Communications Research Centre

FORWARD ERROR-CORRECTION FOR THE AERONAUTICAL SATELLITE COMMUNICATIONS CHANNEL

by

A. SEWARDS, L. BEAUDET AND H. AHMED

CRC TECHNICAL NOTE NO. 698

Department of Communications
Ministère des Communications

OTTAWA, FEBRUARY 1979

COMMUNICATIONS RESEARCH CENTRE

DEPARTMENT OF COMMUNICATIONS
CANADA

FORWARD ERROR-CORRECTION FOR THE AERONAUTICAL SATELLITE COMMUNICATIONS CHANNEL

by

A. Sewards*, L. Beaudet** and H. Ahmed**

(Space Technology and Applications Branch)

* Department of Communications, Communications Research Centre

** Miller Communications Systems Limited

CRC TECHNICAL NOTE NO. 698

February 1979
OTTAWA

CAUTION

This information is furnished with the express understanding that:
Proprietary and patent rights will be protected.

TABLE OF CONTENTS

FORWARD ERROR-CORRECTION FOR THE AERONAUTICAL SATELLITE COMMUNICATIONS CHANNEL

by

A. Sewards, L. Beaudet and H. Ahmed

ABSTRACT

The paper identifies the effects on an L-band aeronautical satellite communications channel of ocean-surface multipath and shows that data messages transmitted over the channel exhibit unacceptably high error-rates under typical conditions. Channel characteristics and techniques for reducing the error-rate are discussed, leading to the choice of a diffuse threshold-decodable convolutional forward error-correcting code. Two implementations of coder/decoders for this code are described, one using standard IC logic, and the other a Z-80A microprocessor. Results of tests of the IC coder/decoder with noise, data error bursts, and a system simulator are quoted, which show that the codec performed as expected and is capable of reducing the error-rate to 10^{-5} or better.

1. INTRODUCTION

In 1974 a Memorandum of Understanding setting up the AEROSAT program was signed between ESRO, U.S.A. and Canada. This program envisaged the launching of satellites in geostationary orbit to provide communications between aircraft on Atlantic routes and ground stations in North America and Europe. Communications between aircraft and satellite would be in the 1550-1650 MHz L-band, with the satellite/ground backhaul at 5000-5250 MHz. In the same year, the ATS-6 satellite was launched by NASA, and a co-ordinated program of L-band experiments involving ESRO (later ESA), FAA, U.S. Coastguard and Canada commenced (NASA 1973). These experiments had as their objective the measurement of the properties of the transmission link between aircraft and satellites, including propagation and multipath effects, and measurement of the performance of candidate voice and data (digital) modems (channel units) over typical links. Tests were also conducted on several designs of aircraft antenna to determine their characteristics including their ability to discriminate against multipath.

Paper originally presented at AGARD Symposium on Digital Communications in Avionics, June 1978

The results of this experimental program were very revealing. It was demonstrated that the error performance on a typical aeronautical satellite communications channel at these frequencies was grossly different from the classical free-space channel often assumed for satellite communications, due to the interference arising from multipath reflections from the surface of the earth, notably the ocean surface. Since multipath interference arrives from a different direction, it can, to some extent, be discriminated against by judicious design of the aircraft antenna, but even in this case the error probability for digital transmissions does not fall exponentially with increasing signal-to-noise ratio. The channel power required under such conditions to achieve an error probability of better than 10^{-5} at information rates of 1200 and 2400 bps may be prohibitive.

The performance of the aeronautical channel can be improved by applying one or more of the following techniques: automatic repeat request (ARQ) followed by retransmission of all or parts of the message, channel diversity, more complex receivers optimized to work in the presence of multipath, and forward error correction coding. The latter offers, perhaps, the most attractive solution in terms of hardware, bandwidth, cost and system design.

Many of the well-known coding techniques are not suitable for the aeronautical channel because of the fact that the effect of multipath is to produce fading which results in bursts of errors. A classical random error-correcting code designed for a memory-less channel must be very powerful to allow correction where a large percentage of consecutive bits are in error. Burst error-correcting codes may be more efficient as they can be designed to trap and correct isolated error bursts. However, the classical burst model is probably not appropriate either, since it is not highly probable that the necessary guard space on either side of the burst will be free from scattered errors.

The channel model appropriate to the aeronautical satellite channel and the choice of a suitable error-correcting code is discussed in the paper. It is concluded that a diffuse convolutional code offers the best compromise between performance and complexity. Two implementations of such a code are described, using random logic and a microprocessor, together with results of tests including tests using an AEROSAT channel simulator. In particular, using such a code, the desired error probability of 10^{-5} at an information bit rate of 1200 bps with a channel quality of 43 dBHz can be met with a typical aircraft antenna signal-to-interference ratio of 10-13 dB.

[Note: *The AEROSAT program, while not abandoned, is currently in abeyance while the signatories of the Memorandum of Understanding and other interested parties re-examine the requirements for ground-air communications and alternative means of meeting them in the short term. A satellite system is expected to be used eventually.*]

2. CHANNEL MODEL

The L-band communications channel between aircraft and satellites can be characterized as a channel with the theoretical free-space loss plus effects due to ionospheric propagation and earth-surface multipath. The former is generally covered by adding a margin to compensate for periods of signal attenuation: in the case of the L-band aeronautical channel, a margin of 2 dB will only be exceeded 0.1% of the time. Earth-surface multipath is, however, more difficult to describe and it is only since the results of the joint L-band international tests using the ATS-6 satellite were published that an adequate understanding of its effects has been obtained (Schroeder, E.H., 1976, Chinnick, J.H., 1977, Brown, D.L., 1976).

Earth-surface multipath represents the sum of the direct signal between aircraft and satellite and the signal which is reflected from the surface of the earth. Because aircraft antennas at L-band are by their nature (if not their design) relatively wide beam, for satellite elevation angles of 15° or less a significant reflected signal is received. Due to the better and more consistent reflection characteristics of the sea surface as compared with the land surface, this problem is worse over the ocean and for the remainder of this Tech Note we will concentrate on the question of ocean-surface multipath. The effect of the reflection is to present an interfering signal to the receiver whose magnitude can range from zero up to about 6-8 dB less than the direct signal, and which is modified by the reflection properties of the reflecting surface. This surface produces specular and diffuse reflections and as a result, the reflected signal is delayed by the path length difference (5-25 microseconds), spread in time and frequency (1 microsecond and 10-100 Hz), and doppler shifted due to differential movement by up to about 20 Hz. The combined direct and reflected signals have been found to affect receivers in different ways; however, in general, it can be stated that voice modulation schemes were little affected but digital data PSK transmissions were seriously perturbed. The general shape of the error curves resulting from ocean-surface multipath is shown in Figure 1, where it can be seen that the normal bit-error-rate (b.e.r.) curve is flattened out at high signal-to-interference (S/I) ratios to the point where the b.e.r. is almost independent of channel signal-to-noise ratio (C/N_o or E_b/N_o). It was found that, in many typical aeronautical satellite communications conditions, a b.e.r. of better than 10^{-3} was rarely achieved. Examples of measured results for CPSK obtained by Schroeder, E.H., 1976 are given in Figure 2.

