

https://dspace.org/

A software engineering approach to a digitally synthesised modem using a Texas Instruments TMS320 signal processor

Andrews, Donald Graham

2014-09-16
http://hdl.handle.net/10539/15550

Downloaded from a DSpace repository. https://dspace.org/

A SOFTWARE ENGINEERING APPROACH TO A DIGITALLY SYNTHESISED MODEM USING A TEXAS INSTRUMENTS TMS320 SIGNAL PROCESSOR

DONALD GRAHAM ANDREWS

A Dissertation Submitted to the Faculty of Engineering, University of the Witwatersrand, Johannesburg for the Degree of Master of Science in Engineering

Johannesburg 1985

ABSTRACT

This dissertation investigates the design and implementation of a modem, synthesised digitally using a Texas Instruments TMS 320 signal processing microprocessor.

The modem is an FSK 1200 bd type operating at non-standard frequencies for use on Escom's national SCALD system control network. It incorporates an adaptive equaliser giving it flexibility as to the type of channel it may work on without group delay compensation.

A detailed analysis is done of the signal processing software, examining the reasons for making various choices. Tests were performed on the final routine both by simulation and using the actual modem hardware and software.

Bit Error Rate tests are performed on the final product under conditions of varying noise levels and over a simulated channel with severe group delay over the band of interest. The results show that the principles are sound and that the project is feasible. The operation of the adaptive equaliser is examined carefully and guidelines are given for its use.

DECLARATION

I declare that this dissertation is my own, unaided work. It is being submitted for the degree of Master of Science in Engineering in the University of the Witwatersrand, Johannesburg. It has not been submitted before for any other degree or examination in any other University.

D. G. ANDREWS

(Name of candidate)

31st day of December, 19 85

ACKNOWLEDGEMENTS

The topic of this dissertation was suggested by Escom's Supervisory section. My appreciation is due to the members of the Special Projects team for their invaluable assistance in developing the hardware, carrying out the tests and generally making their time and resources available for the duration of this project. Also to Escom for their financial backing without which none of this would have been possible.

The author wishes to thank:

his supervisor, Professor Hanrahan for his encouragement during the project;
Mr Greg Allen of Escom for suggesting the topic;
and Mr Bruce Morris for his many practical comments and suggestions on the application of the modem.

The assistance of these people and many others was invaluable to the author.

CONTENTS

	ABSTRACT	ii
	DECLARATION	iii
	ACKNOWLEDGEMENTS	iv
	CONTENTS	v
1.	INTRODUCTION	1-1
1.1	Overview	1-1
1.2	Specification	1-2
1.3	Structure Of This Dissertation	1-3
2.	PREVIOUS WORK	2-1
2.1	Modems And Data Transmission	2-1
2.2	Digital Signal Processing	2-3
2.3	Adaptive Equalisation	2-4
3.	BACKGROUND THEORY	3-1
3.1	Data Transmission	3-1
3.1.1	Intersymbol Interference	3-1
3.1.2	Transmit And Receive Filters	3-3
3.2	Adaptive Equalisation	3-6
3.2.1	Basic Principles	3-6
3.2.2	Training	3-8
3.2.3	Other Considerations	3-9

4.	SOFTWARE DESIGN OF MODEM	4-1
4.1	The Modem Logical Structure	4-1
4.2	Choice Of A Sampling Frequency	4-2
4.2.1	Principles Determining Choice	4-2
4.2.2	Determining The Sampling Frequency	4-2
4.3	Software	4-4
4.3.1	Software Requirements And Limitations	4-4
4.3.2	Implemenation Of Requirements	4-4
4.4	Transmitter	4-7
4.5	Receiver	4-12
4.5.1	Receive Filter	4-12
4.5.2	Frequency Discriminator	4-15
4.5.3	Post Detection Filter	4-17
4.5.4	Adaptive Equaliser	4-20
4.6	Facilities	4-24
4.6.1	Carrier Detect	4-24
4.6.2	Request To Send	4-25
4.6.3	Overflow Detect	4-25
5.	HARDWARE OVERVIEW	5-1
5.1	TMS 320 Evaluation Module - EVM Board	5-1
5.2	Purpose Built Modem Board	5-3
5.2.1	Introduction	5-3
5.2.2	Clock Generation	5-3
5.2.3	Input And Output Anti-Aliasing Filter	5-4
5.2.4	Multiplexers/Selectors	5-5
5.2.5	Other	5-5

6.	SOFTWARE DESIGN AND SIMULATION AIDS	6-1
6.1	Assemblers And Linkers	6-1
6.2	Filter Design Software	6-3
6.3	Modem Simulation	6-5
7.	PROTOTYPE MODEM RESULTS	7-1
7.1	Overview	7-1
7.2	Adaptive Equaliser Performance	7-2
7.3	The Effects Of Noise	7-4
7.4	The Effect Of Group Delay	7-8
8.	CONCLUSION	8-1
8.1	Results	8-1
8.2	Suggestions For Further Work	8-2
8.3	Work Done	8-3
9.	REFERENCES	9-1
APPENDIX A	FILTER DESIGN EXAMPLE	A-1
APPENDIX B	THE MODEM SIMULATION ROUTINE	B-1
APPENDIX C	CIRCUIT DIAGRAMS AND PAL STATE TABLES	C-1
APPENDIX D	MODEM PROGRAM LISTING	D-1
APPENDIX E	FILTER COEFFICIENT LISTINGS	E-1
APPENDIX F	TMS 32010 SPECIFICATIONS AND INSTRUCTION SET	F-1

1. INTRODUCTION

1.1 Overview

The suggestion for this project came from Escom's Supervisory section which had a requirement for a 1200bd, 4-wire modem to operate on the countrywide SCALD (System Control And Load Despatch) control network. There is also a future requirement for a 200bd modem for the same purpose. The modems are required as part of an extension to and upgrading of the data-gathering and control network.

Escom's communications network is made up mainly of microwave radio and power line carrier (PLC) links. The SCALD system operates over these trunks radiating from the National Control Centre at Simmerpan in Germiston. Each major power- and sub-station is equipped with a Remote Terminal Unit (RTU) connected via its modem and a channel to the central control computer. The channels are configured as multidrop links with several RTU's being polled in turn by the master. This necessitates the use of a common set of modem frequencies. Due to the fact that a microwave and PLC link may be connected in tandem, they must of necessity accomodate the same data frequency band or else some form of frequency translation apparatus is required.

The telecontrol equipment as originally purchased used standard 1200bd frequencies viz Mark=1200Hz and Space=2200Hz. However, PLC equipment purchased at some later date had a speech band cut-off frequency of 2400Hz, which caused degraded data transmission at the 1200Hz/2200Hz frequencies. To overcome this, the frequencies were lowered by a factor 7/8 to 1050Hz/1925Hz making the modem non-standard.

The supervisory section decided to develop its own modem for use at these frequencies. It was also considered a desirable factor that a future 200bd modem might use the same basic components. The 200bd modem is standard and its frequencies are 3000Hz±120Hz. The decision to investigate a digital signal processor based modem follows from the advantages of flexibility offered by programmable systems, coupled to the fact that both the present 1200bd and future 200bd modem might be incorporated on the same board and be switch selectable. A further advantage of a signal processor based system is that it will interface more elegantly with future microprocessor based RTU's.

The initial approach wa he Intel 2920 as the heart of the system t scarded in favour of the TMS 320 which allow ially more flexibility in program design and in ation, with a further advantage of greater speed.

