the designer cannot be sure how several different distortions taken together will be treated subjectively and must proceed cautiously with his design compromises.

The vocoder bandpass filters which make up three banks covering the band from 180 to 3300 Hz are designed from Bessel parameter low-pass prototypes. They were chosen after an intensive simulation study[13] and all three banks are electrically identical. They have adequate frequency discrimination, excellent single-lobed impulse responses, and have been designed to cover the band from 180 to 3300 Hz with an envelope delay within 1 ms between any two filters. They cover the band as follows: 10 third-order filters 120 Hz wide crossing at 3 dB (180 to 1380), 3 sixth-order filters 260 Hz wide crossing at 3 dB (1380 to 2160), 3 ninth-order filters 380 Hz wide crossing at 3 dB (2160 to 3300). The filters were synthesized passively. The step in design from 3- to 6-pole filters at the 1380-Hz crossover point introduces a phase difference at crossover of close to 180°, and consequently a severe amplitude dip. This phase difference is measured through both excitation filter bank and final synthesis bank in cascade, since phase is additive through the vocoder synthesizer from excitation input to final input. The simple expedient of reversing the excitation drive connections to the filter above 1380 Hz reduced this dip to less than 3 dB (an old vocoder remedy which has not lost its efficacy). The low-pass filters used to smooth the half-wave detected bandpass signal are also Bessel parameter filters. They are third-order low-pass filters whose 3 dB point is set at 20 Hz. This same design is used for desampling filters on the channel signals after the synthesis sample-and-hold circuits. The filter is synthesized actively, using a single operational amplifier to keep the physical element size small.

Wherever possible the active linear circuits have used a common operational amplifier design[14] with minor modifications to suit a particular end. For example, the low-pass filter operational amplifier and sample-and-hold amplifiers use a special temperature-compensated input stage[15] to reduce DC drift, while the limiter circuits are the same amplifiers with no temperature compensation, used open loop with clamp diodes. All DC drifts and outputs in amplifier, modulators and gates have been kept to ±0.5 mV to insure low output noise and low error at low ranges. The overall dynamic range including the quantizing and digitizer circuits is 56 dB in 1.75-dB steps for both VEV baseband samples and spectral coefficient samples.

Careful analog design techniques including magnetic shielding where necessary and differential common mode rejection have kept the overall background noise in a rather noisy 400-Hz environment to 2 mV peak to peak. Since peak signal levels are designed for a 10-V peak-to-peak swing, the overall signal-to-noise ratio is about 74 dB.

As mentioned, all digital timing and multiplexing functions as well as the digital pitch computer have been designed with integrated circuit elements to reduce the overall size of the vocoder. In line with this reduction, equalizing delays used for the voice and pitch excitation have been introduced digitally rather than with lumped parameter audio delays. New design techniques make it possible to construct[16] nonminimum phase filters to introduce the required delays in the VEV mode for example. This is an alternate approach not used in LET.

F. RESULTS

The combination of the flexible coding and modulation techniques of the LET and the flexibility of the dual-mode terminal vocoder result in a good quality speech communication link highly reliable, error free, resistant to jamming and flexible with regard to talker input quality. The dual-mode vocoder capable of functioning as a 9600-bit/s VEV or 4800-bit/s PEV occupies 21 inches of rack space in the transportable terminal.