The principal cause of channel errors is destructive interference between the direct and reflected signals, i.e., fades. Due to the nature of the reflected signals, as noted above, these fades do not last long and measurement statistics indicate that the probability of a burst of more than seven consecutive bit errors at a data rate of 1200 bps is very low, as shown in Figure 3 from Schroeder, E.H., 1976. However, unlike some other fading channels, there is not a high probability of an error-free space each side of the burst.

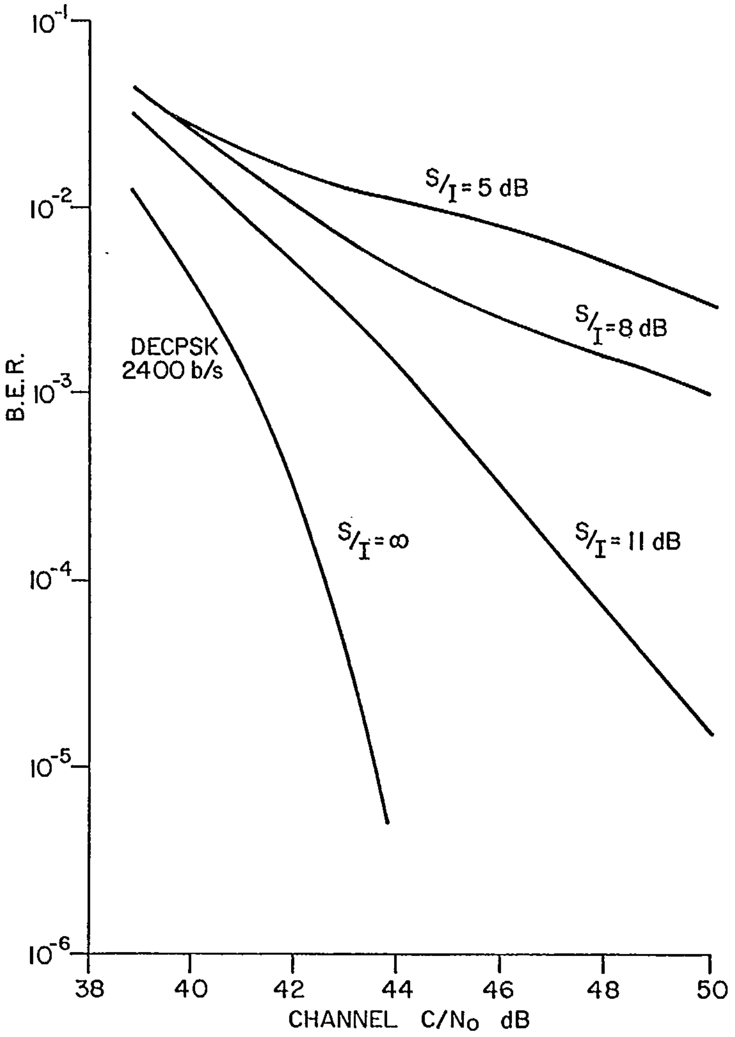


Figure 1. Effect of Multipath on Aeronautical Satellite Communications Data Transmission (Measured Results)

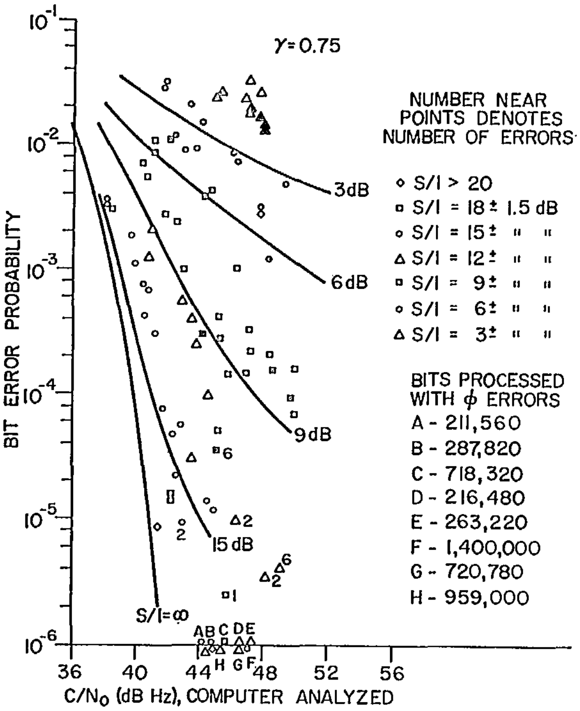


Figure 2. Bit-Error-Rate Performance, CPSK Demodulator

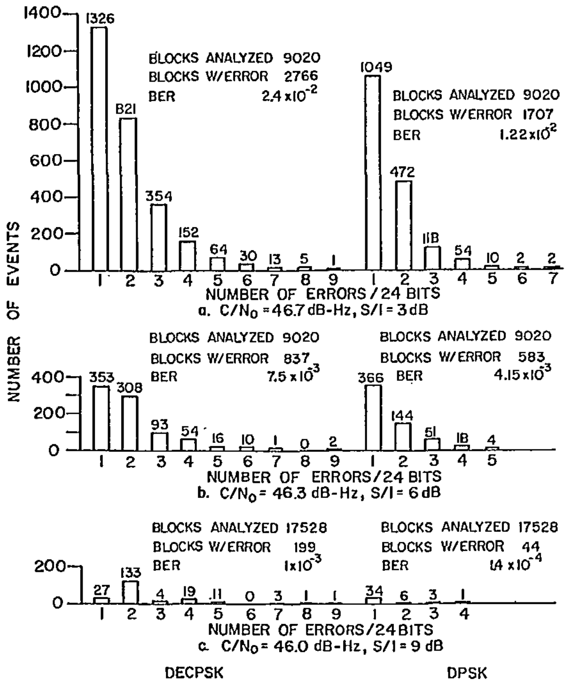


Figure 3. Block Error Histograms

It will be evident from the above discussion that multipath effects on digital PSK signals cannot be simply compensated for by increasing channel C/N_0. As an example, to improve the b.e.r. obtained for a S/I ratio of 8 dB with a doppler spread of 100 Hz on the multipath signal would require an increase in channel C/N_0 of about 10 dB. Clearly this is extremely expensive. Other means of obtaining the desired b.e.r. must therefore be sought, and the use of forward error-correction is one such way.