The paper which follows investigates the theory, design and implementation of TMS 320 based 1200bd modem.

1.2 Specification

Data Rate		1200 bd
Input Levels		as per CCITT recommendation V.28, Yellow Book, Vol VIII.1, Geneva, ITU 1980
Data	Mark (1)	<-3V
	Space (0)	>+3V
Control	Off	<-3V
	On	>+3V

Required Signals

Input	Data In	Pin 2 RS-232
	Request To Send	Pin 4 RS-232
Output	Data Out	Pin 3 RS-232
	Receive Line Signal Detect	Pin 8 RS-232
Common	Signal Ground	Pin 7 RS-232

Frequencies of Operation

Mark	1050 Hz
Space	1925 Hz

Levels

Transmit	-6 dBm / 600 Ω
Receive	-6 dBm / 600 Ω nominal (0 dBm0) 0 dBm / 600 Ω maximum before clipping
Squelch	-16 dBm / 600 Ω -10 dBm0

1.3 Structure Of This Dissertation

Initially previous work in the field of data transmission and signal processing, including adaptive equalisation is examined. The same aspects are then considered in greater detail.

The design of the modem and its individual components is then considered with an in depth look at each segment of the system and the role it plays in the overall transmission and reception of data. The hardware elements of the system are also considered briefly.

The systems used in developing this project are considered with a view to assisting new converts to the field of Digital Signal Processing using the TMS 320.

The results obtained are examined, showing that the concept is a valid one, but there remain areas where further work could be done to investigate whether or not improved performance might be obtained.

Further information is included on how to design filters, the simulation routine, and the TMS 320.

2. PREVIOUS WORK

2.1 Data Transmission and Modems

Baseband Data Transmission is the process of transmitting data between two points without the use of a bearer. This may be done over a continuous pair of conductors or some suitable combination of conductors and coupling equipment such as transformers and relays. The basic baseband transmission system is shown in Fig. 2.1

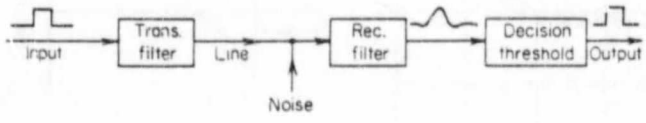


Fig 2.1 Binary Baseband System

The transmitting and receiving filters are chosen together to limit the effects of channel noise and intersymbol interference. Sunde [1] has shown that for a channel with an idealised low pass response and corresponding impulse response impulses may be transmitted at intervals equal to those between the zero-crossings of the impulse response without mutual interference between the received peaks. Sunde goes on to show that a realisable channel response with odd symmetry about the cut-off frequency will have an impulse response with zeros at the same time intervals as that of the impulse response of the ideal low-pass filter, but with much reduced tails which leads to a less critical timing requirement for the sampling instant. The various responses are shown in Fig 2.2 taken from [1].

Bennett and Davey [2] show that for a raised cosine roll-off ie

$$A(\omega) = \tfrac{1}{2}(1+\cos\tfrac{\pi}{2}\tfrac{\omega}{\omega_1}) \quad 0 \leq \omega \leq 2\omega_1 \qquad 2.1$$

the impulse response has little oscillation and is consequently insensitive to signalling rate.

Additionally they show that the impulse response has a width of one Nyquist interval at the half amplitude level giving full interval pulses if the signal is sliced at this level.

Stremler [3] also looks briefly at Nyquist Waveforms of which the raised cosine is one.

A modem is a device inserted in the "Line" of Fig 2.1, facilitating the transmission of data over circuits without galvanic interconnection. It incorporates the necessary modulation and demodulation schemes to enable accurate data transmission. The operation and design of

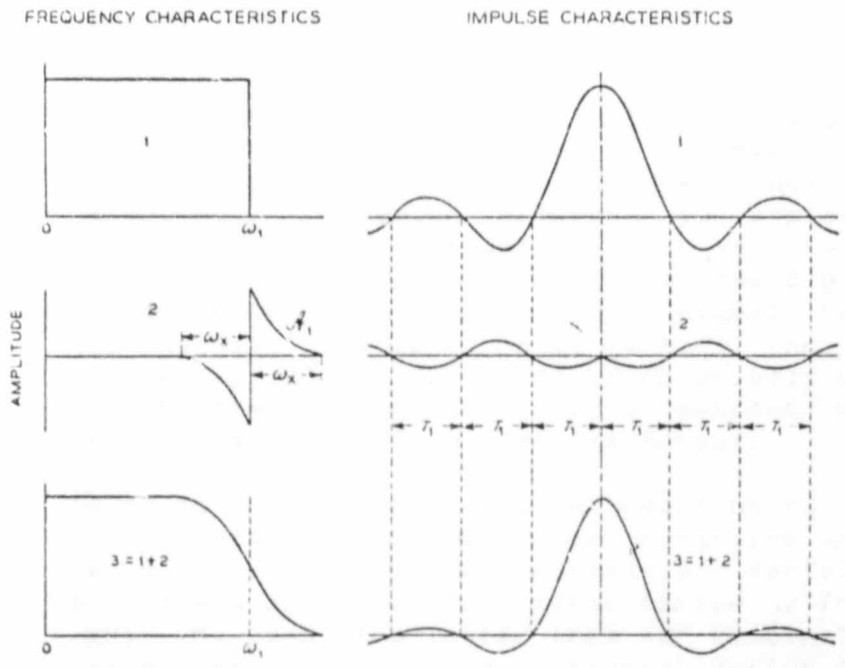


Fig 2.2 Idealised transmission characteristic with gradual cut-off, 3, obtained by superposition of characteristic with sharp cut-off, 1, and characteristic, 2, with odd symmetry about ω_1.

a modem is dealt with in detail in this thesis. Bennett and Davey examine the requirements for various modulation schemes, among them being Frequency Modulation (FM) or Frequency Shift Keying (FSK).

2.2 Digital Signal Processing

Digital signal processing is a well documented subject and it is not proposed to review available literature at length. A few works of direct relevance to the subject in hand will however be mentioned.

Digital processing of signals requires the signals to be sampled at a constant rate. Stremler [3] and Stanley [4] develop the concept of the "Nyquist Rate" for sampling. Stremler proceeds to explain the necessity for an anti-aliasing filter at the analog input to the sampler and the output of the Digital to Analog converter to limit the analog signal bandwidth to the Nyquist frequency which is half the Nyquist Rate. Stanley [4] and Intel [6] touch briefly upon the fact that the reconstitution of a sampled waveform results in a sinx/x roll-off at the output.

Stanley [4] presents a useful introduction to digital filtering. His text covers both recursive and non-recursive types and develops methods of deriving the former from standard analog filter tables as found in Williams [5] and the latter from a combination of Fourier series and windowing. Several design examples of each type are worked through.

Intel [6] also provide a straightforward introduction to practical digital processing techniques such as the synthesis of VCO's, Limiters, Auto Level Controls and so on. Stremler [3], Stanley [4] and Intel [6] all examine in varying detail the effect of quantisation on the overall noise figure of digital processors and the effect of the word length in bits on this figure.

2.3 Adaptive Equalisation

Equalisation is the means by which a signal, distorted in amplitude and phase by the channel response, is restored to its original transmitted form. The absence of channel noise is assumed. It is logical therefore that the equaliser response is the inverse of the channel. This is the broad definition of equalisation given by Clark [7]. Kumar and Moore [8] define adaptive equalisation as the use of transversal equalisers whose taps may be dynamically adjusted during reception of data so as to minimise intersymbol interference.

