where f_c=1487.5 Hz, and i is the coefficient number. The transformation is in fact the frequency translation property of the Fourier Transform applied to the real part of the function only. The resultant response was examined. To tailor the response to an acceptable shape, the number of coefficients, the low-pass bandwidth and the Kaiser window coefficient were manipulated. A combination of 23 taps, a design bandwidth of 900 Hz and a Kaiser parameter of 2.5 result in a filter which

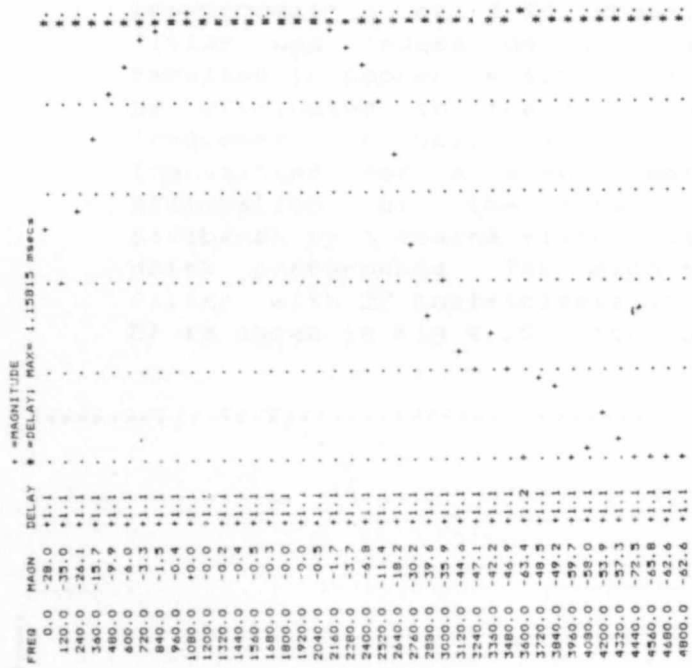


Fig 4.8 Decibel amplitude and group delay response of the modem channel filter. Note the flat group delay.

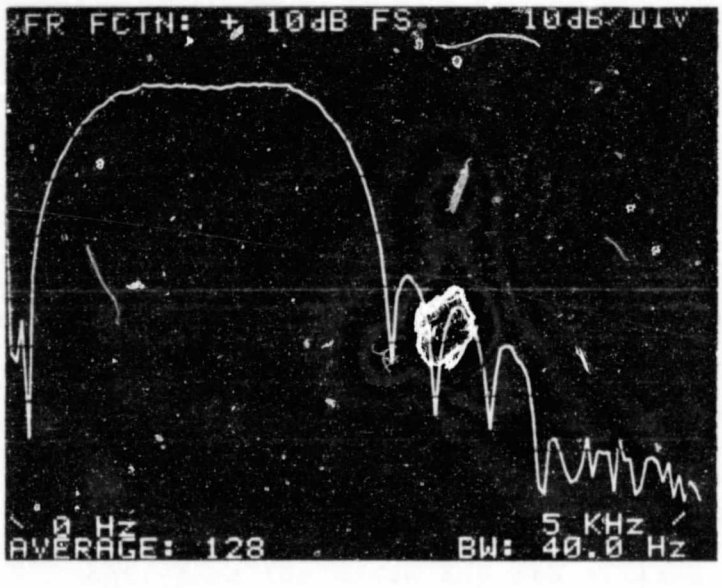
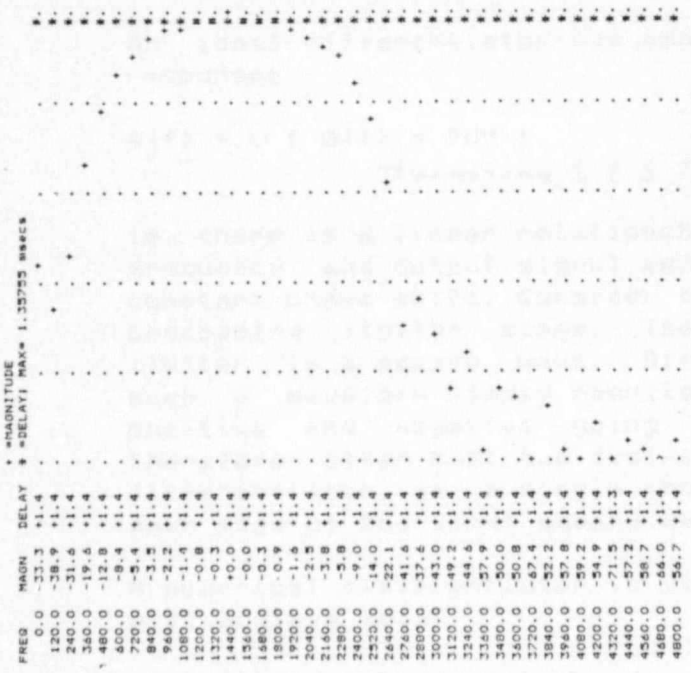


Fig 4.9 Photograph of amplitude response of modem channel filter. Compare this response with the spectrum of Fig 4.7.

satisfies the requirements.

The resulting calculated response is shown in Fig 4.8 and the achieved response in Fig 4.9. The filter coefficients are listed in Appendix E. It will be noted that the filter is not a positive lobe of a cosine function. The cosine response may simply be achieved by deriving coefficients from the impulse response of a cosine filter at the origin (by taking the Inverse Fourier Transform) and applying the transformation of 4.10 above. This type of filter was indeed designed and tested, but resulted in poorer rejection of noise. This may be attributed to the fact that at the mark frequency in use, a full cycle is not transmitted for a single mark. The greater attenuation of the first and subsequent sidebands by a cosine filter results in a poorer noise performance. The response of the cosine filter with 27 coefficients, (listed in Appendix E) is shown in Fig 4.10 for comparison.



FREQ	MAGN	DELAY
0.0	-33.3	+1.4
120.0	-38.9	+1.4
240.0	-31.6	+1.4
360.0	-19.6	+1.4
480.0	-12.8	+1.4
600.0	-8.4	+1.4
720.0	-5.4	+1.4
840.0	-3.5	+1.4
960.0	-2.2	+1.4
1080.0	-1.4	+1.4
1200.0	-0.8	+1.4
1320.0	-0.3	+1.4
1440.0	+0.0	+1.4
1560.0	+0.0	+1.4
1680.0	-0.3	+1.4
1800.0	-0.9	+1.4
1920.0	-1.6	+1.4
2040.0	-2.5	+1.4
2160.0	-3.8	+1.4
2280.0	-5.8	+1.4
2400.0	-9.0	+1.4
2520.0	-14.0	+1.4
2640.0	-22.1	+1.4
2760.0	-41.6	+1.4
2880.0	-37.6	+1.4
3000.0	-43.0	+1.4
3120.0	-49.2	+1.4
3240.0	-44.6	+1.4
3360.0	-57.9	+1.4
3480.0	-50.0	+1.4
3600.0	-50.8	+1.4
3720.0	-67.4	+1.4
3840.0	-52.2	+1.4
3960.0	-57.8	+1.4
4080.0	-59.2	+1.4
4200.0	-54.9	+1.4
4320.0	-71.5	+1.4
4440.0	-57.2	+1.4
4560.0	-58.7	+1.4
4680.0	-64.0	+1.4
4800.0	-56.7	+1.4

Fig 4.10 Decibel amplitude and group delay response of a cosine shaped channel filter. Compare the attenuation at the various sideband frequencies to Fig 4.8.

A further important characteristic of a transversal filter designed with a symmetrical impulse response, is that the phase response is linear, important in ensuring that all the signal distortion results from the channel and not from the modem and its filters. The effect of phase distortion on a digital signal is to upset the symmetry of a pulse. This may still make it possible to achieve 0% apparent bias distortion, but the receiver will be unable to distinguish between pulses in random pulse trains.

4.5.2 Frequency Discriminator

The frequency discriminator is composed of three elements viz Limiter, Differentiator and Envelope Detector, combined to operate as a frequency detector.

The limiter is a simple squarer operating directly on the output of the channel filter. All positive samples are converted to some arbitrary positive value, in this case 2^{-4}, or in the TMS 320's arithmetic, 0.25. Negative samples are handled in the same manner. The limiter represents the FM detector's main noise rejection capability. Any amplitude noise superimposed on the signal is effectively blanked by the limiting action. The effects of noise are seen, however, at the zero crossings. If additive or phase noise shifts the zero crossing point, then this noise is passed by the limiter. This indicates that total noise rejection is not possible and some noise is passed through to the differentiator.