Based on the channel model described above, the code must improve a channel error rate typically in the 10^{-2} to 10^{-3} region to an output data b.e.r. of not worse than 10^{-5}. It must operate in the presence of random errors as well as burst errors, where the burst length is assumed to be no longer than seven bits at 1200 bps. Other parameters assumed are summarized below:

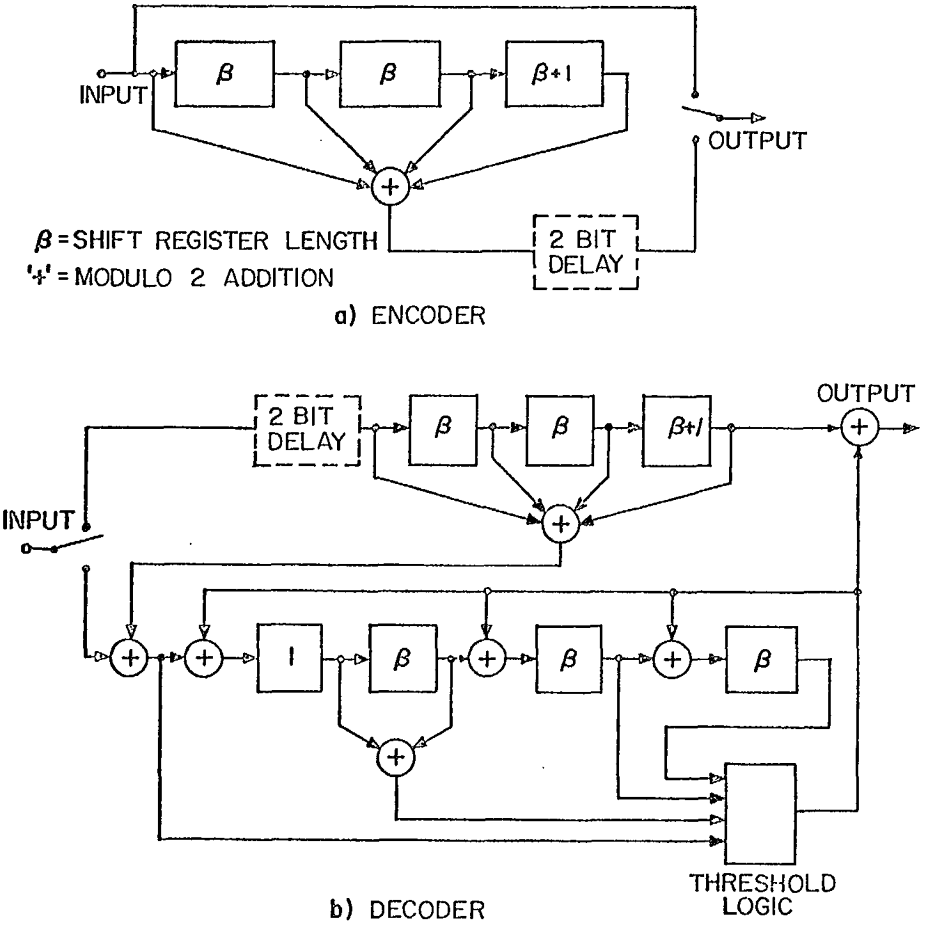
S/I ratio	10-13 dB
C/N_0	43/44 dBHz
Multipath specular reflection delay	5-25 microsec
Multipath spread	1 microsec
Multipath bandwidth	10-100 Hz
Differential doppler	20 Hz

3. CHOICE OF ERROR-CORRECTING CODE

Two basic approaches exist to the problem of combatting noise in a channel where both random and burst disturbances occur. The classical method is to use a code that is good for both random and burst errors, such as diffuse threshold-decodable convolutional codes; interleaved random-error-correcting block codes (with or without channel measurement decoding); character-error-correcting block codes, with or without interleaving; cyclic product block codes; cyclic block codes; and iterated burst and random error-correcting block codes. The other approach is to use an adaptive scheme in which separate decoding algorithms are employed for the same code, depending on whether burst or random errors are detected. Implementations of this scheme have been proposed for block codes and orthogonizable convolutional codes. Its overall decoded error probability is lower-bounded by the probability that the decoder fails to pick the correct decoding algorithm.

A number of coding schemes were considered for the aeronautical satellite application and the pros and cons are discussed in more detail elsewhere (Lyons, R., 1978). The conclusion was that the most suitable code for this application was a rate one-half diffuse threshold-decodable convolutional code. The theoretical basis of the code is described by Wilson, S.G., 1976 and Kohlenberg, A.K., 1968, and it has been shown to provide about 8 dB of coding gain at a 10^{-5} b.e.r. for 1200 bps information transmission over a simulated DPSK AEROSAT channel with an S/I of 10 dB and a channel fading bandwidth of 120 Hz (Wilson, S.G., 1976).

The coder/decoder, to be discussed in more detail below, is shown in block diagram form in Figure 4. The code rate is one-half, and it is systematic, i.e., the information appears explicitly in the symbol stream. Generation is similar to that for any convolutional code, except that the encoding constraint length is made long to diffuse the information over a span of output bits. At least β bits separate each tap in the shift register used to provide the parity or check bits. The greater β is, the more diffuse the code, and the greater its burst-correcting power. In the decoder, parity bits are obtained from the received information bits by the same process used in the encoder, and compared with the received parity bits to produce "syndrome" bits. A syndrome bit of "1" indicates that the parity bits differ and that an information or a parity bit error has occurred. Syndrome bits are stored in a register of length $3\beta+1$, with majority logic used to correct information bits. For this code, the decision to invert the information bit is made if three of the four syndrome bits examined are "1". In addition, once the decision to change an information bit is made, the correction is fed to the syndrome register so that the effects of that information bit originally in error are removed from the syndromes.



a) ENCODER

b) DECODER

Figure 4. Coder and Decoder for a Diffuse Threshold-Decodable Convolutional

It has been shown (Kohlenberg, A.K., 1968) that the decoder will correct all bursts up to length 2β (β information bits + β check bits) provided that there is an error-free guard space between bursts of length 6β+2 bits. The decoder will also correct any single or two-bit error pattern in the eleven bits used in the decoding process. The principal source of errors lies in three-out-of-eleven patterns, of which, however, about half are correctable. Of the 165 possible patterns only 85 produce an error, so the "first error" probability is about 85 p^3 (where p = error probability for a binary symmetric channel). The error probability assuming a white Gaussian noise channel with random errors is about 166 p^3 for small p (Kohlenberg, A.K., 1968). The closeness of these two values results from the negligible error propagation observed with these diffuse codes. One factor of importance here is that no more than two incorrect syndromes resulting from past decoding errors are used simultaneously. Thus, a decoding error at worst can act like two single errors in the decoder, which will not make another mistake unless a separate third error is present. In other words, the decoder will not continue to make errors if the channel has ceased to produce them.

4. IMPLEMENTATION OF CODER/DECODER

The error-correcting code described above has been implemented in two forms, one using hardwired integrated circuit logic, and the second using a Z-80 microprocessor. Besides encoding the data, the encoder produces a clock at the output bit rate. For message control, the encoder and decoder require a start and end-of-message signal which indicates the presence of valid data at the input. In the case of the encoder, this signal reflects the encoding delay and the presence of the overhead bits. The hardwired version will be described followed by the microprocessor version.