Clark defines a baseband channel as a bandpass channel eg radio multiplex together with a linear modulator and demodulator at the transmitting and receiving ends respectively. The equaliser is applied at the output of the demodulator, with the data detector following.

Clark reviews the concepts of linear and non-linear equalisation. The former has a transversal equaliser before the detector, the latter has a similar equaliser in feedback loop around the detector, resulting in the dynamic minimisation of intersymbol interference. Belfiore and Park [8] give a thorough analysis of non-linear or Decision-Feedback equalisation. This is a form of adaptive equalisation as it adapts the detector to cope with distortions of the data caused by limitations of the channel impulse response.

Clark suggests that true adaptive equalisation need only be applied where the channel varies appreciably during the length of a single message, assuming that a known training sequence is transmitted before the message. He proceeds to introduce the Least Mean Squares (LMS) adaptive transversal equaliser in linear and non-linear forms as described in the previous paragraph. The equaliser block diagram and basic equations are presented with a brief introductory operational description. The assumption that the equaliser has been "trained" is made. The adaptive equaliser as opposed to the decision-feedback equaliser minimises the mean squared errors in the data in the presence of both noise and intersymbol interference. The adaptive equaliser convergence coefficient, if small, allows integration over many received data samples, adjusting the equaliser tap values towards the optimum setting for noise rejection and signal acceptance.

Various papers [9,10,11] dwell on the fact that the LMS equaliser has slow convergence characteristics in comparison with other types such as Fast Quantised State methods [9], Adaptive Lattice Algorithms [10,11]. Kumar and Moore [9] in a tabular comparison between methods of equalisation indicate that although the LMS algorithm has slow convergence, it is simple, has a low computational requirement and is robust for limited precision arithmetic. It thus meets 3 of the 4

requirements as stated in their paper for an adaptive equaliser.

In the envisaged application as set out in 1.3, namely 1200 bd FSK over a "clean" channel, where the channel is not expected to vary appreciably for the duration of the message, the convergence rate is not important.

Widrow et al [12] examine the LMS adaptive filter in detail with specific emphasis on its learning characteristics. Several examples of this filter are presented with their results. Comparisons are drawn between theoretical and experimental results for different numbers of weights, iterations, average mean squared error at misadjustment. Design formulae are presented and both stationary and non-stationary characteristics are examined. Of importance are the conclusions drawn as to the recommended number of weights. A large number of weights may improve equaliser performance, but result in longer convergence times. A rule of thumb offered is that misadjustment equals the number of weights divided by the settling time.

It should be noted that in all the above papers, when reference is made to a sampling interval, this refers to the interval between data pulses and not some arbitrary sampling rate for other processing. . This has an important bearing on the design of a modem and equaliser using digital techniques.

3. BACKGROUND THEORY

3.1 Data Transmission

3.1.1 Intersymbol Interference

Conventional digital data consists of a stream of rectangular pulses transmitted over a continuous pair of wires. This "channel", if short and if the data is fairly low speed, does not significantly limit the required bandwidth. However when data speeds become higher and circuits longer, bandwidth limitations and how to overcome them become important.

Consider a rectangular pulse of length T, where

$$s(t)=1 \quad -T/2 \leq t \leq T/2 \qquad 3.1$$

$$=0 \quad \text{otherwise}$$

and s(t) has Fourier Transform S(ω) where

$$S(\omega)=\frac{T \sin \omega T/2}{\omega T/2} \qquad 3.2$$

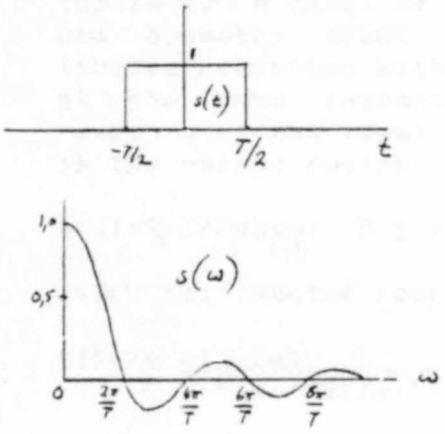


Fig 3.1 Rectangular pulse and its Fourier Transform

From Fig 3.1 it can be seen that although the bulk of the energy of the pulse appears at frequencies below $2\pi/T$ rad/s, a fair portion of the available energy lies above this frequency. To preserve the signal waveform, it can be shown that the channel amplitude characteristic must be flat and the phase response linear with frequency. Clearly this is impractical in real channels as the amplitude response is flat only to a certain extent and likewise the phase response is only linear over a portion of the band.

Accepting that the bandwidth is limited and will therefore distort any pulse we attempt to transmit, let us examine band limited systems.

Consider an ideal low- pass filter and its impulse response as in Fig 3.2. If impulses were to be transmitted over a channel with such a response, only if the impulses are separated by $T=n\pi/\omega_1$ will it be possible to distinguish peaks from one another.

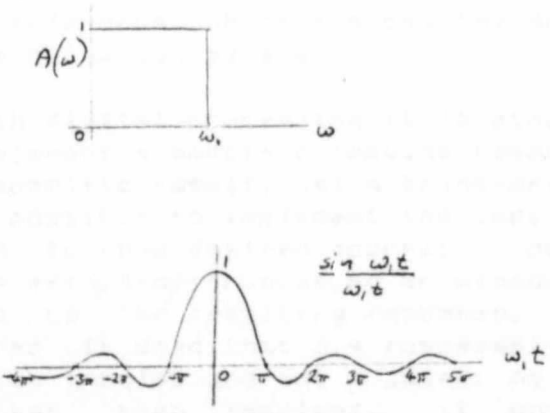


Fig 3.2 Ideal low-pass filter and its impulse response

If however, the impulse response has a large central peak and negligible tails, there is much greater freedom in the positioning of the impulses. A class of low pass functions having odd symmetry about the cut-off point, has impulse responses with reduced tails, but zeros at the same instants in time as the impulse response of the ideal filter. An example of this is the raised cosine

$$A(\omega)=\tfrac{1}{2}\left(1+\cos\frac{\pi\omega}{2\omega_1}\right) \quad 0 \leq \omega \leq 2\omega_1 \qquad 3.3$$

which has impulse response

$$g(t)=\frac{\omega_1}{\pi}\,\frac{\sin 2\omega_1 t}{2\,t\left(1-(2\omega_1 t/\pi)^2\right)} \qquad 3.4$$

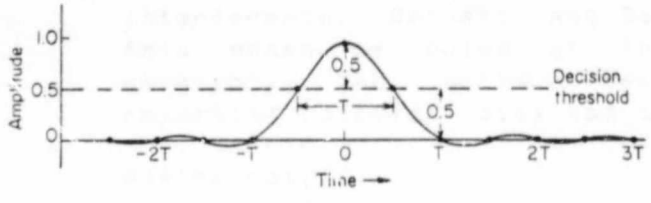


Fig 3.3 Pulse shape with raised cosine spectrum

As can be seen from Fig 3.3, this function has time zeros at the same points as the impulse response of the original ideal filter, but with much reduced oscillatory tails. This function will preserve pulse area and the width of the response at the half amplitude point is equal to the time between impulses. Another important characteristic of this response is that in

addition to the zeros at the instant of sampling, it has zeros midway between ie at the time at which the pulse is at the half amplitude slicing .evel.