An ideal differentiator has amplitude and phase responses

$$A(f) = \omega \; ; \; \emptyset(f) = 90^{o} \; ; \quad -f_{sampling} \leq f \leq f_{sampling} \qquad 4.11$$

ie there is a linear relationship between input frequency and output signal amplitude, with a constant phase shift. Consider the output of the preceeding limiter stage. The output of the limiter is a square wave. Differentiation of such a waveform simply results in a series of positive and negative going spikes. It is therefore clear that the desired output of any differentiator is a single short impulse for each edge of the input square wave.

A numerical differentiator in its simplest form will achieve this

$$d/dt\ f(t) = \frac{f(t_{i+1}) - f(t_i)}{t_{i+1} - t_i} \qquad 4.12$$

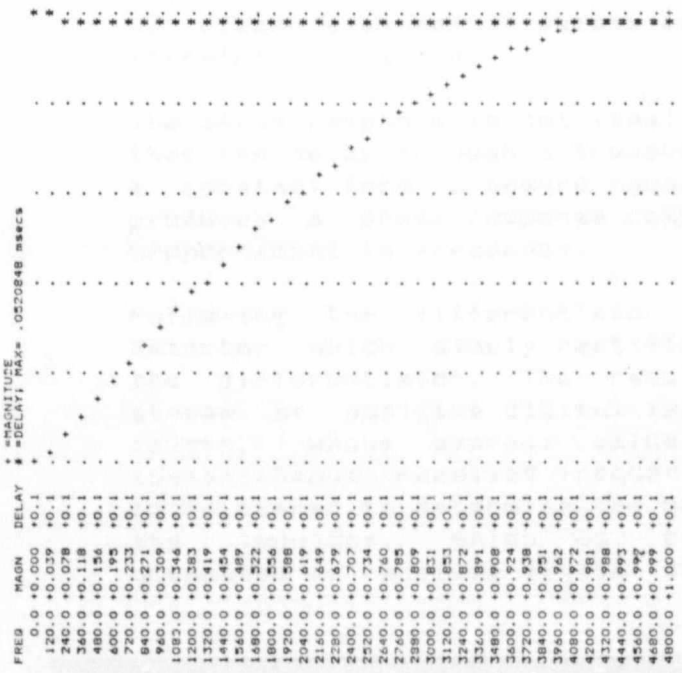
or

$$d/dt\ f(t) = \frac{f(t+\Delta t) - f(t)}{\Delta t} \qquad 4.13$$

This is very simple to implement using digital techniques as the time interval Δt is naturally present as the sampling interval. Implementation of this function does not achieve an ideal response, the response being slightly bowed, but the output for a square wave input is a series of digital impulses with the desired shape for application, after envelope detection, to a

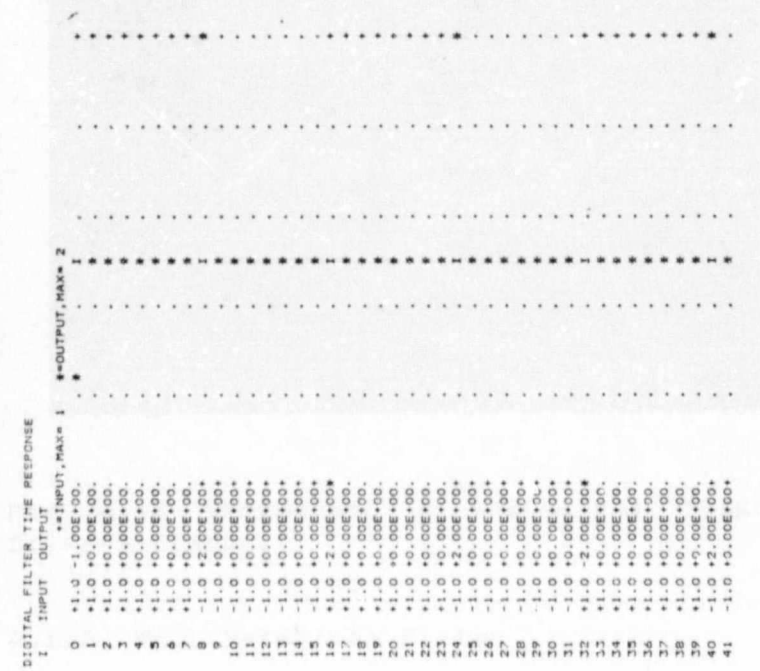
filter.

The figures 4.11, 4.12 and 4.13 illustrate the operation of the differentiator.



FREQ	MAGN	DELAY
0.0	+0.000	+0.1
120.0	+0.039	+0.1
240.0	+0.078	+0.1
360.0	+0.118	+0.1
480.0	+0.156	+0.1
600.0	+0.195	+0.1
720.0	+0.233	+0.1
840.0	+0.271	+0.1
960.0	+0.309	+0.1
1080.0	+0.346	+0.1
1200.0	+0.383	+0.1
1320.0	+0.419	+0.1
1440.0	+0.454	+0.1
1560.0	+0.489	+0.1
1680.0	+0.522	+0.1
1800.0	+0.556	+0.1
1920.0	+0.588	+0.1
2040.0	+0.619	+0.1
2160.0	+0.649	+0.1
2280.0	+0.679	+0.1
2400.0	+0.707	+0.1
2520.0	+0.734	+0.1
2640.0	+0.760	+0.1
2760.0	+0.785	+0.1
2880.0	+0.809	+0.1
3000.0	+0.831	+0.1
3120.0	+0.853	+0.1
3240.0	+0.872	+0.1
3360.0	+0.891	+0.1
3480.0	+0.908	+0.1
3600.0	+0.924	+0.1
3720.0	+0.938	+0.1
3840.0	+0.951	+0.1
3960.0	+0.962	+0.1
4080.0	+0.972	+0.1
4200.0	+0.981	+0.1
4320.0	+0.988	+0.1
4440.0	+0.993	+0.1
4560.0	+0.997	+0.1
4680.0	+0.999	+0.1
4800.0	+1.000	+0.1

Fig 4.11 Frequency response of differentiator implemented according to 4.13.



DIGITAL FILTER TIME RESPONSE

I	INPUT	OUTPUT
		*=INPUT, MAX= 1
0	+1.0	-1.00E+00
1	+1.0	+0.00E+00
2	+1.0	+0.00E+00
3	+1.0	+0.00E+00
4	+1.0	+0.00E+00
5	+1.0	+0.00E+00
6	+1.0	+0.00E+00
7	+1.0	+0.00E+00
8	-1.0	+2.00E+00
9	-1.0	+0.00E+00
10	-1.0	+0.00E+00
11	-1.0	+0.00E+00
12	-1.0	+0.00E+00
13	-1.0	+0.00E+00
14	-1.0	+0.00E+00
15	-1.0	+0.00E+00
16	+1.0	-2.00E+00
17	+1.0	+0.00E+00
18	+1.0	+0.00E+00
19	+1.0	+0.00E+00
20	+1.0	+0.00E+00
21	+1.0	+0.00E+00
22	+1.0	+0.00E+00
23	+1.0	+0.00E+00
24	-1.0	+2.00E+00
25	-1.0	+0.00E+00
26	-1.0	+0.00E+00
27	-1.0	+0.00E+00
28	-1.0	+0.00E+00
29	-1.0	+0.00E+00
30	-1.0	+0.00E+00
31	-1.0	+0.00E+00
32	+1.0	-2.00E+00
33	+1.0	+0.00E+00
34	+1.0	+0.00E+00
35	+1.0	+0.00E+00
36	+1.0	+0.00E+00
37	+1.0	+0.00E+00
38	+1.0	+0.00E+00
39	-1.0	+2.00E+00
40	-1.0	+0.00E+00
41		

Fig 4.12 Time response of Differentiator showing output impulses for a 600 Hz square wave input.

Other types of numerical differentiators were considered notably

$$d/dt\ f(t) = \frac{f(t+\Delta t)-f(t-\Delta t)}{2\Delta t} \qquad 4.14$$

but these proved unsatisfactory due to the fact that the length of the square pulse from the limiter may only be two samples long in some instances, and this was found not to produce a as clean a stream of pulses as 4.13 and were therefore discarded.