The encoder block diagram is shown in Figure 5. The parity generator incorporates a shift register of length $3\beta+1$, which is initialized to zero. Information bits are clocked into this register at 1200 bps and the bits at four taps are modulo-2 added to produce a parity bit. A parity bit is produced for each information bit clocked into the register and the two bit streams are then multiplexed to produce the 2400 bps output stream. Parity bits are transmitted until the last information bit has been clocked through the parity generator shift register. Zeroes are inserted into the information bit slots during the time it takes for the final bit to clock through the register. This produces an overhead of $3\beta+1$ bits for each message.

The decoder is shown in Figure 6. Information bits are passed through a parity generator identical to that in the encoder. The resulting parity bits are added modulo-2 with the corresponding received parity bits to produce the syndromes. These latter bits are clocked into a shift register and the syndromes at five taps examined by a majority logic circuit to decide whether or not an error has occurred. If so, the erroneous information bit is inverted together with the syndromes which identified it.

One possibility that has been envisaged is that the receiver may lose one bit in which case the information and parity bits demultiplexed by the decoder will be interchanged. If this occurs, a large number of errors will ordinarily be detected. To cope with this situation, a bit synchronizer circuit has been added which counts the errors in the information bit stream, as well as those in the parity stream on the assumption that they are actually information bits. Provided that the error rate of both streams does not exceed a threshold (4 errors in 50 bits), the synchronizer locks the decoder on the stream producing the fewest errors. As the synchronizer requires 50 bits to make a decision, the data is delayed by 50 bits (at the output rate). Resynchronization is achieved by switching between the input data and the input data delayed by one bit.

Because the codec was designedfor experimental purposes the value of β was made variable, selectable by on-board switches. β values of 4, 8, 12 and 16 were provided. An additional feature implemented in the codec was the option of selecting a two bit delay between parity and information bits before multiplexing in the encoder, and a corresponding delay in the decoder (shown dotted in Figure 4). If PSK with differential encoding is used to resolve the phase ambiguity at the receiver, with DECPSK demodulation two consecutive bit errors are produced for each isolated channel error. To avoid degradation of codec performance in this situation, the parity bit may be separated by two bits from its corresponding information bit by selecting this delay.

The codec was implemented using CMOS logic operating at 5V with TTL input/output buffering. It was laid out on two 230x100 mm PC boards, one containing the encoder and interface buffers, and the second the decoder. A total of 53 integrated circuits (ICs) were used, 16 in the encoder and 37 in the decoder. 17 of the decoder ICs were required to implement the resynchronization scheme. Power consumption was less than 1 watt at 5V.

5. MICROPROCESSOR IMPLEMENTATION

The codec was also implemented in software using a microprocessor integrated circuit to evaluate the advantages of using a microprocessor and specifically to determine: (1) how much hardware can be eliminated, (2) the amount of software required, (3) the timing constraints and maximum data rates possible, and (4) the interface requirements.

Among the 8 bit microprocessors the Zilog Z-80A microprocessor was chosen because it presently has the most powerful instruction set and the fastest machine cycle time. The Z-80A is similar in architecture to the Intel 8080 but has a much larger instruction set, more addressing modes, and more than twice as many internal registers. The clock frequency of the Z-80A is 4 MHz which gives a machine cycle time of 250 nsec.