From the foregoing, the importance of the effect the channel impulse response has on the received pulse shape and the resulting intersymbol interference which reduces the detectability of the pulse can be seen.

With digital processing it is straightforward to implement a specific impulse response to achieve a specific result, using transversal filters. It is possible to implement the impulse response of 3.4 to some desired accuracy, bearing in mind the effect of truncating or windowing a function has on the resulting response. It should be borne in mind that 3.4 represents the response to an impulse and not a pulse. As data is pulsed rather than impulsed, it would be more appropriate to find some other impulse response which has 3.4 as its pulse response, thus retaining the stated advantages of the pulse shape given by 3.4. This may be done either by numerically deconvolving 3.4 for a pulse of length T, or by multiplying 3.3 by x/sinx and taking the Inverse Fourier Transform.

3.1.2 Transmit and Receive Filters

The noise performance of a baseband data transmission system is examined by Bennett and Davey [3] for the purpose of determining the optimum response of the transmit and receive filters. A pair of optimum filters are derived for a raised cosine pulse spectrum at the detector and gaussian white noise.

As seen in 3.1.1 above, the raised cosine pulse spectrum is chosen to minimise intersymbol interference. Bennett and Davey show that for this shape of pulse at the input to the detector, the optimum receive filter for rejecting Gaussian noise has the following form

$$A(f) = K \cos \frac{\pi f}{2 f_s} \qquad 3.5$$

where f_s is equal to the data rate.

This filter will not return a raised cosine pulse without prior shaping of the transmitted pulse. We have:

$G_o(f)$ = Fourier Transform of pulse generated by the transmitter

$X(f)$ = Required response of transmit filter

$Y(f)$ = Response of receive filter

$$= \cos\frac{\pi f}{2f_s} \qquad 3.7$$

$S(f)$ = Raised cosine spectrum of pulse at detector input

$$= \cos^2\frac{\pi f}{2f_s} \qquad 3.6$$

For a full length (T) rectangular pulse of unit amplitude

$$S_o(f) = \int_{-T/2}^{T/2} \cos 2\pi f t \, dt$$

$$= \frac{\sin \pi f T}{\pi f}$$

$$= \frac{\sin \pi f/f_s}{\pi f} \qquad 3.8$$

Note that constants have been ignored in the interests of clarity.

We can see that $S(f) = Y(f).X(f).S_o(f)$ 3.10

$$\therefore X(f) = \frac{S(f)}{X(f)S_o(f)} \qquad 3.11$$

$$= \frac{f}{\sin \frac{\pi f}{2f_s}} \qquad 3.12$$

The transmit and receive filters have now been defined with the dual purpose of reducing sensitivity to noise and intersymbol interference.

Bennett and Davey carry out a similar analysis for FSK transmission. This is substantially more involved than the baseband transmission example as the Gaussian noise of the channel takes on some other distribution when passed through the non-linear FM detector. It is also shown that the effect of the noise is dependent on the history of the pulse sequence to some extent.

The results derived are applied to Sunde's [1] FSK system model in an attempt to define a channel filter shape. The shape arrived at is a positive lobe of a cosine centred about f_c, the channel centre frequency and reaching zero at $f_c \pm f_s$. The post detection filter need now only remove all unwanted components from the output ie all the components which appear as a result of the limiting and rectification processes in the detector. Bennett and Davey suggest that little is to be gained by attempting to find an

optimum post detection filter since the improvement in performance does not justify the effort involved. It would appear logical that a filter giving some similar shape to that in Fig 3.3 would be best provided that it suppresses sufficiently the unwanted components present.

In the design of the modem to follow, the intention is to apply the above derivations to the filter requirements of the modem, and to indicate why it was found necessary to deviate from the theory for reasons of practibility and compatibility.

In particular the Post Detection filter has to be designed bearing in mind the requirements that the pulse shape should not be distorted causing intersymbol interference, but at the same time, it is necessary to attenuate the 2nd harmonic of the data frequency which is contrary to the result above.

3.2 ADAPTIVE EQUALISATION

3.2.1 Basic Principles

An equaliser is a device inserted between the post detection filter and the data detector to compensate for the data channel imperfections in phase and amplitude response. The purpose of equalisation is to minimise the effects of intersymbol interference due to the channel. Note that the channel includes the modulator and demodulator as well as the external channel on radio or other bearer. Equalisation is not applied to the channel receive filter as there is no simple direct linear relationship between the mean squared error at the input to the data detector and the channel filter coefficients. Since the frequency modulation process is non-linear, it is a very complex operation to determine the effect of various sidebands on the signal at the data detector input.

The purpose of an adaptive equaliser is to actively compensate for channel variations in amplitude and phase response and to reduce the effects of background noise on a continuous basis. Fig 3.4 illustrates the linear adaptive equaliser. The delays T represent the time between data pulses and not, as in the case of other digital filters, the time between samples at the Nyquist rate.

The detector is simply a slicer, with all values positive of a threshold level giving an output

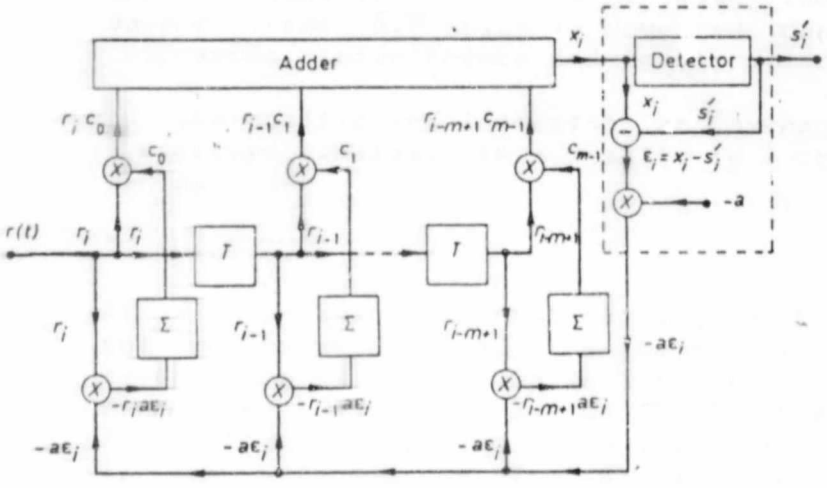


Fig 3.4 Structure of Linear Adaptive Equaliser

of one, and all values negative of the threshold resulting in an output of zero. At each data sampling instant, an error signal e_i is developed from the input to and output of the detector.

$$e_i = x_i - s_i' \qquad 3.13$$

where s_i represents the output at instant i.

The action of the equaliser is now to minimise this error by feeding the error signal back to the coefficients in such a way as to ultimately cause e_i to be minimised.

The coefficients are modified as follows. Firstly e_i is multiplied by -a, a simple constant determining the rate of convergence, and the integration time for noise. The tap gain coefficients are then incremented by

$$\delta c_j = -ae_i r_{i-j} \qquad 3.14$$

$$c_j = c_j - ae_i r_{i-j} \qquad 3.15$$

where r_{i-j} is the jth sample at instant i, and c_j is the jth coefficient.

Consider the case where e_i is negative ie x_i should be incremented to minimise e_i. After multiplication by -a, the new factor $-ae_i$ is positive. From 3.14, if r_{i-j} is positive, c_j is increased positively and when multiplied by the positive r_{i-j} will cause x_i to increase. Conversely, if r_{i-j} is negative, c_j will be decreased negatively and will reduce the effect of the negative r_{i-j} on x_i. In this simplistic way the equaliser tends to minimise e_i.