The phase response is not ideal, due to the fact that the delay through a transversal filter has a constant term to ensure causality, and this produces a phase response component which is proportional to frequency.

Following the differentiator is an envelope detector which simply rectifies the output of the differentiator. The result is simply a stream of positive digital impulses of equal length, whose average value represents the instantaneous received frequency. This signal is filtered to determine the moving average of the impulses, which may be detected and outputted as received digital data.

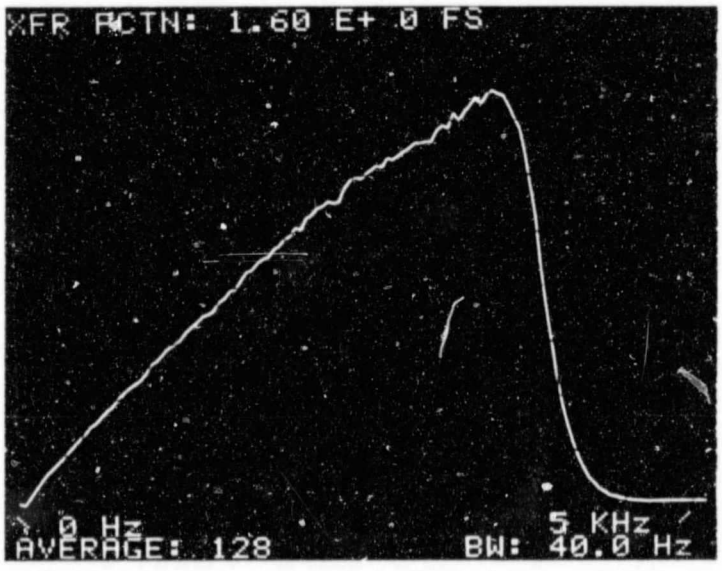


Fig 4.13 Photograph of the actual response of the Differentiator.

4.5.3 Post Detection Filter

The Post Detection Filter must average the rectified pulses from the differentiator to present a smoothed signal to the slicer. The filter impulse response should have zeros at the data sampling instants as set out in section 3.1.1 to avoid causing intersymbol interference. The decision threshold of the slicer is set to halfway between the levels of the mark and space outputs of the post detection filter. To ensure symmetry at the output of the slicer, the filter

output in response to a pulse of length one bit, should have a response width at the slicing level equal to one bit. Pulse area should also be preserved.

The filter should pass without significant attenuation the fundamental frequency of the highest data frequency viz 600 Hz, but it is not necessary nor desirable that the second harmonic be passed. Also very important is the requirement that the mark and space frequencies be properly rejected, as residual amounts of these tones can create ripples on the averaged signal which will confuse the slicer and result in erroneous data being outputted.

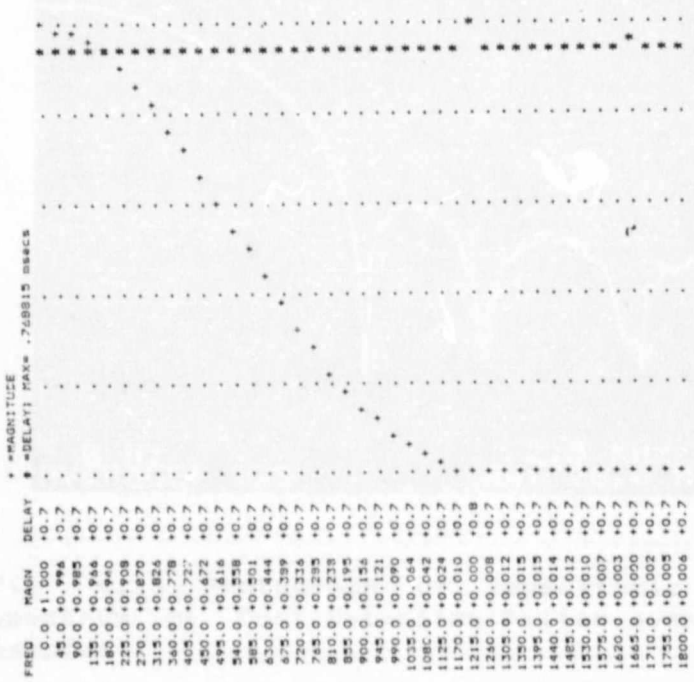
The frequencies which must be passed or rejected are then as follows. The 600 Hz data frequency should be passed unhindered if possible, but the frequencies at the second harmonic ie 1200 Hz and higher should have a high level of rejection. It may be thought strange that such low frequencies should need rejection, as these components should not be present after the rectifier, as frequency doubling takes place. However it is found that components exist at high level at lower frequencies than expected. This is attributed to the presence of sidebands at frequencies below the mark frequency which when doubled are still at frequencies of the order of 1200 Hz.

This means that Bennett and Davey's ideal filter viz the cosine shaped receive filter of section 3.1.2, is not ideal for this application, as the roll-off is not sufficient below 1200 Hz. The filter of 3.3 with its impulse response as 3.4 was chosen. Although this filter already has 6 dB of attenuation at 600 Hz, the 40 dB of rejection at frequencies above 1200 Hz is important. This filter also contributes little to intersymbol interference. It should be noted before continuing that Bennett and Davey do not recommend any specific post detection filter shape, but it seems logical to use a filter such as would be used on a baseband channel.

The filter is implemented as a 17 tap transversal filter which allows three bits to be partially covered simultaneously. A 17 tap filter covers the impulse response of 3.4 between the first negative and positive zeros (note that the 1st and 17th coefficients are actually zero). The coefficients are derived by considering the impulse response over the time period -T to +T where T is the data pulse length. This leads to 17 coefficients a_i where $i=-8,-7,\ldots 0,\ldots 7,8$ since there are 8 samples per data bit. The coefficients are evaluated at the time intervals

where T and i are as defined above. The coefficient at t=0 is taken as the reference and all the values are normalised against it. The coefficients derived thus are listed in Appendix E. Note that the coefficients at i=±8 are zero leading to the actual use of only 15 coefficients.

The flat delay characteristic of a symmetrical transversal filter is again of importance as it reduces "smearing" of the pulses - it should again be remembered that a linear phase



FREQ	MAGN	DELAY
0.0	+1.000	+0.7
45.0	+0.996	+0.7
90.0	+0.985	+0.7
135.0	+0.966	+0.7
180.0	+0.940	+0.7
225.0	+0.908	+0.7
270.0	+0.870	+0.7
315.0	+0.826	+0.7
360.0	+0.778	+0.7
405.0	+0.727	+0.7
450.0	+0.672	+0.7
495.0	+0.616	+0.7
540.0	+0.558	+0.7
585.0	+0.501	+0.7
630.0	+0.444	+0.7
675.0	+0.389	+0.7
720.0	+0.336	+0.7
765.0	+0.285	+0.7
810.0	+0.238	+0.7
855.0	+0.195	+0.7
900.0	+0.156	+0.7
945.0	+0.121	+0.7
990.0	+0.090	+0.7
1035.0	+0.064	+0.7
1080.0	+0.042	+0.7
1125.0	+0.024	+0.7
1170.0	+0.010	+0.7
1215.0	+0.000	+0.8
1260.0	+0.008	+0.7
1305.0	+0.012	+0.7
1350.0	+0.015	+0.7
1395.0	+0.015	+0.7
1440.0	+0.014	+0.7
1485.0	+0.012	+0.7
1530.0	+0.010	+0.7
1575.0	+0.007	+0.7
1620.0	+0.003	+0.7
1665.0	+0.000	+0.7
1710.0	+0.002	+0.7
1755.0	+0.005	+0.7
1800.0	+0.006	+0.7

Fig 4.14 The Post Detection Filter amplitude response. Note the attenuation at 600 Hz.