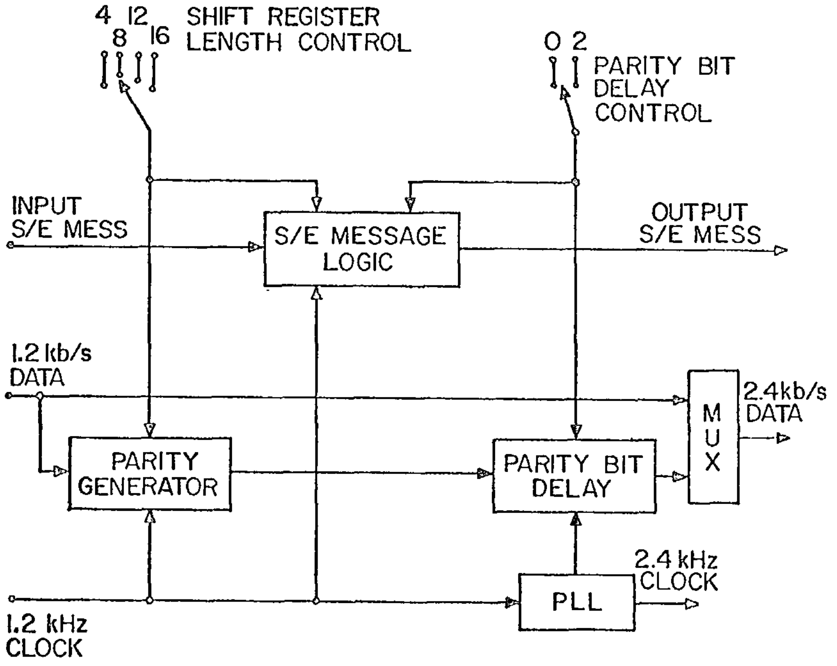


Figure 5. Encoder Block Diagram

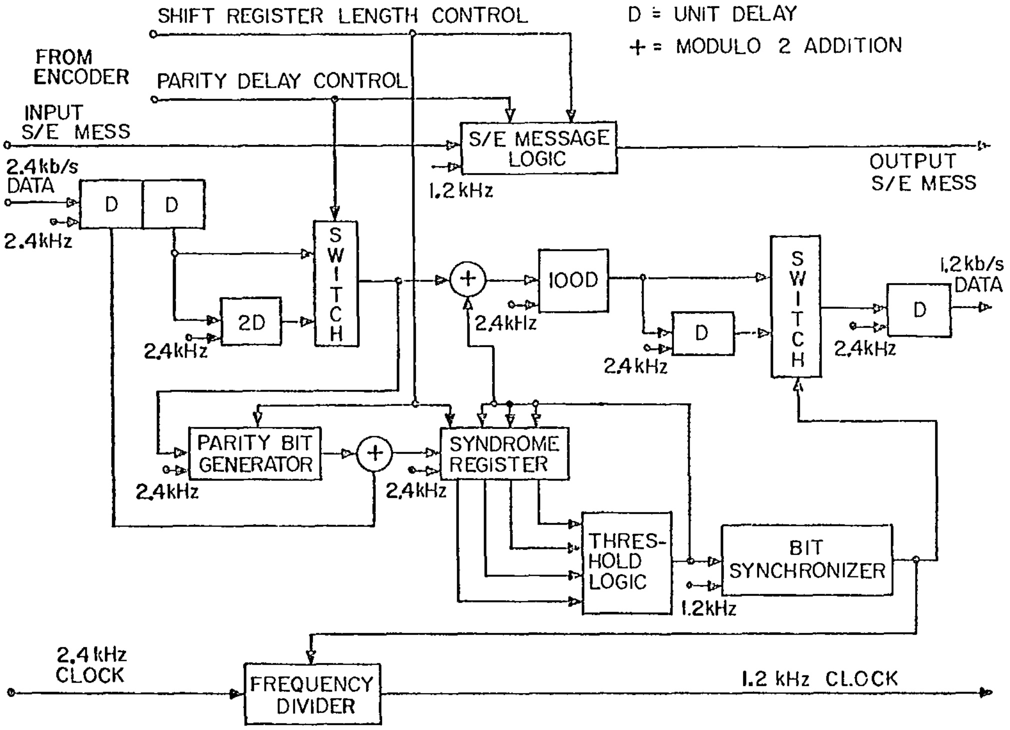


Figure 6. Decoder Block Diagram

The objective was to have one processor handle both the encoding and decoding functions at an input information rate of 1.2 kb/s. The initial approach was to make the interface hardware as simple as possible and to operate on a bit-by-bit basis. Using this approach, an interrupt flag is raised by the interface when a valid bit arrives. The microprocessor transfers the bit to a circular buffer in RAM created by software, performs the necessary modulo 2 operations with bits previously entered, and outputs the results. Since the amount of memory is not critical each bit is stored in a separate byte to avoid shifting and masking operations. Thus the encoder requires a buffer of $3\beta+1$ bytes or 49 when β is 16. In addition a pointer is required for each tap used to generate the parity bit, and these pointers must be updated with each new bit. The decoder requires buffers for the input data, the syndromes, and the delay required in the bit synchronizing scheme. The software also has to respond to the input start and end of message (S/E MESS) signal and generate an appropriately timed S/E MESS signal for the output.

The timing constraints for an information rate of 1.2 kb/s are as follows. The critical timing is governed by the interval between the arrival of two bits at the decoder, 0.416 msec, during which an encoder and decoder bit must be processed. At 4 MHz one machine cycle requires 250 nsec and a typical instruction requires 7 to 10 machine cycles. For example a register add requires 4 machine cycles, a read from or write to memory requires 7, and a simple jump instruction requires 10. Thus the encoder and decoder programs together must have less than 150 to 200 instructions. With the overhead required to set up the buffers and pointers the number of instructions was found to surpass this limit by a considerable amount so this approach was discarded.

A more promising approach at the expense of greater interface complexity is a byte-oriented scheme in which the interface accummulates 8 bits of the message before raising an interrupt flag. The microprocessor reads the 8 bits and stores them as a byte in RAM. Furthermore the algorithms for parity bit generation, error correction, and bit synchronization are carried out 8 bits at a time. The time available between successive interrupts of the decoder is now 3.33 msec which allows for 800 to 1200 instructions.

However the complexity of the interface is increased because a counter is required to determine when the 8th bit has arrived, the processor must be told whether the message had ended somewhere within the byte, and the data must be converted from serial to parallel format. Double buffering is required at the interface output to ensure that it will always be ready to accept the most recently processed byte and that there will always be a smooth flow of data.

When 8 bits have accumulated at the interface, they are read by the microprocessor and stored in RAM according to the structure shown in Figure 7. Notice that of the 8 bits that require processing, all of the $3\beta+2$, $2\beta+2$, and $\beta+2$ bits occupy a single byte. Furthermore, the subscript "1" bits can also be made to occupy the same byte through one masking operation.

Consequently, one can form the parity "word" simply as the exclusive-or of the bytes mentioned above. This requires only four EXOR operations or 4 μsec.

The data structures depicted in Figure 7 have one important feature. Since eight bits are effectively processed in parallel, and since the byte size is related to the chosen values of β in a simple manner a certain amount of masking is required to set up the subscript "1" bits. As a result, for $\beta = 8$ a storage location is required to hold i_1^1. This storage is however not required for $\beta > 8$ since i_1^1 will be the first bit in the final byte of the data structure.

Bench mark programs were written for the various values of β to ascertain the timing requirements. Elapsed time per eight bits was split into two parts:—

(a) the actual time for processing to form a parity byte;

(b) the additional time for data management.

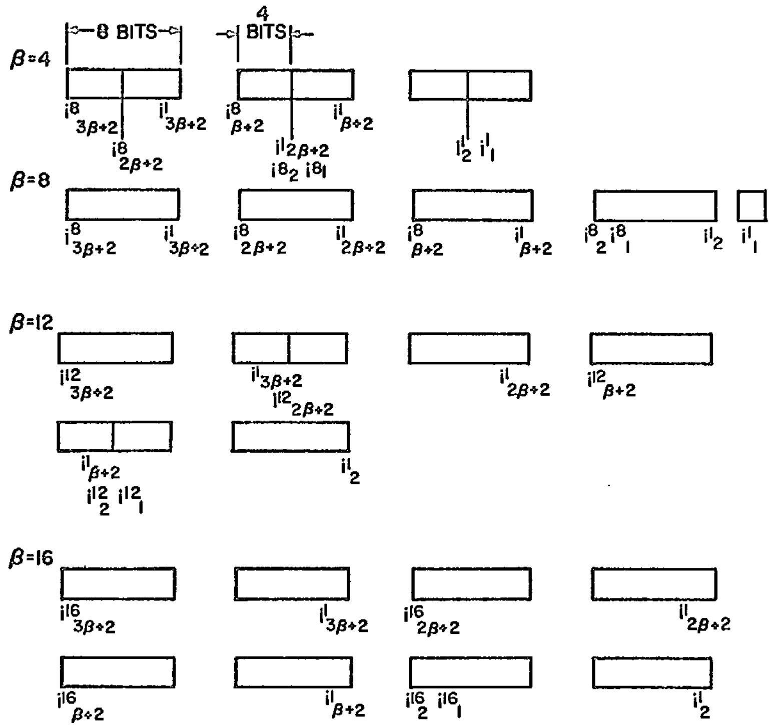


Figure 7. Encoder Data Structures

The quoted execution times in the following table are for a 0.25 μsec cycle time. All values are given per eight bits for an information bit rate of 1.2 KHz.

TABLE 1

Encoder Processing Requirements for 8 Information Bits

β	Management Program States	Processing Program States	Total States	Execution Time (μs)
4	498	273	771	193
8	560	54	614	154
12	684	261	945	238
16	808	82	890	224

These are the worst possible times which includes code to initialize the encoder at the end of a message in preparation for the subsequent message. For long messages (compared to eight bits), these states are only executed once such that the average execution time is reduced as shown:

TABLE 2

Encoder Processing Time Excluding Initialization Procedures

β	States Saved	Time Saved (μs)	Average Execution Time (μs)
4	158	40	153
8	258	65	89
12	358	90	148
16	458	115	108

Table 2 should be viewed with caution since the worst case timing must be utilized at least once per message.

Finally, it is interesting to note that less time is required for $\beta = 8$ than for $\beta = 4$ and similarly for $\beta = 16$ as opposed to $\beta = 12$. This results from the masking overhead required to form the $3\beta+2$, $2\beta+2$, etc. bytes when β is not an integral multiple of the processing word size of 8 bits.

In the decoder, structures similar to that of the encoder are employed in the data, syndrome and output registers. Again for $\beta<8$, there are two bits at the end of the structure in analogy to the single bit of the encoder. In addition, there is a single bit preceding the data register to save the most recent parity bit because a parity word can only be computed for up to and including the most recent information bit. A similar storage precedes the syndrome register. Its purpose is to cause the $S_{3\beta+1}$, $S_{2\beta+1}$, $S_{\beta+1}$ and S_1 syndromes to occupy single bytes rather than being resident in two bytes, thus reducing the masking overhead.