REFERENCES

1. B.E. Nichols and P. Rosen, "Lincoln Experimental Terminal," Chap. 1.
2. P.R. Drouilhet, Jr., "Signal Processing System," Chap. 2.
3. E.E. David, Jr., M.R. Schroeder, B.F. Logan and A.J. Prestigiacomo, "Voice-Excited Vocoders for Practical Speech Bandwidth Reduction," IRE Trans. Inform. Theory 8, S101 – S105 (1962).
4. J. Tierney, B. Gold, V. Sferrino, J. Dumanian and E. Aho, "Channel Vocoder with Digital Pitch Extractor," J. Acoust. Soc. Am. 36, 1901 – 1905 (1964).
5. B. Gold, "Description of a Computer Program for Pitch Detection," Paper G34 in Proceedings of the Fourth International Congress on Acoustics, Copenhagen, 1962 (Organization Committee of the 14th ICA and Harland and Toksvig, Copenhagen, 1962).
6. N.L. Daggett, "A Computer for Vocoder Pitch Extraction," TN-1966-3, Lincoln Laboratory, M.I.T. (18 February 1966), DDC 629361.
7. B. Gold, "Note on Buzz-Hiss Detection," J. Acoust. Soc. Am. 36, 1659 – 1661 (1964).
8. B. Gold, "Experiment with Speechlike Phase in a Spectrally Flattened Pitch-Excited Channel Vocoder," J. Acoust. Soc. Am. 36, 1892 – 1894 (October 1964).
9. B. Gold and J. Tierney, "Pitch-Induced Spectral Distortion in Channel Vocoders," J. Acoust. Soc. Am. 35, 730 – 731L (1963).
10. B. Gold and J. Tierney, "A Digitized Voice-Excited Vocoder for Telephone-Quality Inputs, Using Band Pass Sampling of the Base Band Signal," J. Acoust. Soc. Am. 37, 753 – 754L (April 1965).
11. F.E. Heart and W.R. Crowther, "Computer System," Chap. 4.
12. I.L. Lebow and P.G. McHugh, "A Sequential Decoding Technique and Its Realization in the LET," Chap. 5.
13. C.M. Rader, "Study of Vocoder Filters by Computer Simulation," J. Acoust. Soc. Am. 36, 1023A (1964).
14. J.N. Harris and K.E. Perry, "Characteristics and Applications of Solid State Operational Amplifiers," Group Report 62G-2, Lincoln Laboratory, M.I.T. (23 July 1963), DDC 413473.
15. J.N. Harris and R.S. Orr, "Input Current Compensation for Transistor Operational Amplifiers," TN-1964-31, Lincoln Laboratory, M.I.T. (9 January 1964), DDC 431781.
16. R.M. Lerner, "Band Pass Filters with Linear Phase," Proc. IEEE 52, 249 – 268 (1964).

DOCUMENT CONTROL DATA – R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)*
Lincoln Laboratory, M.I.T.

2a. REPORT SECURITY CLASSIFICATION
Unclassified

2b. GROUP
None

3. REPORT TITLE
The Lincoln Experimental Terminal

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*
Technical Report

5. AUTHOR(S) *(Last name, first name, initial)*

Craig, John W.
Crowther, William R.
Drouilhet, Paul R., Jr.
Harris, John N.
Heart, Frank E.
Hutchinson, Ben H.
Lebow, Irwin L.
McHugh, Paul G.
Nichols, Burt E.
Rosen, Paul
Russell, Stephen B.
Tierney, Joseph

6. REPORT DATE
21 March 1967

7a. TOTAL NO. OF PAGES
120

7b. NO. OF REFS
63

8a. CONTRACT OR GRANT NO.
AF 19(628)-5167

b. PROJECT NO.
649L

c.

d.

9a. ORIGINATOR'S REPORT NUMBER(S)
Technical Report 431

9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*
ESD-TR-67-182

10. AVAILABILITY/LIMITATION NOTICES

This document has been approved for public release and sale; its distribution is unlimited.

11. SUPPLEMENTARY NOTES
None

12. SPONSORING MILITARY ACTIVITY
Air Force Systems Command, USAF

13. ABSTRACT

Lincoln Laboratory has designed and constructed three satellite communications ground terminals (Lincoln Experimental Terminals) as part of its program in space communications. The first of these (LET-1) is a self-contained, transportable, X-band terminal in which the RF equipment, antenna and mount are on one vehicle, and the signal processing equipment, prime power and operating console are in a second van. The other two systems (LET-2 and LET-3) consist of signal processors identical to that of LET-1. They are similarly housed in vans and are used in conjunction with antennas and RF equipment associated with other terminals. The three systems were completed in 1965 and tested extensively since that time, using active satellite repeaters and the moon as a passive reflector.

The function of the terminals is to demonstrate satellite communication in which multiple-access and anti-jam performance are emphasized. The use of vocoders for speech compression and an efficient modulation and coding system for signaling result in a low required satellite power per access. Bandspreading by pseudorandom frequency hopping provides the desired multiple-access and anti-jam capabilities.

This report is a description of selected portions of the system emphasizing the most novel features. The first chapter gives a general description of LET-1 as a whole, and the second an over-all description of the signal processing system. Chapter 3 discusses the equipmental realization of the signal processing system and Chapter 4 the utilization of a general-purpose computer as an element of the signal processing system. Chapters 5 and 6 treat two specific subsystems: the sequential decoder and the vocoder.

14. KEY WORDS

LET
vocoder
computers
computer applications
signal processing
coding
communications
satellites
modulation
anti-jamming

Printed by
United States Air Force
L. G. Hanscom Field
Bedford, Massachusetts