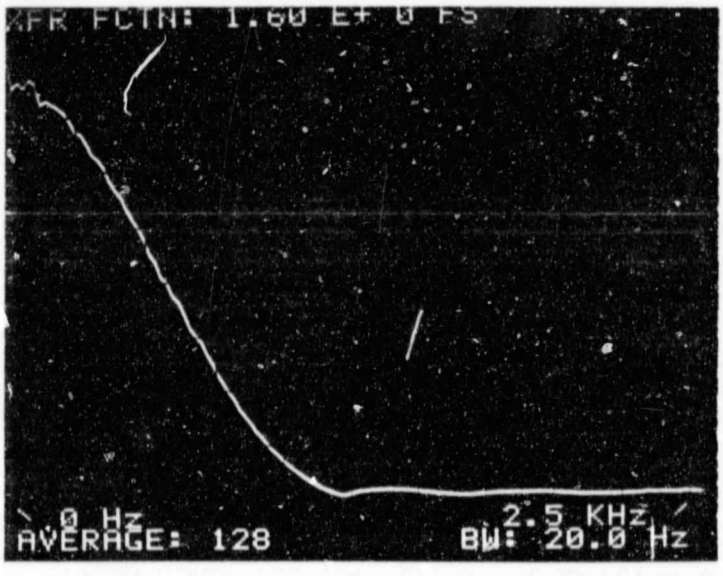


Fig 4.15 Photograph of linear spectrum of Post Detection Filter averaged over 128 samples.

characteristic is important in all aspects of data transmission and reception and that all the filters incorporated in the modem are of this type: data input filter, channel filter, differentiator, and post detection filter.

The predicted and actual responses of the post detection filter are shown in Figures 4.14 and 4.15 respectively. Fig 4.16 shows the decibel response of the filter.

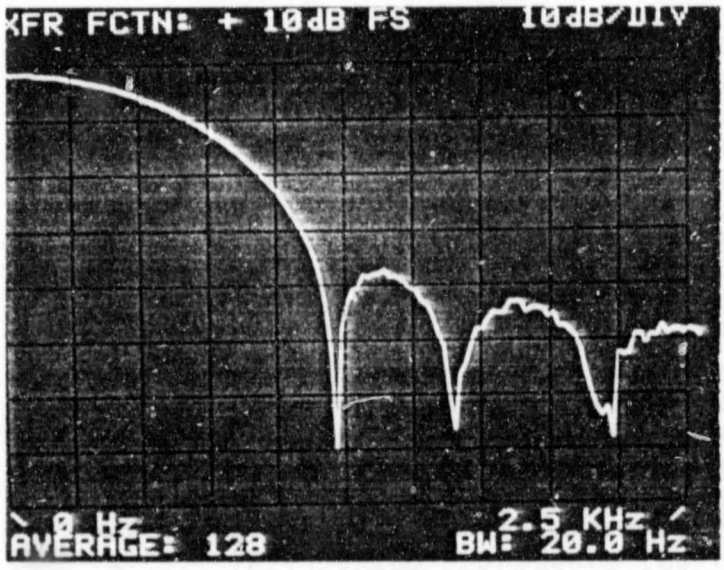


Fig 4.16 Photograph of logarithmic (ie decibel) spectrum of Post Detection Filter averaged over 128 samples.

4.5.4 Adaptive Equaliser and Slicer

The environment in which the modem is to be used dictates that all parts of the system should be robust and have a high reliability. The channels over which the system shall operate are fairly stable, with little short term drift. Longer term drifts may take place over a period of several hundred bits. An adaptive equaliser in this application need not exhibit rapid convergence. Simplicity, numerical robustness and a low computational requirement are however important, the latter especially as many functions have to be performed in any sampling period, with more to come possibly at a later date. The Least Mean Square type of adaptive equaliser is well suited to these requirements.

The equaliser is implemented with 5 weights. From the rule of thumb of Section 2.3, for 10% misadjustment, the settling time will be approximately 50 iterations. As most messages will be much longer than 50 bits, and with a

slowly varying channel, this length equaliser will be adequate.

Although the equaliser is arranged with 5 taps, it has 33 delays. All except coefficients 1,9,17,25 and 33 are zero. This has two results. One, the classical equaliser structure is satisfied where the sampling rate is the same as the data rate, and two, less program and data memory is used for implementation as only 5 coefficients need updating. Figure 4.17 shows a block diagram of the equaliser.

In the Least Mean Square type of equaliser, the square of the error signal is minimised by judicious adjustment of the equaliser coefficients. Before the equaliser may be applied to the output of the post detection filter however, the bias at the filter output must be removed, in accordance with the recommendations of section 3.2.3. With reference to Fig 4.1, the bias removal is performed at the Post Detection Filter output, before the Adaptive Equaliser. This is done by subtracting from the samples at the output of the filter the mean level of the signal which may simply be derived by sending 1:1 data and detecting the data at the filter output. The reference level resulting in 0% bias distortion is the correct value. This value represents the true mean of the output of the filter. It may also be calculated although simulation would be a simpler approach.

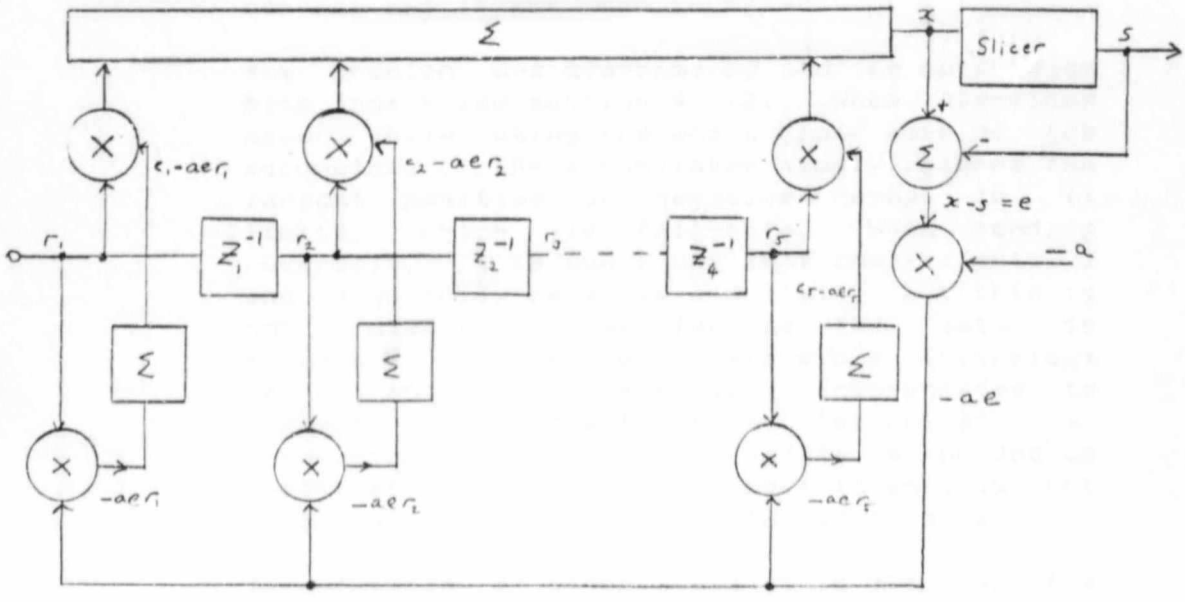


Fig 4.17 Block diagram of 5-tap Adaptive Equaliser. Note that each Z^{-1} represents a delay of 8 $T_{sampling}$

To permit rapid start-up and convergence on loop-back, the equaliser is preloaded with values similar to those to which it converges

with pseudo-random data. The convergence coefficient, a, is small (0.015) giving long convergence times, but as a result, a better adaptation to noise is obtained. Widrow et al show that small values of a produce little gradient noise, but exhibit lag, the opposite being true for larger values of a. The final value of a chosen was as a result of experiment, a compromise value being selected giving acceptable noise as well as delay equalising performance.

Initially it was found, notwithstanding the theoretical characteristics of the LMS equaliser, that the updating of the coefficients was "unstable", and that the system was sensitive to the value of convergence coefficient chosen. The problem was found to be in the TMS 320's arithmetic when in Overflow mode. It was found that the sign bit was being inverted, resulting in inversion of the output data which could have disastrous results if it occurred in use. The inversion appeared most frequently after a stream of reversals, 1:1, was sent. The central coefficient appeared to increase until the sign bit was inverted and an incorrect result was stored. The inversion of the sign bit will be further explained in section 4.6.3. If the reversals continued to be sent, the bit would again invert, but the transmission of normal data after inversion could not correct the situation. The equaliser can only operate on data perceived to be correct, although the data would have been correct had it not been inverted.