Since the actions described above apply to both positive and negative e_i, we may write

$$\overline{e_i^2} = \overline{(x_i - s_i)^2} \qquad 3.16$$

and the equaliser tends to minimise the mean-square error $\overline{e_i^2}$ which is composed partly of intersymbol interference and partly of noise.

If the coefficient increments are averaged over l received samples, this results in a change in tap gain of

$$\Delta c_j = -la\overline{e_i r_{i-j}} \qquad 3.17$$

It is clear therefore that for a given Δc_j, the smaller the value of a, the longer the received data sequence should be and consequently the effective integration period is lengthened. A long integration period means that the equaliser is not affected by instantaneous noise but rather by the long term noise variance σ^2. It therefore minimises the long term effects of the noise but does not tend to hunt due to the effects of individual noise peaks on received data samples.

In a channel free of noise ie with a high signal to noise ratio, th equaliser tends to minimise the effects of intersymbol interference due to

channel imperfections. The equaliser tracks a slowly-varying channel to maintain correct equalisation at all times.

A compromise has to be made between ability to track faster channel variations and a greater sensitivity to noise.

The advantages of the type of equaliser illustrated, the Least Mean Squares or LMS equaliser, over types with faster convergence are:

i) Computational efficiency: for each tap coefficient there is one multiplication and one addition, with an overhead of one subtraction and one multiplication. Therefore if an equaliser has m taps, the number of required operations is:

   m+1 multiplications and
   m+1 additions/subtractions.

ii) Arithmetic robustness: due to the simple construction requiring simple arithmetic, the equaliser is insensitive to limitations in numerical accuracy.

iii) Stability: due to its "ruggedness" resulting from simple structure and robustness, the equaliser is inherently stable and since no positive feedback is employed, this stability may be maintained for widely varying inputs.

The main disadvantage of the LMS equaliser is slow rate of convergence. However, in an application such as the one described in this paper, fast convergence is unnecessary as channels are largely time-invariant, and noise levels are low. At the data rate in question, viz 1200 bd, fairly large channel variations are necessary to cause a significant change in the error rate.

3.2.2 Training

An adaptive equaliser has to be "trained" to some extent. To allow the coefficients to converge on the values giving correct channel equalisation, ie to develop zeros to complement and cancel the channel poles, the consecutive values of s_n, the transmitted data pulses, should be statistically independent. The start up coefficients are not very critical, but at least one value should be non-zero, presenting an initially unequalised channel to the detector and allowing equalisation to proceed from there.

In the case of the modem as envisaged here, the data "channel" includes the modulator and demodulator which determines a basic channel response. If the equaliser starts from scratch, it has also to equalise the built in channel. If the equaliser could be started with these coefficients already set up, then it will converge much more quickly to the equalisation required for the bearer channel.

3.2.3 Other Considerations

It is important that the input to the equaliser be bipolar with preferably zero mean level ie no offset. This is to ensure that samples below the mean have the same weight as samples above the mean. This may be achieved by subtracting the offset prior to the equaliser or by subtracting the offset at the time at which each coefficient is calculated. This latter however leads to more computation and for reasons of time constraint, this is undesirable.

4. SOFTWARE DESIGN OF MODEM

4.1 The Modem Logical Structure

The functions implemented digitally within the modem are illustrated in Fig 4.1. This figure does not show

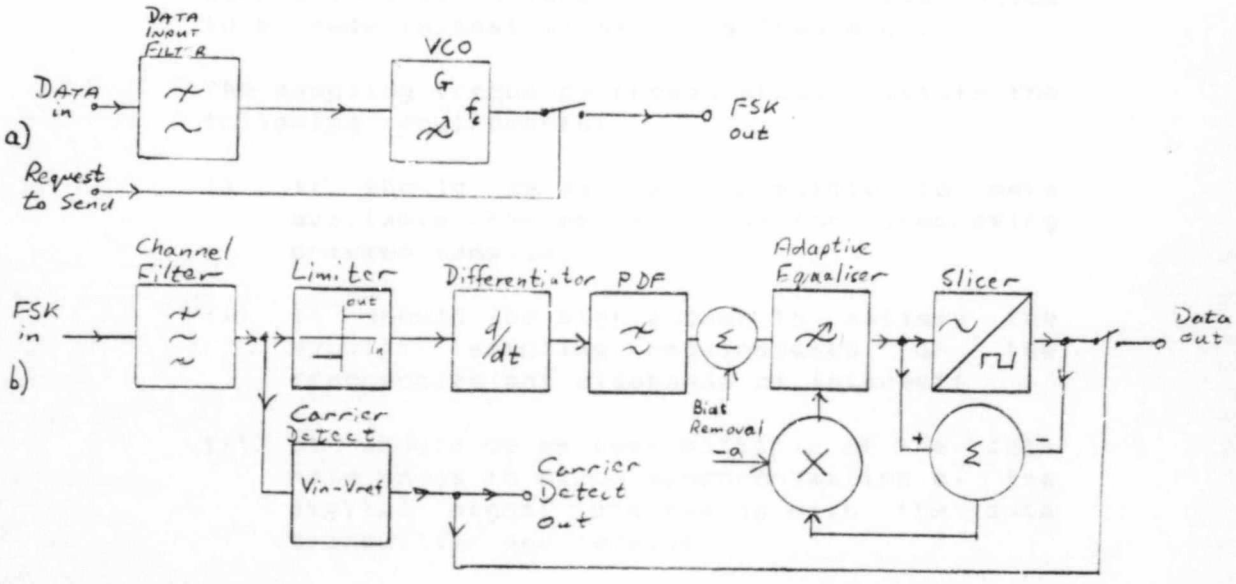


Fig 4.1 Modem Block Diagram a) Transmitter Section; b) Receiver Section

external anti-aliasing filters, latches etc.

The transmitter modulates the incoming digital data onto a phase continuous carrier. The data is filtered to limit sharp edges which would give rise to unwanted output sidebands. This filtered signal is applied to a linear frequency modulator for conversion to FSK. The Request-to-Send input determines whether or not the FSK signal is output to line. A more detailed explanation is provided in section 4.4.

The receiver operates as follows. The input band-pass filter limits the input to those sidebands which carry the bulk of the signal energy. Noise outside the band of signal frequencies is excluded as are adjacent channels etc. The limiter ensures that all the signals applied to the input of the differentiator are of the same amplitude as this is necessary for the correct operation of the whole discriminator. The differentiated signals have an amplitude linearly proportional to frequency and when rectified and filtered, this signal can be squared up and represents the data output. Between the Post Detection Filter and Slicer is the Adaptive Equaliser. A carrier detect prevents data output under low signal conditions. The receiver is explained in greater detail in section 4.5.

4.2 Choice of a Sampling Frequency

4.2.1 Principles Determining Choice

The design of software for real time signal processing poses some interesting and conflicting problems which necessarily result in compromises being made. The first obvious choice to be made is that of sampling frequency.

The sampling frequency chosen should satisfy the following requirements:

i) It should be as low as possible to make available the maximum time for processing between samples;

ii) It should be high enough to satisfy the Nyquist sampling requirements for the frequencies and sidebands of interest;

iii) It should be an even multiple of the data rate so as to allow synchronisation of the digital signal processing with the data transmitter and receiver;

iv) It should be as high as possible to limit the maximum uncertainty or jitter in the received data.