The software implementation of the decoder follows the hardware structure of Figure 6 very closely with the exception that the output register utilized in the bit synchronization scheme is 12 bytes or 96 bits long as opposed to 100 bits in the hardware version. Thus four information bit errors in 96 bits is the threshold for activation of the bit synchronization scheme provided less than four parity bit errors have occurred.

Benchmark programs were again written to ascertain the worst case timing based on a 0.25 μsec clock cycle. Results are given in Table 3.

TABLE 3

Decoder Processing Requirements for 8 Data Bits

	Execution Times (ms)		
β	Management Program	Processing Program	Total Execution
4	1.078	.39	1.468
16	1.078	.63	1.708

To support the encoding and decoding processes which are independent and asynchronous, the interface to the processor was designed with the following characteristics.

Encoder and Decoder Inputs:

(a) For each input the interface receives eight bits and causes an interrupt on the 8th.

(b) At the same time it generates a control byte which contains a "1" in each position in which the data byte contains a valid data bit. The control byte is used by the processor to determine where the last valid data bit of a message occurs.

Encoder Output:

(a) The interface accepts a data byte, a parity byte, and a control byte. It multiplexes the data and parity on a bit by bit basis for serial transmission, and uses the control byte to set and reset the S/E MESS line.

(b) It interrupts the processor when transmission of the current byte is complete. The processor then outputs another byte for transmission.

Decoder Output:

(a) A byte consisting of multiplexed information and parity bits and a control byte are supplied by the processor to the interface. The interface demultiplexes the information bits and uses the control byte to set and reset the S/E MESS line.

The software for the codec requires about 1.5 K bytes of storage and the complete circuit comprises less than 20 integrated circuits.

6. TEST RESULTS ON FEC CODEC

Three tests have so far been made to measure the performance of the hardwired logic version of the codec. These were performed in the laboratory and consisted of:

(a) tests of codec with PSK channel in presence of additive white Gaussian noise;

(b) tests of codec with simulated error bursts;

(c) tests of codec with a complete PSK channel using AEROSAT channel simulator at Transportation System Centre.

Tests over a real satellite link using a C130 aircraft and the ATS-6 satellite are also planned.

Tests (a) and (c) were conducted using the codec with a DECPSK channel unit manufactured by SED Systems Limited, Saskatoon, Canada. In test (a) thermal noise was added to the 70 MHz IF link between the modulator and demodulator, and the E_b/N_o was varied, at 2400 bps with and without the FEC codec in the circuit. A Hewlett Packard Model 1645A Data Error Analyzer was used as the data source. Results are shown in Figure 8, from which it can be seen that the channel unit, without coding, has an implementation loss of about 1 dB compared with the theoretical curve for DECPSK at 2400 bps. The effect of the codec should be to improve the DECPSK performance by 166 p^3. The coding gain for the theoretical DECPSK is also shown in Figure 8. It can be seen that the codec produces the anticipated gain, amounting to some 3.6 dB at 10^{-5} b.e.r..

The tests with simulated error bursts were performed without noise using a burst error generator operating on data. The generator produces a pseudo-random sequence of length 2^{36}-1 at a clock rate of 307.2 KHz. Whenever a selected bit pattern in the sequence occurs, a data bit is inverted, thus producing an isolated error. By varying the length of the selected pattern, the b.e.r. produced can be varied. Error bursts are generated by detecting a second pattern, 3 bits long, whenever the isolated error pattern is detected. In effect this produces an error burst instead of an isolated error one time in eight. The error burst length is under switch control and can be varied from 1 to 64 bits, all of which will be in error. Tests were run using this generator with several bursts lengths and a fixed value of $\beta = 8$ for the codec. Results are shown in Figure 9. It can be seen that useful coding gains are obtained for burst lengths up to 16, but with little or no gain for bursts of 20 bits. Tests were also made on the codec for several values of β from 8 to 16 with burst lengths of 16 and 20 bits. Results are shown in Figure 10, and confirm the ability of the codec to correct bursts up to 2β in length. It should be noted that bursts with every bit in error are not likely in actual channel conditions, and so the performance of the codec in a real environment should be better than measured in this test.

The tests using the AEROSAT channel simulator were performed at the Transportation Systems Centre, Cambridge, Massachusetts. The simulator was designed to represent as closely as possible the characteristics of the L-band aeronautical satellite communications channel and allows the various parameters of interest to be varied over ranges of values typically encountered. A full description of the simulator is given by Duncombe, C.B., 1975. The tests were conducted using the SED Limited channel unit noted above. Unfortunately, during transportation the SED channel unit suffered some internal damage which was manifested in an increase of