The problem was overcome by storing both sign bits (again see section 4.6.3). When overflows occur while using the whole upper word of the accumulator, the accumulator simply returns the largest positive or negative number ie it limits, which is fail-safe. When sending reversals, it is now found that coefficients 2 and 4 go fully negative and limit, but this is not catastrophic as far as the data is concerned, since no irreversible inversions occur. A further safeguard incorporated to prevent the coefficients "sticking" on unfavourable values, each update is rounded up positively, ensuring that coefficients do not remain at extremes, but are forced to move.

The success of these measures is seen by the fact that it is now impossible for the equaliser to "hang-up", no matter what combinations of data are sent. It is poor practice to attempt to train an adaptive equaliser on 1:1 data as there are large areas of the spectrum undefined, and this leads to the situation above where the equaliser cannot cope with the input. It is for this reason that an equaliser must be trained

with uncorrelated data samples as these will cover the whole spectrum to be equalised evenly, allowing the equaliser to adjust itself for any condition.

The slicer is included in the discussion of the equaliser as it plays an important role in its operation. The output of the equaliser is fed to the slicer which acts in the same manner as the limiter described in Section 4.5.2. This positive or negative value is subtracted from the signal at the slicer input, scaled by the factor a, and then used along with the sample values to update the coefficients.

4.6 Facilities

The modem has three "Facility" routines viz Carrier Detect which blocks the receiver when the input signal is too low; Request to Send which prevents output when off; and an Overflow detect function to indicate the presence of large numbers causing overflows in the signal processing routines.

4.6.1 Carrier Detect

The carrier detector is inserted between the channel filter and the limiter. The function is implemented simply by rectifying the channel filter output and then smoothing with a low-pass filter with very low cut-off frequency, in this case 10 Hz. The filter is a recursive filter with one pole, derived from an analog prototype and transformed to digital using the impulse invariant technique. This method first converts the s-plane pole to a z-plane pole, then calculates the coefficient value (in this instance the same as the pole constant).

Care has to be taken in designing filters with such narrow bandwidths relative to the sampling frequency as the coefficient value becomes very close to one and results in the filter having very high gain. The s-plane pole lies at s=-62.8318. When transformed to the z-plane we have a z-plane pole at

$$z = e^{-sT} \qquad 4.16$$

$$= 0.993477$$

ie $b_1 = 0.993477$ and the stage gain is

$$G = 1/(1-b_1) \qquad 4.17$$

$$= 153$$

It can be seen therefore that to avoid arithmetic overflows the input data should be scaled before applying it to the filter.

The output of the filter is an almost constant level which will not track rapid variations in channel level. The FSK signal itself contains inherent amplitude variations caused by non-ideal channel frequency response characteristics and it is not desirable that the level detector should track these variations. If the filter output does not reach a level equivalent to -16 dBm or greater, then the output is blocked, an output flag - the Receive Line Signal Detect - is reset, and the adaptive equaliser is disabled by jumping straight to the transmitter routine.

This feature is necessary to prevent the equaliser adapting to noise and other undesirable inputs.

4.6.2 Request To Send

This is a very simple test applied at the beginning of the transmitter routine. If the relevant input bit (bit 0) is set, the transmitter may operate, otherwise the output signal is set to zero and the transmitter routine is bypassed.

4.6.3 Overflow Detect

This function is required to detect when the result of an operation overflows the accumulator. When detected, a flag is set, lighting an indicating LED on the card. The TMS 320 Overflow mode is set and the BV "Branch on oVerflow" command is used to test the overflow flag. If set, the output flag is set warning the operator that something is amiss. This should only be necessary during setting-up of the system as all levels between the stages of the modem have been carefully matched to prevent overloading subsequent stages.

There is one drawback of the overflow mode which has come to light. If the contents of the accumulator increase past 7FFFFFFFh for positive numbers or decrease beyond 80000000h for negative numbers, then these become limiting values and the overflow flag is set. However in normal arithmetic operation overflows may occur without detection. This arises from the arithmetic employed by the TMS 320. Consider positive numbers only for simplicity, although the following is applicable to negative numbers as well. The device considers the 16-bit numbers as fixed point, 15-bit numbers, the 16th bit being the sign bit. These numbers are termed Q15 numbers as they have 15 bits of accuracy.

The problem arises in a quite common situation viz the multiplication of two Q15 numbers. Consider

```
   0.111 1111 1111 1111
x
   0.111 1111 1111 1111
=  00.11 1111 1111 1111 1111 1111 1111 1111
```

the result being a Q30 number as it has 30 bits of accuracy, and two sign bits. When the result is stored, the first sign bit is discarded, and the remaining sign bit and the following 15 bits

are stored, using the SACH instruction with a single left shift, as a correctly signed 16-bit number. It is recommended, however, in the application literature that the result be rounded by the addition of "0.5" ie the underlined bit is incremented. However in this example this will cause the result to become

01.00 0000 0000 0000 0011 1111 1111 1111

It will be noticed that the number which would be selected for saving has "rolled over" to become negative. This is not detected by the TMS 320 as an overflow, nor are 16-bit overflows in the low part of the accumulator. This example also applies when Q12 numbers are used such as with the MPYK instruction which operates with a 13-bit multiplier ie one sign and 12 data bits.

This complicates the design of signal processing structures as they are not "fail-safe" under overflow conditions as rollovers may still occur. The above example may appear to be a very specific case, but it may well be found that at least the first 15 data bits of the Q30 number may be "1" and that the rounding operation causes an unsuspected overflow. In the system under consideration, only the multiplications in the adaptive part of the equaliser are rounded up. Both sign bits are also utilised in the equaliser, making it impossible for rollover to occur while the Overflow mode is set.

5. HARDWARE OVERVIEW

5.1 TMS 320 Evaluation Module - EVM Board

The Evaluation Module serves as a very powerful tool in the development of signal processing systems employing the TMS 320.

The module operates as a low-level emulator, having an on-board TMS 320 processor, high speed RAM for program storage, an EPROM programming facility, and ports for communicating with an external terminal, microcomputer or mainframe on which the actual program development may be done. There is also a facility for storing programs to cassette tape.

Software included is comprehensive, allowing the EVM board to stand alone as a simple system. The main features of the software are a Screen Editor, an Assembler, Reverse Assembler and Patch Assembler. Facilities are included for execution with or without breakpoints, single stepping, examination and modification of internal (to the TMS 320) data RAM and program memory or internal registers and flags, and various other tools for conversion between signed or unsigned decimal and 16- or 32- bit binary numbers etc.

Block operations on program and data memory include filling with a value, or in the case of program memory, NOP's. Search facilities exist to search the various memories for specific entries for modification or examination in hexadecimal or signed or unsigned decimal. String instructions are available, making a powerful tool for single-stepping through a program. An instruction of the form

```
SS 1;;DDM 0 10;SD!
```

causes single-stepping to take place, displaying all the internal registers and flags, display the data memory between 0h and 10h in signed decimal form. the "!" at the end causes the instruction to be repeated every time the space bar is pressed. If a "%" is substituted. then execution is continuous, although still with the same displays, but stopping and starting alternately as the space bar is pressed. This feature permits detailed debugging of routines.

Finally the EVM allows operation with either its own 20 MHz clock driving the TMS 320, or an external clock situated on the target board. The EVM's processor has a short flying lead attached to it connecting it to the prototype, resulting in the emulator function. EVM based program memory or target memory may be selected, selection of the latter resulting in the loss of many memory manipulation features.

In use it is most advantageous to carry out program

development on an external microcomputer or other system, and then to download the program to the EVM board either in assembly language or as assembled and linked code. The former makes use of the EVM assembler while the latter simply writes straight to the program memory. The use of external hardware usually has the advantage that disk drives or other mass storage facilities are available, which are obviously more convenient to use than a tape based system. Software is available for the VAX 11/780 minicomputer, or for IBM PC compatibles for assembling and linking TMS 320 code. This will be further described in Chapter 6 on software development facilities.

5.2 Purpose Built Modem Board

5.2.1 Introduction

Initial development and familiarisation was done on the Texas Instruments Analog board, which interfaces with the EVM board. However, once the desired hardware facilities were identified, a board was designed carrying only the components and circuits necessary for the modem.