4.2.2 Determining the Sampling Frequency

Since all the processing is within the standard audio channel bandwidth of 4 kHz, a seemingly obvious choice of sampling frequency is 8 kHz, in line with the international PCM transmission and other standards. This frequency poses no problems for the FSK signal, either in generation or reception, and permits the use of standard anti-aliasing filters, but for the 1200 bd data signal which also has to be sampled at this rate, a major problem appears viz that the data is asynchronous to the clock, resulting in jitter due to the fact that it is impossible to synchronise the data in any way with the sampling clock. It is therefore desirable that the sampling rate be a multiple of the data rate. This is in line with the principle stated in (iii) above.

Several sampling rates present themselves, from 7200 Hz and up. The sampling frequency should be an even multiple of 1200, to ensure that each bit is sampled an equal number of times and should allow sufficient time for the required signal processing. A high sampling rate initially appears attractive, as with synchronised sampling and data clocks, the

theoretical maximum jitter will be

$$2 \times \frac{T_{sample}}{T_{data}} \times 100\% \qquad 4.1$$

The numerator allows for one sampling instant uncertainty either side of the correct transition point. This is depicted in Fig 4.2.

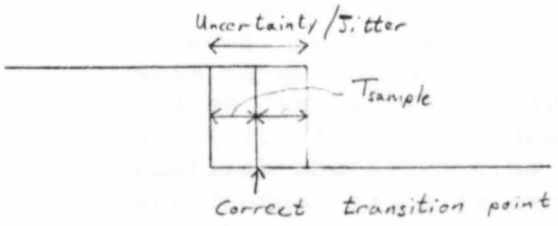


Fig 4.2 The effect of jitter on the data transitions.

A sampling rate of 19200 Hz was considered as this would result in a maximum jitter of 12.5%. However the time between samples, T_{sample}, is only 52 us, and as the TMS 320 executes instructions at approximately 5 MHz, this allows for only 260 instructions. Transversal filters, which are very attractive for their flat delay characteristics as well as stability and simplicity, have a length roughly inversely proportional to their bandwidth and this mitigates against a high sampling frequency as not enough instructions would be available to implement an adequate filter and carry out all the other required functions as well.

A sampling frequency of 9600 Hz was chosen. This frequency complies with principles (ii) and (iii). The highest frequency of possible interest is 3400 Hz as determined by a standard channel bandwidth, and 9600 is a factor eight greater than the bit rate, giving eight samples per bit. The time between samples is 104 micro-seconds resulting in a possible 520 instructions between samples (the TMS 320 executes instructions at the rate of 5 per usec); shorter filters may be used for a given bandwidth resulting in a saving of memory and execution time. For elements such as the Post Detection filter and Adaptive Equaliser, the filter length should at least span three data bits, and in the case of the latter, preferably more. At only 8 samples per bit, it is clear that excessive filter lengths are avoided. The foregoing are in line with (i) and represent a compromise with (iv) as the jitter is potentially high at 25%. This may be offset against improved modem performance in other areas made possible by implementation of more accurate filters, equalisers and other facilities.

4.3 Software

4.3.1 Software Requirements and Limitations

The software is required to carry out all the necessary signal processing to implement a modem. Additionally, provision must be made for the future addition of general or special purpose routines to aid in the operational setting-up or testing of the modem and the system in which it operates. It should make provision for the incorporation of self-checking routines and allow time for these to be run on a continuous basis to permit early detection of possible fault conditions within the processing system.

The constraints placed on the system are time and memory. As seen in section 4.2.2 above, there are only 520 steps between samples which may be used to implement the modem, carry out self-checking procedures and execute any other functions as required. The TMS 320 has 144 16-bit words of internal data RAM and can address 4k words of external program memory. (See Appendix F for the TMS 320 specifications) In the project being described here, only 2k words of memory are allowed for in the hardware. To permit as many functions as possible to be implemented and carried out during the sampling interval, the code should be as time efficient as possible within the constraints of the limits placed on the program memory.

4.3.2 Implementation of Requirements

To synchronise the signal processing portion of the program, an input is required from the timing hardware to start the loop. The TMS 320 offers two means of achieving this - a software interrupt trapped by a special test and branch instruction, BIO, and a hardware interrupt facility. The former requires the programmer to determine within a few instructions the exact length of the loop, if it is desired to make maximum use of the available time for background or other tasks. All tasks should be complete before the interrupt is expected, whereupon the processor loops, while polling the BIO input for the interrupt. The hardware interrupt input works conventionally, placing the return address on the stack and branching to a fixed program location, in this case 0002. This permits background tasks to run continuously, signal processing being done on demand from the interrupt input. This latter method has been chosen for this application as no knowledge is

required for the execution time of the background loop.

The system always runs in a background self-checking mode, until interrupted, when it will execute the processing loop falling back into the self-checking routines when complete, awaiting the next interrupt. This permits the two functions of the program to be transparent to each other. The method employed presupposes that the processing code does not use all the available sample time, allowing some for self-tests to be executed. Some time overhead must be allowed for the interrupt, the branch to the service routine, in this case the main processing loop, saving of important parameters from the background task, the re-enabling of interrupts, and return from interrupt.

The start-up routines clear all the data RAM, load the constants, and test for the desired function. There is at present only one function implemented ie the 1200 bd modem. Other envisaged functions are a 200 bd modem, tone generators etc up to a total of 16, set by switches and selected on power-up. This flexibility emphasises the potential advantages of a digital signal processor as opposed to discrete components when implementing complex functions. When these start-up routines are complete, the processor will enter the background task which will check the contents of data RAM and program memory for incorrect check-sums, resetting when an error is encountered. The check-sums could be replaced by more detailed integrity tests. The only areas of data RAM checked will be that containing constants, although the operation of the balance of the RAM may be tested by writing and reading to and from it. Finally, if the modem is incorporated as part of an overall system, it may be possible to interrogate its status directly, and trigger the operation of the various test functions incorporated.

The approach taken to obtain the highest level of time efficiency in the signal processing software has been to use straight line code throughout. This removes all looping instructions, pointers and the required setup instructions with a large saving in the time required to implement structures such as long filters. For example a 23 tap transversal filter implemented using looping and where the data RAM does not happen to lie at the bottom end of the internal memory (this would permit the loop counter to be used as a pointer as well), requires 161 operations compared with 48 for the same filter using straight line code. The former requires less program memory than the latter viz 14 instructions versus 48. Also the looped code

length does not change with length of filter, whereas the straight line coding adds 2 instructions per filter tap. However with a possible 2048 instructions and a maximum of 520 per sample interval, even if the whole sample interval were used, there is still ample space for the program and any tables that may be required.

The time penalty involved with looping code rules out the use of more elegant software approaches such as a generalised filter procedure which may be called with the relevant variables set up as required. Looping code is employed in the set-up routines at present, and future background tasks will make use of this technique, as the emphasis is placed on memory efficiency rather than time.

This latter point emphasises an important consideration and trade-off when developing real-time software, into which category signal processing software falls. Real-time applications attempt to squeeze as many functions as possible into a limited time, with the result that where possible, branches and procedure calls are removed, being replaced by straight line code. This is possible only where the straight line code segments are sufficiently short so as not to absorb too much program memory. A disadvantage of this type of code is that the self-documenting characteristics of structured code are lost. Fortunately in this application, the code segments in the signal processing portion of the routine are individually simple, although long.

The above paragraphs have examined in detail the considerations for the development of application software for real-time signal processing, and has indicated how these were applied to the present project.