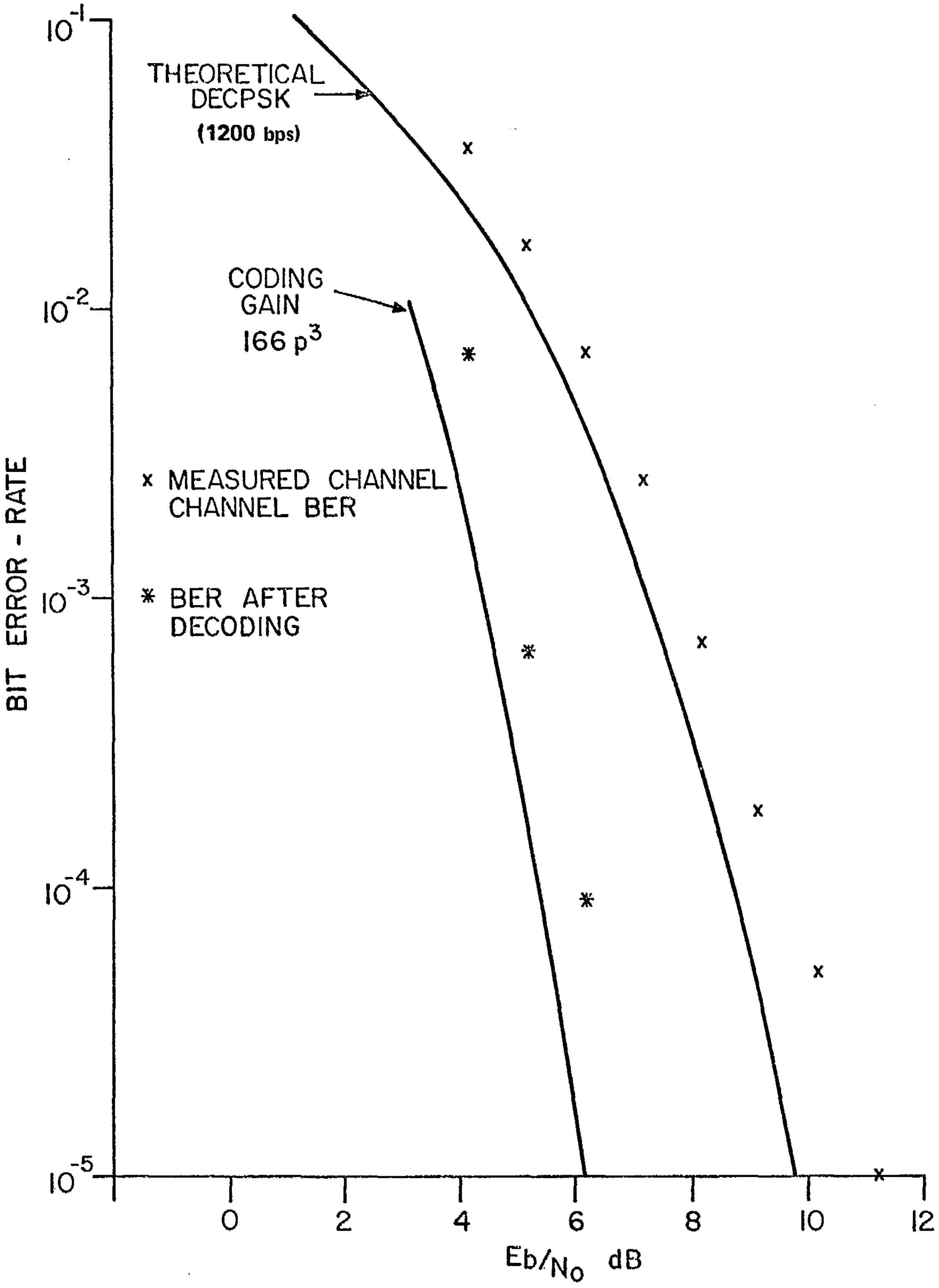


Figure 8. Bit-Error-Rate vs E_b/N_o for DECPSK Transmission at 2.4 kb/s Before and After Forward Error-Correction

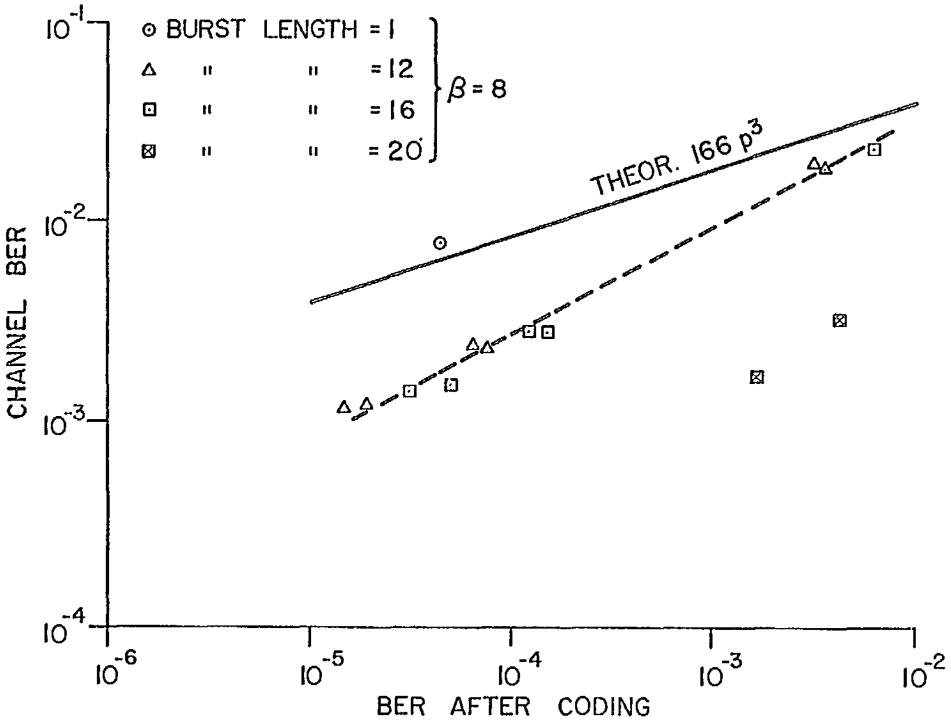


Figure 9. FEC Codec Performance Using Burst Error Generator: $\beta = 8$

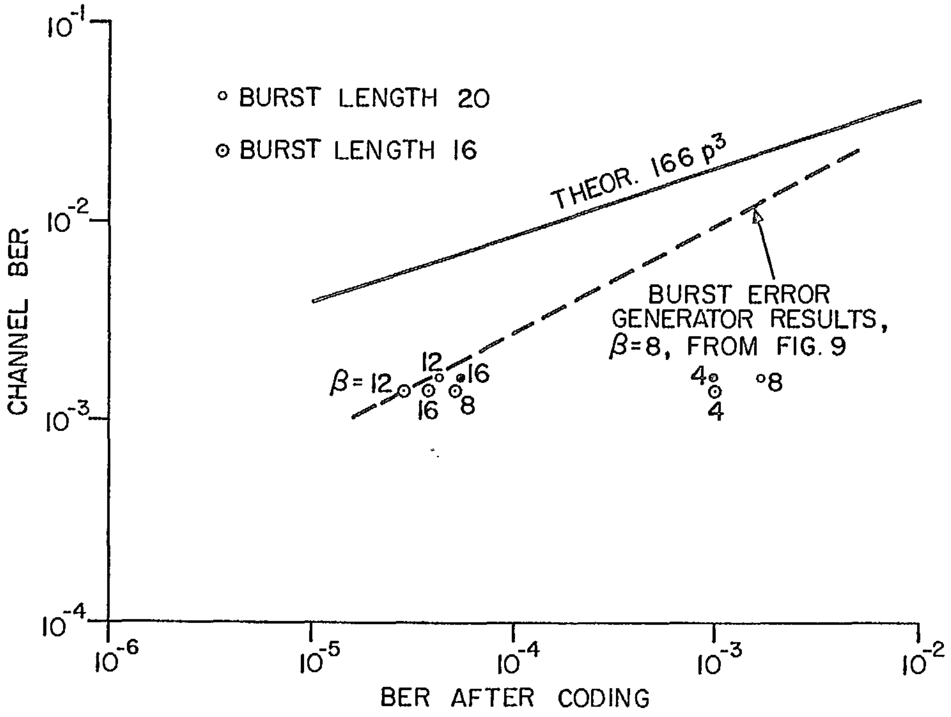


Figure 10. FEC Codec Performance Using Burst Error Generator: Variable β

implementation loss to more than 2 dB at 10^{-5} b.e.r., and as a result, the absolute performance of the codec and channel unit is somewhat poorer than anticipated. However, measurements indicated that the coding gains expected were obtained. The channel and decoder b.e.r. were measured for various combinations of C/N_0, simulator multipath bandwidth, and codec β. Values of C/N_0 between 42 and 52 dBHz, multipath bandwidths of 100, 50 and 10 Hz, and β values of 4, 8, 12 and 16 were used. Results are presented in Figure 11. It is interesting to note from Figure 11 that the coding gains obtained are much closer to the results for an AWGN channel than the results from the burst tests where every bit was in error. As the S/I decreases or the channel C/N_0 increases the coding gain tends towards the burst test values.