The board is on the standard telecontrol profile of approximately 150 mm square. It carries the Analog-to-Digital and Digital-to-Analog converters, anti-aliasing filters, input and output buffers and ports, clock generation circuitry and miscellaneous other circuits such as chip select logic, reset controller etc. The circuit diagram of the modem appears in Appendix C, as do state tables for the two multiplexers implemented using Programmable Array Logic (PAL) devices.

5.2.2 Clock Generation

The clock generator has to supply three unrelated frequencies to the system: the 20 MHz clock for the TMS 320; the 9600 Hz sampling clock; and 1.544 MHz for the 2912 switched capacitor PCM channel filter used as an anti-aliasing filter. The generator should use the external clock from the RTU at 307.2 kHz when available to derive the 9600 Hz sampling clock to keep the data and sampler locked, but should generate a substitute sampling frequency if the external clock should not be available - in older equipment for example.

The 20 MHz clock is simply derived from a simple crystal oscillator and is fed to the TMS 320. The 5 MHz output from the processor is used then to derive the further frequencies required.

The 5 MHz is divided by 3 to provide 1.67 MHz which drives the filter. This amounts to 8% higher than the required frequency and has the effect of shifting the various cut-off frequencies of the filter up by an equal amount. As a PCM channel filter is designed to attenuate signals beyond 3400 Hz, and a 8% increase only raises this to 3660 Hz, no problems arise as with a sampling frequency of 9600 Hz, frequencies up to 4800 Hz are acceptable.

The 5 MHz reference is also divided by 33 and then by 16 giving

$f_{sampling}$ = 9470 Hz

ie 1.36% low. As this signal will only be used upon failure of the accurate reference, or where the interconnection does not exist, this error is not considered a problem.

The clock generator circuit is arranged such that while an external clock is present, it is automatically given preference. The external clock, with its slightly higher frequency and consequent shorter period, resets the divide-by-16 counter before the internal clock causes the Q4 output to toggle. The external clock effectively forces this counter to synchronise to it, although the local input to the counter is still present, and is actually necessary to "re-toggle" the Q4 output prior to resetting by the external clock.

5.2.3 Input And Output Anti-Aliasing Filters

As mentioned above, the required Anti-Aliasing filters on the input to and output from the modem are implemented using a single switched capacitor device, an Intel 2912 PCM channel filter. This device provides an input filter with roll-off above 3400 Hz as well as hum suppression below 60 Hz. In addition, an output filter is provided complete with sinx/x correction, although this is only of limited value here due to the different sampling frequency in use. The output driver is capable of driving 600 ohm balanced line directly, but in this application, is simply used to drive an output isolating transformer.

As stated above, the clocking frequency is not exact, but this is of little consequence. The level produced by the input filter to the Analog-to-Digital converter is adjusted so that clipping results at this point for a signal of 0 dBm applied to the input. The digital input filter coefficients are adjusted so that maximum output levels are reached just after the input filter clips, in an attempt to limit the level of signal handled by the digital processor, preventing internal overflows with their attendant problems. A further advantage accrues from this viz that a technician may simply see the clipping using an oscilloscope, rather than having it hidden in the processor. This leads to a better understanding between maintenance personnel and the equipment being serviced. The output level may be set by a preset potentiometer to give -6 dBm as per specification.

$f_{sampling}$ = 9470 Hz

ie 1.36% low. As this signal will only be used upon failure of the accurate reference, or where the interconnection does not exist, this error is not considered a problem.

The clock generator circuit is arranged such that while an external clock is present, it is automatically given preference. The external clock, with its slightly higher frequency and consequent shorter period, resets the divide-by-16 counter before the internal clock causes the Q4 output to toggle. The external clock effectively forces this counter to synchronise to it, although the local input to the counter is still present, and is actually necessary to "re-toggle" the Q4 output prior to resetting by the external clock.

5.2.3 Input And Output Anti-Aliasing Filters

As mentioned above, the required Anti-Aliasing filters on the input to and output from the modem are implemented using a single switched capacitor device, an Intel 2912 PCM channel filter. This device provides an input filter with roll-off above 3400 Hz as well as hum suppression below 60 Hz. In addition, an output filter is provided complete with sinx/x correction, although this is only of limited value here due to the different sampling frequency in use. The output driver is capable of driving 600 ohm balanced line directly, but in this application, is simply used to drive an output isolating transformer.

As stated above, the clocking frequency is not exact, but this is of little consequence. The level produced by the input filter to the Analog-to-Digital converter is adjusted so that clipping results at this point for a signal of 0 dBm applied to the input. The digital input filter coefficients are adjusted so that maximum output levels are reached just after the input filter clips, in an attempt to limit the level of signal handled by the digital processor, preventing internal overflows with their attendant problems. A further advantage accrues from this viz that a technician may simply see the clipping using an oscilloscope, rather than having it hidden in the processor. This leads to a better understanding between maintenance personnel and the equipment being serviced. The output level may be set by a preset potentiometer to give -6 dBm as per specification.

5.2.4 Multiplexers/Selectors

Due to space limitations, PAL's were used to implement special functions and to combine functions. The special function implemented is a multiplexer selecting the routing of data. Data and transfer control signals such as Request to Send and Receive Line Signal Detect may be routed three ways selected by a switch. In the normal position the modem is connected to the edge connector data terminals to allow communciation between the remote and master. The second option is to connect the modem data terminals to a 25-pin "D" connector on the card front edge to allow connection of a data test set for testing the modem, and finally, the card edge terminals and the "D" connector may be linked to carry out tests from the RTU test set to exercise the RTU's functions directly.

A second PAL is used partly for the gates it offers, and partly to provide the I/O chip selector to select one of four ports: two for input and two for output. Spare gates on the chip are used in various applications in the circuit.

5.2.5 Other

The modem is provided with the following indications:

i) Request to Send

ii) Carrier Detect or more correctly Receive Line Signal Detect

iii) Overflow

The first two of these are simply LED's placed on the relevant signal lines, but the latter indication is driven from a monostable triggered by the overflow flag outputted as part of the data output word. This has to be done as overflows may only exist fleetingly, but are nevertheless important in understanding the operation of the circuit.

Data inputs to and outputs from the card are at V24 levels, and are driven by line-driver devices. This is to ensure proper interfacing to driving and driven equipment.

6. SOFTWARE DESIGN AND SIMULATION AIDS

6.1 Assemblers and Linkers

An Assembler and Linker for TMS 320 code is available on various machines. The operation and use does not differ markedly from system to system, the major differences being the actual manipulation of files by the computer. Software is available for the VAX 11/780 and for the IBM-PC and compatibles, both of which have been used in developing this project.

The assembler is straightforward to use as the TMS 320 instructions, although powerful for signal processing, are basically individually simple to make programming a relatively uncomplicated task, the complications mainly arising out of the design of the signal processing aspects of the task. The instruction set is small compared to conventional microprocessors with only 60 instructions, although many of these may be used in several ways with either direct or indirect addressing.

Since the TMS 320 can in the form used here ie the TMS 32010, can only address 4k words of external program memory, of which some will often be used for tables, and due to the fact that signal processing applications cannot, by their nature, require long programs, the task of allocating memory space for program storage, tables etc is simplified. Additionally the use of on-board RAM for data storage removes the requirement for memory management of external memory, with space for stack and other data areas not having to be allocated.

The TMS 32010 User's Guide [13] includes a section on Macro programs. However it is considered that their application is of limited use where time is constrained as the facilities offered may consume a lot of time to carry out. It is instructive to examine the methods used to carry out functions, and then extract only that which is needed.

The simplest program structure, and the one employed in this project, uses a single block of Relocatable code, started by "RORG". The final location of this code is determined by the Linker. The data memory is organised into a Data Segment bounded by the Assembler directives "DSEG" and "DEND'. When linking occurs, this area will also be allocated. The code must also have an identifier allocated in an "IDT" directive. The whole program is terminated in an "END" statement.

The Assembler is invoked by entering the name of the version available. A name of a program to be assembled will be requested, which will normally be expected to have the suffix ".ASM", whereafter the Assembler requests the names of a listing file with suffix ".LST" and a relocatable code file with suffix ".MPO". The Linker operates in an exactly similar fashion, requesting the names of the files to operate on. The

linker requires a control file "XYZ.CTL" which contains the following commands.

```
TASK M96                    refers to the program IDT
PROGRAM 0                   memory area for program code
DATA 0                      memory area for data storage
INCLUDE M96                 define the relocatable code
                            file to be linked
END                         the end of the linker
                            control file
```

The Linker, like the Assembler, operates on three files: the ".CTL" file which controls the linking operation, the ".MAP" file which gives the breakdown of how the linking was done, and a ".LOD" file which is the load module for downloading to the EVM or other emulator. It may be noticed on examination of the ".MAP" file that there is a warning that the program and data memory are located at the same address, but this is clearly of no consequence in the case of the TMS 320 as the data area is on board the processor.