4.4 Transmitter

The purpose of the transmitter is to convert the incoming data pulses to either the mark or space frequency. The generation of the two frequencies and the switching between them shall be phase continuous to avoid the generation of undesirable sidebands. Furthermore the incoming data pulses are shaped by rounding off the sharp edges. This further slightly reduces the energy in wasted in high order harmonics and their resulting sidebands. It is naturally desirable to have the available power concentrated in those sidebands which carry the bulk of the information. The outer sidebands help in shaping the received pulses, but as the receiver already has pulse shaping features, these sidebands are of limited importance.

A further danger incurred by allowing too many sidebands is foldover at the low frequency end of the spectrum. The input pulse shaping assists in the reduction of these foldover effects.

The transmitter or FSK modulator tone generator operates as follows. There is a rotating vector r whose position is determined by the phase angle α. See Fig 4.3.

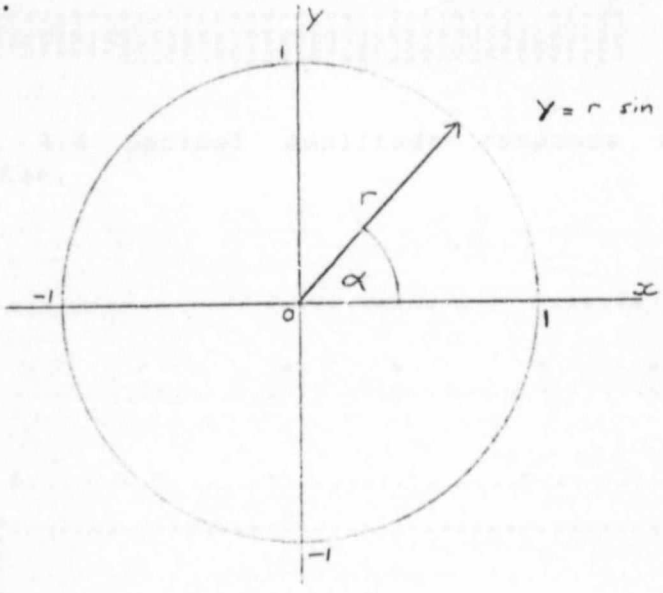


Fig 4.3 Tone Generator Operation

α is incremented on every pass through the program loop, the amount by which it is incremented determining the rate of rotation and hence frequency of rotation. The output frequency is y where

$$y=\sin \alpha \qquad 4.2$$

and

$$\alpha_{i+1}=\alpha_i+\phi \qquad 4.3$$

where ϕ is the increment angle.

By changing ϕ in accordance with the input data, the

output of the modulator will be a frequency modulated signal ie Frequency Shift Keyed.

The incoming data passes through the Data Input Filter, a simple 3 tap transversal filter which smooths the sharp edges. Fig 4.4 shows the response of the filter and Fig 4.5 shows the effect on the incoming square pulses. The filter output drives the modulator.

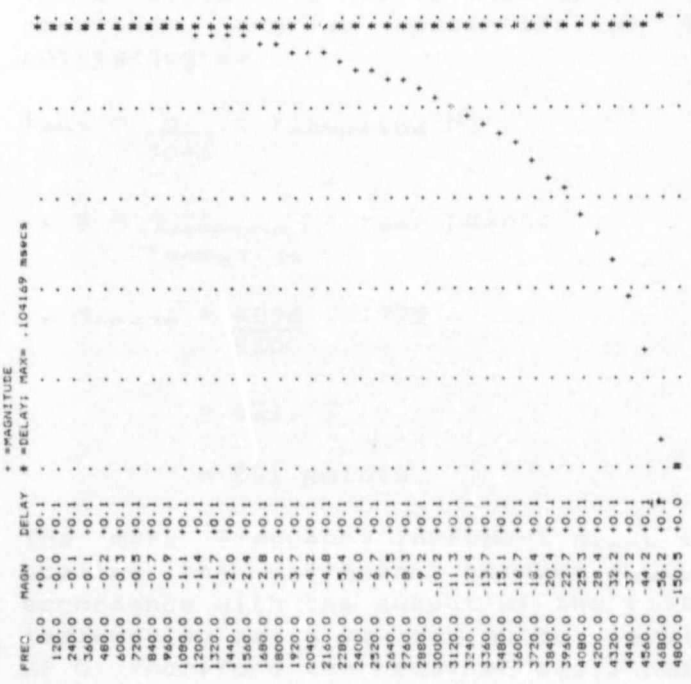


Fig 4.4 Decibel amplitude response of Data Input Filter.

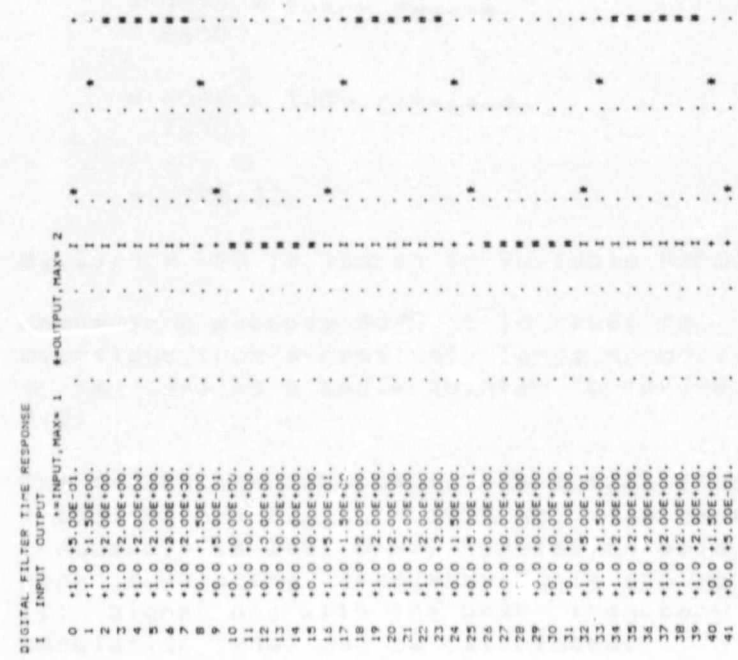


Fig 4.5 Time response of Data Input Filter showing rounding of the edges of the data pulses. Pulses at 600 Hz, 1:1 mark to space ratio.

Without stimulus, the modulator outputs the space frequency ie 1925 Hz. ϕ is stored in the variable RAMPHI in data RAM and has the value 821_{10} derived as follows.

The circle of Fig 4.3 is made up of 4096 points, with a resultant angular resolution of 4096/360 = 11.4 points/degree.

$$f_{out} = \frac{\phi}{4096} \times f_{sampling} \text{ Hz} \qquad 4.4$$

$$\therefore \phi = \frac{4096}{f_{sampling}} \times f_{out} \text{ points} \qquad 4.5$$

$$\therefore \phi_{space} = \frac{4096}{9600} \times 1925$$

$$= 821.33$$

$$= 821 \text{ points}$$

The mark frequency increment ϕ_{mark} is calculated by scaling the difference between ϕ_{mark} and ϕ_{space} in accordance with the output of the filter. The filter as implemented has a maximum output level of 4 and minimum of 0. Therefore the required ϕ_{diff} has to be divided by 4. We have

$$\phi_{space} = 821.33$$

$$\phi_{diff} = \phi_{mark} - \phi_{space} \qquad 4.6$$

$$= \frac{4096}{9600} \times f_{mark} - \phi_{space}$$

$$= \frac{4096}{9600} \times 1050 - \phi_{space}$$

$$= -373.33$$

$\phi_{diff}/4 = -93$ is stored in variable RMPDIF.