7. CONCLUSIONS

It has been shown that ocean surface multipath on an L-band aeronautical satellite communications channel will prevent the transmission of 1200 bps digital PSK signals with the required b.e.r. of 10^{-5} without a prohibitive increase in satellite power or avionics performance. Increases of channel quality of the order of 10 dB are required to compensate for multipath where signal to interference ratios lie in the typical region of 10-13 dB. These effects of multipath may be reduced by the use of forward error correction, using a rate 1/2 diffuse convolutional code. Such a code is simple to generate and decode, and will give the required b.e.r. at a C/N_0 of about 43 dBHz. The codec has been implemented in hardwired logic form and also in microprocessor form where the coding and decoding functions are performed in software. Tests on simulated Aeronautical satellite channels have demonstrated that the predicted coding gain was obtained. Use of the technique involves a message delay of approximately 100 information bits plus the transmission of an overhead of 25 information bits for each message. While for some applications these requirements might cause difficulties, for the types of messages expected on the AEROSAT system, this forward error-correction technique is proposed as a cost effective implementation.

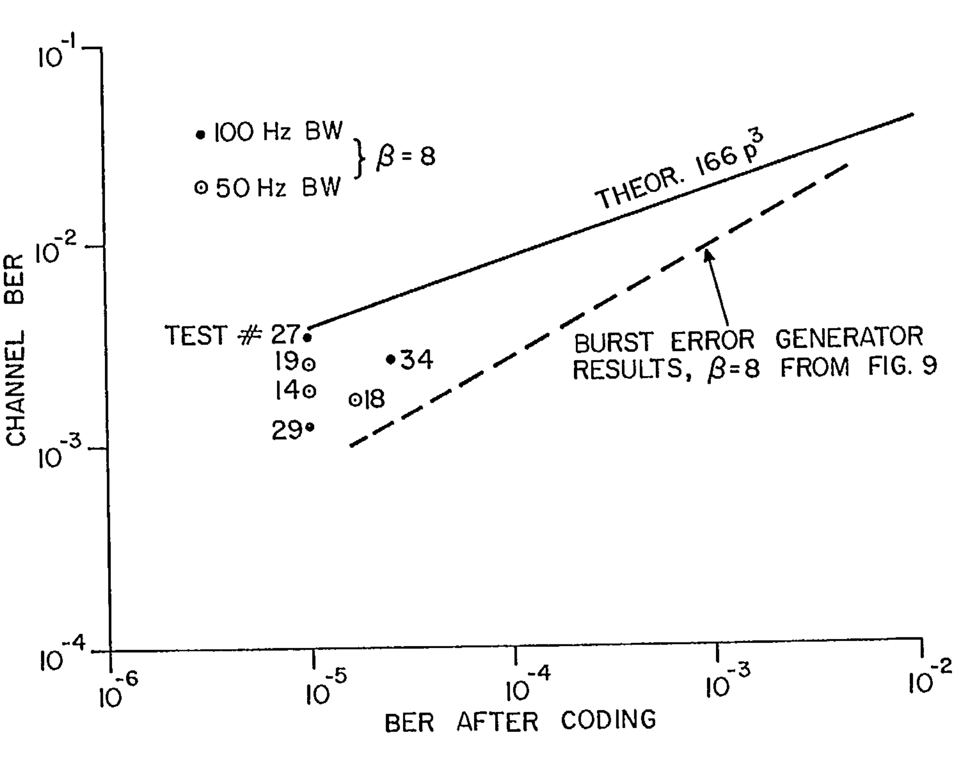


Figure 11. FEC Codec Performance With TSC Aerosat Channel Simulator

8. REFERENCES

Brown, D.L., 1976, *European Aeronautical Communications Experiments in L-Band with ATS-6, 1974–75,* **Volume 1,** European Space Agency Report NOX/0343/05/YG/1C, January 1976.

Chinnick, J.H., 1977, and Burtt, D., *Canadian Aeronautical Satellite Tests Using the ATS-6 Satellite, 1974-75,* CRC Report 1308, Communications Research Centre, Ottawa, Ontario, February 1978.

Duncombe, C.B., 1975, and Salwen, H., *Performance Evaluation of Data Modems for the Aeronautical Satellite Channel,* Transportation Systems Centre, Cambridge, Massachusetts, Report FAA-RD-75-150, September 1975.

Kohlenberg, A.K., 1968, and Forney, G.D., *Convolutional Coding for Channels with Memory,* IEEE Trans. Information Theory, September 1968, pp 618–626.

Lyons, R., 1978, and Beaudet, L., *Application of Forward Error-Correction Over Aeronautical Satellite Data Links,* CRC Report 1314, Communications Research Centre, Ottawa, Ontario, April 1978.

NASA, 1973, *Integrated Test Plan for ATS-F L-Band Experiment,* NASA Document TP-750-73-1.

Schroeder, E.H., 1976, and Thompson, A.D., Sutton, R.W., Wilson, S.G., and Kuo, C.J., *Air Traffic Control Experimentation and Evaluation with the NASA ATS-6 Satellite – Volume III: Summary of U.S. Aeronautical Technology Test Plan,* Boeing Commercial Airplane Company. Report FAA-RD-75-173, Volume III, September 1976.

Wilson, S.G., 1976, and Sutton, R.W., Schroeder, E.H., *Differential Phase Shift Keying Performance on L-Band Aeronautical Satellite Channels: Test Results and a Coding Evaluation,* IEEE Trans. Communications, March 1976, pp 374–380.

CRC DOCUMENT CONTROL DATA

1. ORIGINATOR: Department of Communications/Communications Research Centre

2. DOCUMENT NO: CRC Technical Note No. 698

3. DOCUMENT DATE: February 1979

4. DOCUMENT TITLE: Forward Error-Correction for the Aeronautical Satellite Communications Channel

5. AUTHOR(s): A. Sewards, L. Beaudet and H. Ahmed

6. KEYWORDS:
(1) Forward Error-Correction
(2) Aerosat
(3) Communications

7. SUBJECT CATEGORY (FIELD & GROUP: COSATI)

17 Navigation, Communications, Detection, and Countermeasures
17 02 Communications

8. ABSTRACT:

The paper identifies the effects on an L-band aeronautical satellite communications channel of ocean-surface multipath and shows that data messages transmitted over the channel exhibit unacceptably high error-rates under typical conditions. Channel characteristics and techniques for reducing the error-rate are discussed, leading to the choice of a diffuse threshold-decodable convolutional forward error-correcting code. Two implementations of coder/decoders for this code are described, one using standard IC logic, and the other a Z-80A microprocessor. Results of tests of the IC coder/decoder with noise, data error bursts, and a system simulator are quoted, which show that the codec performed as expected and is capable of reducing the error-rate to 10^{-5} or better.

9. CITATION: ______________________

Government of Canada
Gouvernement du Canada