The EVM Assembler may also be invoked directly to assemble code as it is downloaded. If this option is used, the assembly language file should start with ">" and end with "<" to inform the EVM of the beginning and end of the file being passed down. Some form of handshaking will be necessary to give the EVM time to carry out the assembly. Hardware handshaking, XON/XOFF or ACK/NAK are available. The assembled code may be uploaded simultaneously, either via the same port or via the second port. It is possible to use the EVM editor as the source for the Assembler only if the editor output is stored either to a port, the tape system, or to a RAM chip inserted into the EPROM programmer socket.

If the EVM Assembler is used on source code from another system, it is necessary that all code be absolute as there is no linker to perform the task of relocation. The EVM Assembler is a two-pass assembler, allowing use to be made of labels in the source. This also applies to the Patch Assembler.

6.2 Filter Design Software

The Electrical Engineering Department of the University of the Witwatersrand has a powerful filter design software package, allowing the development and analysis of many types of filters, analog and digital. Extracts have been taken from this and used with slight modifications on a microprocessor. Many additional routines have been written to enable a detailed analysis to be done on the performance of a digital modem. The package as used allows transversal and recursive digital filters to be designed using simple techniques and examine their frequency, phase and frequency responses. The diagrams representing various responses of filters which have been presented in this paper were developed using this software. In addition, modules were written to generate an FSK signal which could then be applied to the filter as in a modem, then limited, differentiated, rectified and then filtered again. The signal could finally be sliced to examine the effects of different slicing points. An adaptive filter routine was not written, but implemented instead in a simulation program which models the modem as a whole.

The routines used and their capabilities are listed below. The singularities and coefficients are held in memory, and there is an input and an output array for the time response operations.

BP	Generates Butterworth low-pass poles.
CWIN	Applies a Hamming, Cosine, Triangular or Blackman window to the coefficients.
DBGDOUT	Presents the logarithmic response (in Decibels) of a filter as calculated by GDZC, or GDSS.
FILTER	Passes a signal defined by INPUT or FSK through a filter defined by the coefficients under development.
FS	Permits entry or examination of a sampling frequency to be used by routines carrying out signal processing operations eg FILTER.
FSK	Provides a digital FSK signal such as would be generated by the modem under discussion, allowing the centre frequency and deviation to be specified. The digital input is provided by the INPUT routine, as modified by various manipulatory routines.
GDOUT	Same as DBGDOUT but simply displays normalised voltage response of a filter.
GDSS	Calculates the frequency response and group delay of a filter defined by s-plane singularities.

GDZC — Similar to GDSS but calculates the response from the z-plane coefficients.

IDLP — Generates "Ideal" Low-Pass Filter coefficients for a given cut-off frequency and for various numbers of coefficients. The coefficients are located on the sinx/x curve of a theoretically ideal square filter.

ILPBP — Converts transversal filter low-pass coefficients to band-pass coefficients, upon entry of the centre frequency.

INPUT — Generates digital waveforms: impulse, step, sine or cosine, square wave or pulse at a frequency which may be entered. Alternatively this routine will take the output of a filter as an input, allowing a sequence of operations to be carried out on the same original signal.

KWIN — Applies a Kaiser window to the coefficients, upon entry of the Kaiser coefficient.

LIMIT — Limits the input signal to between +1 and -1. All positive input values are returned as +1 and negative values as -1.

PZAB — Converts s- or z-plane singularities to coefficients.

RECT — Returns the absolute value of the input signal.

SETX — Defines the x-axis ie the frequency axis between various frequency limits and for differing numbers of points.

SLICE — Similar to LIMIT but permits the user to specify the slicing level.

TOUT — Displays the time response of the input and output to a filter or similar module eg RECT or LIMIT or FSK.

ZS — Converts from s-plane singularities to z-plane singularities or vice-versa.

Other routines are available for editing, displaying storing and recovering coefficients and singularities.

6.3 Modem Simulation

A simulation routine was written in BASIC to enable the operation of the modem to be examined. It includes a 511 bit pseudo-random sequence generator, as well as 1:1, 1:7 or 7:1 patterns. Facilities are included to insert noise, as well as a channel simulating filter. The routine is listed in Appendix B.

All the elements of the modem are simulated, including data filter, channel filter, limiter, differentiator, envelope detector, post detection filter, equaliser and slicer. The simulator outputs a list of the input to and output from the "modem", allowing comparison to be made between data in and data out. No software has been included to give a bit error count, or bias indication, as this would make the simulation very cumbersome and although execution time for a single loop is already long, such an addition would lengthen the time unacceptably.

The adaptive equaliser coefficients are displayed every 100 bits, allowing their convergence to be examined. It was the use of the simulator which indicated the need for the input to the equaliser to be symmetrical about zero, as there was a tendency after 2000 to 3000 bits for the bias to move completely in favour of "1's". When the symmetry was adjusted, the bias was found to remain correctly adjusted. The instability problems found with the equaliser coefficients was not discovered with the help of the simulator, though, as there was no problem with arithmetic overflows since floating-point arithmetic is employed.

The effectiveness of a simulator for digital signal processing applications is limited by the very slow speed of languages such as BASIC, especially when using an interpreter, although this has advantages in terms of allowing modifications to be quickly implemented and tested, without the problems of compilation and linking associated with compiled languages such as FORTRAN or Pascal. A simulator does, however, permit an impression to be obtained of how a system might perform. This advantage is strengthened by the fact that a high level language is much easier to manipulate than assembly language, even in the case of the TMS 320. It can be seen therefore that simulation may be advantageous when examining short term effects, but has limited value when examining long term tendencies. The loop time of the simulator is of the order of 1 second, a factor 9600 times greater than the actual system. It can simply be seen that to allow say 3000 bits to be transmitted at "1200 bd", will take nearly 7 hours. However effects appearing within the first 100 bits only take 15 minutes to take effect. Thus simulation has an application between these extremes.

7. PROTOTYPE MODEM RESULTS

7.1 Overview

The modem was constructed as has been described in the preceeding chapters. Much of the resulting hardware and software is a result of ongoing testing carried out during development. A specific example of hardware which underwent changes is the means of interrupting the program for the next sample. The interrupt was originally performed using the BIO pin, but to permit the future inclusion of comprehensive self-test features, the hardware was modified to use the hardware interrupt, although this has a penalty of at least 6 instruction cycles for the interrupt, branch and return. The use of interrupt allows greater future flexibility and independence of background routines performing other tasks. No self-test routines are included at this stage as they would confuse the development of the main part of the routine viz the modem, neither is use made of the data read from the DIP switches.

The main criterion for measuring the performance of a modem is the Bit Error Rate, BER, measured for differing noise levels and channel group delays. In the case of the example under consideration, an additional element has been added by the inclusion of an Adaptive Equaliser, as its effect on the performance should be analysed, to determine its worth. In the following sections, the performance of the system is examined for the different criteria. In each case the test circuit and instruments are included for reference.

Each BER measurement was made using 511-bit pseudo-random data sequences, with varying test lengths. In many cases this type of data does not represent the type of data with which the modem would be used. Many of the messages sent and received by the SCALD system are very short, of the order of 100 bits, and messages of this length have differing effects on equaliser adaptation, the effects of error bursts and the resulting effective BER. Notwithstanding these problems, the tests were performed as stated, as this is in line with international data-testing standards.

7.2 Adaptive Equaliser Performance

One of the more interesting aspects of the testing of the system was the question of the way the equaliser would perform, especially as it has to be stable and it should not be possible to make it latch up at extremes of its range. Tests were performed to test this. The test circuit used is shown in Fig 7.1. The data test-set used in all the performance tests is a Trend Data Tester, Model 105, capable of sending 1:1 reversals at various bit rates, as well as several data streams eg 63-bit pseudo-random, 511-bit PR and 2047-bit PR.