Whenever $\propto$ exceeds 4095 it is reset to -4096 to avoid overflows from excessively large numbers. The resulting $\propto$ is used as a table pointer to derive $y=\sin\propto$ as in 4.2.

The transmitter therefore has output frequencies f_{mark} = 1050 Hz and f_{space} = 1925 Hz. The centre or "carrier frequency" is 1487.5 Hz, deviation being 437.5 Hz. The peak modulating frequency f_m is 600 Hz for a 1200 bd 1:1 signal and with the peak frequency deviation, a modulation index may be calculated.

$$\beta = \frac{\Delta f_{peak}}{f_m} \qquad 4.7$$

$$= \frac{437.5}{600}$$

ie β = 0.73

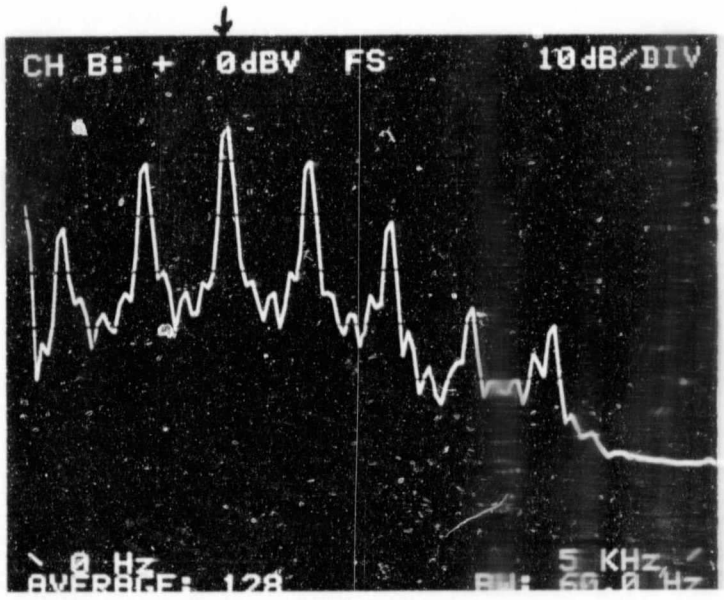


Fig 4.6 Photograph of spectrum of FSK signal modulated with 1:1 mark:space ratio data at 600 Hz. The centre frequency of 1487.5 Hz is arrowed. Note the sidebands at intervals of 600 Hz. The spectrum has been averaged over 128 samples.

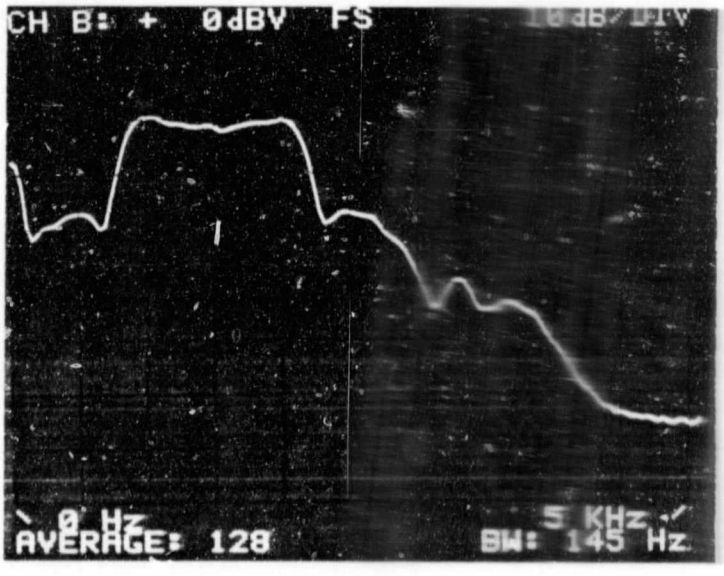


Fig 4.7 Photograph of spectrum of FSK signal modulated by 2047 bit pseudo-random data averaged over 128 samples. Note that the main portion signal lies in the region $f_c \pm 600$ Hz. See also section 4.5.1.

This value of β only applies for sine-wave modulation, but for the sake of simplicity, harmonics of 600 Hz will be ignored. From tabulated Bessel functions, $J_n(\beta)$ the relative amplitudes of the carrier and sidebands may be found, where n is the sideband order. In this

case, sidebands 1 and 2 are significant, sideband 3 being 34 dB down on the modulated carrier. (From Fig 4.6). This indicates that foldover effects at the low frequency end of the spectrum are unimportant.

When pseudo-random data is frequency modulated onto a carrier, the characteristic shown in Fig 4.7 results. The shape of this characteristic is more representative of real signals and contrary to Fig 4.6, suggests that beyond a bandwidth equal to the maximum data rate, preservation of the signal would not improve performance.

4.5 Receiver

The overall operation of the filter is explained in section 4.1.

4.5.1 Receive Filter

The purpose of the receive filter is to pass the band of frequencies carrying the most significant portion of the information. For FSK data transmission, this is considered to be one sideband on either side of the centre frequency f_c. The signalling rate f_s is double the data frequency for binary Non Return to Zero (NRZ) data ie if f_1 is the data frequency, then

$$f_1 = \frac{f_s}{2} \qquad 4.8$$

The signalling or data rate is 1200 bd, resulting in a data frequency of 600 Hz. With FSK as for FM, the sidebands are spaced about the carrier f_c at intervals of f_m, the modulating frequency. The sideband frequencies are then

$$f_{sb} = f_c \pm n f_m \qquad n=1,2,... \qquad 4.9$$

These can clearly be seen in Fig 4.6. Consider also Fig 4.7 for the spectrum of a pseudo-random data stream. From these photographs it may be seen that the most significant portion of the signal lies in the range $f_c \pm f_m$.

The input filter was designed with these spectra in mind. A very wide bandwidth, although passing more sidebands and theoretically resulting in a better pulse shape at the FM detector output, also admits noise in proportion to the available bandwidth.

The filter was designed to have a nominal bandwidth of 1200 Hz, and to achieve at least 30 dB attenuation above 2800 Hz to allow a 200 bd modem to be operated with a centre frequency of 3000 Hz over the same channel. The filter should have the minimum number of steps to reduce computation time, as well as the amount of straight line code. A Low-Pass prototype was developed from "ideal" coefficients ie those derived from the impulse response of an ideal filter. A Kaiser window was applied to the "ideal" filter, after which the filter was transformed to bandpass by scaling each coefficient by the following factor

$$a_i = a_i \times 2 \cos 2\pi i \frac{f_c}{f_{sampling}} \qquad 4.10$$

Author Andrews Donald Graham
Name of thesis A Software Engineering Approach To Digitally Synthesised Modem Using A Texas Instruments Tms320 Signal Processor. 1985

PUBLISHER:
University of the Witwatersrand, Johannesburg
©2013

LEGAL NOTICES:

Copyright Notice: All materials on the University of the Witwatersrand, Johannesburg Library website are protected by South African copyright law and may not be distributed, transmitted, displayed, or otherwise published in any format, without the prior written permission of the copyright owner.

Disclaimer and Terms of Use: Provided that you maintain all copyright and other notices contained therein, you may download material (one machine readable copy and one print copy per page) for your personal and/or educational non-commercial use only.

The University of the Witwatersrand, Johannesburg, is not responsible for any errors or omissions and excludes any and all liability for any errors in or omissions from the information on the Library website.