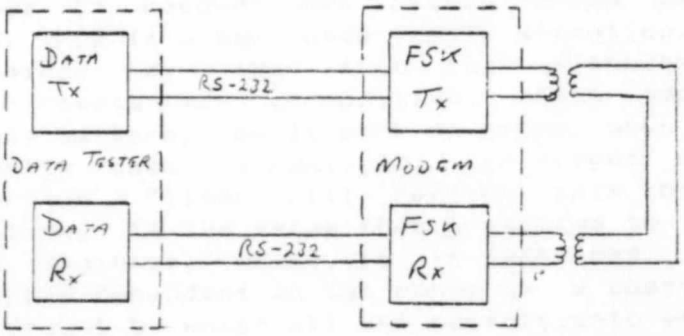


Fig 7.1 Circuit used to test the modem on loop-back

The equaliser's basic performance was tested with the modem simply looped as shown with no delay other than that contributed by the anti-aliasing filters, and no injected noise. The purpose of the tests was to get a feel for the way the equaliser adapted to various types of data, and whether varying lengths of random data had an effect on the resulting coefficients; 1:1 reversals and steady mark and space data was also used. The tests were commenced with the equaliser coefficients preset to 0,0,K,0,0 where K represents some constant, in this case 8192 was used, but any arbitrary constant could be inserted.

The results were enlightening as to the operation of the equaliser. Convergence coefficients ranging from 50 to 5000 were tested to examine their effects on rate of convergence, possible latch up and so on. For pseudo-random data with all three possible sequences viz 63-bit PR, 511-bit PR and 2047-bit PR, the equaliser converged on a similar set of values, representing the correct equalisation for the modem and its own filters. The values obtained thus are approximately those used in the final version of the software for initialisation, as listed in Table E.3. Although the coefficients as listed are symmetrical, the measured values were not quite so, but the difference in value between one side and the other is of the order of 1%, and this is therefore ignored for initialisation purposes. The number of bits sent to realise a stable set of coefficients varied depending on the convergence factor, a, as would be expected, varying approximately from 2000 for a=5000 to 10000 for a=50. Convergence is therefore reasonably rapid for most factors.

When 1:1 reversals are sent, an interesting phenomenon

occurs viz that the coefficients polarise quite drastically. The first, third and last values go very positive, the second and fourth limiting negatively. This is acceptable from the point of view of 1:1's, as since the equaliser coefficients each operate independently on successive bits, a high-low-high-low high sequence exactly matches the data being received.

From the point of view of convergence and adaptation to new patterns of data, the steady mark or space present the most problems. When a small convergence coefficient is used, say 200 or 400, the equaliser was found to latch up with all the coefficients positive, and be unable to recover when pseudo-random patterns were sent. This is a very undesirable situation. This may be explained by the fact that although all the coefficients may be positive, which emphasises the steady pattern, be it mark or space, when the opposite polarity data is received, the output of the Post Detection Filter will reflect this by going say negative. If the value then presented to the detector goes negative, which it in fact does, then it is entirely dependent on the convergence coefficient as to the amount by which all the coefficients are reduced. A small factor will take possibly a very long time to recover or may not recover at all. This latter appeared to be the case for small factors of the size mentioned, but did not apply to the values greater than approximately 1000.

It can be seen from the foregoing that the equaliser does indeed adapt to changing data patterns, so as to encourage the correct detection of the data. This is especially so in the case of 1:1 reversals.

7.3 The Effects of Noise

The circuit of Fig 7.2 was employed for the noise injection tests. The noise was Gaussian, and white over a band nominally 2.5 kHz wide, but in fact slightly wider, and centred at 1487.5 Hz, the modem centre frequency. The noise level was measured in a slot 100 Hz wide also centred at 1487.5 Hz. Due to the fact that the noise source could not deliver sufficient level an amplifier was used simply to raise the level of the noise to that required. The noise source was a Hewlett-Packard 3582A FFT Spectrum Analyser, with the noise generator set to Random, the data tester as above, the measuring set a Wandel & Goltermann PMG13, set to bridged measurement. The amplifier was set to x5 gain with an output impedance of 600 ohm.

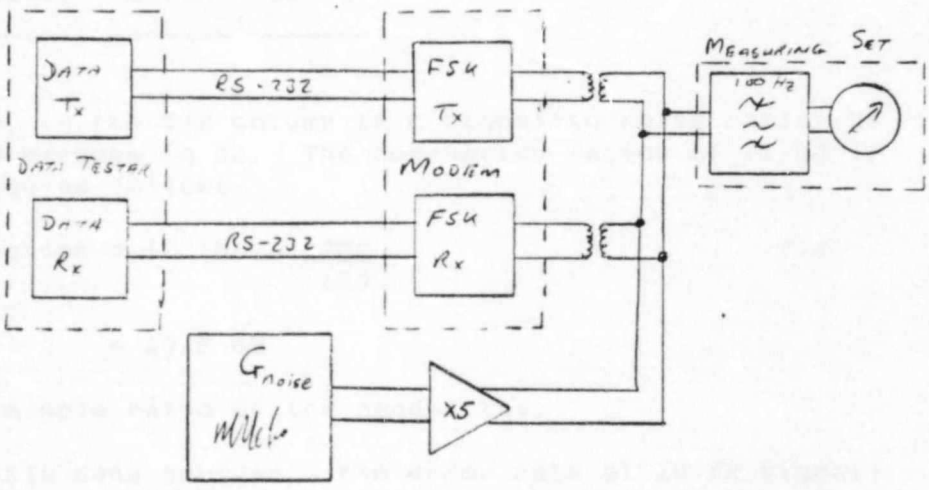


Fig 7.2 Connections for Noise Injection tests on modem

The FSK receive level was set to -6 dBm, with the noise source removed, but with the amplifier still connected and terminating the port. This was set at the transmitter on the 1:1 reversal spectrum, measuring wideband. The noise level was set with the FSK transmitter disconnected and the port terminated in a 600 ohm resistor. Six tests were done at different signal-to-noise levels for three convergence factors. In each case 10^6 bits were sent, and the bit errors measured. The results are presented in Table 7.1

It is interesting to note that there is little substantial difference between the error rates for the various convergence factors. Factors were only modified after the series of noise tests was completed. The results tend to bear out the fact that the use of a small convergence coefficient results in noise being better averaged, and hence resulting in an improved performance. The improved performance is more marked at high signal-to-noise ratios than at low ratios. This is because there is a limiting point beyond which it is not possible to go. Sunde's ideal modem achieves a theoretical error rate of 1% at a signal-to-noise ratio of 4 dB, measured in a bandwidth equal to the bit rate. If the 100 Hz band used here is replaced with a 1200 Hz band this would result in an 11 dB reduction in the

Table 7.1 Bit Errors Measured for Different Signal-to-Noise Ratios and Convergence Factors. 10^6 Bits Sent

S/N (dB) \ a	400	800	1000
36	1011	1263	2474
33	1895	2612	3704
30	2648	3108	4708
27	7460	9221	13596
24	25004	19903	21590
21	82502	83784	86945

value in the S/N column ie a signal-to-noise ratio of 21 dB becomes 10 dB. The correction factor of 11 dB is derived as follows

$$\text{Correction} = 10 \log_{10} \frac{1200}{100} \qquad 7.1$$

$$= 10.8 \text{ dB}$$

ie a simple ratio of the bandwidths.

In this case however, the error rate at 10 dB signal-to-noise is 8%. This difference may be ascribed to a number of factors. The first is that Sunde in his model considered idealised structures, especially for simplifying the analysis of the transmitter. Secondly, the digital sampling process contributes to "noise" in the form of jitter. Although it was stated earlier that the jitter would be reduced by synchronising the data and sampling clocks, this was not done in this case. There are two reasons for this viz that it is desirable to have an absolute reference for the modem, and secondly that the data tester would not reliably accept an external clock. When tests were first started, the data test set accepted an external clock; unfortunately, it later refused to accept external stimulus, and the tests were completed without synchronisation. All indications were however, that if the two clocks are synchronised, an improvement is possible in the error rate, although it was not possible to quantify it.

Fig 7.3 shows the error probability for a=1000 as a function of signal-to-noise ratio measured in the corrected bandwidth of 1200 Hz. Unlike the classical theory of error performance of a modem, the shape of the curve does not tend toward zero probability of error at high signal-to-noise ratios. It appears however as if it might be asymptotic to some value of error probability. This is reinforced by the finding